Available online at www.sciencedirect.com

ScienceDirect

Computers and Mathematics with Applications 53 (2007) 1658–1664

An International Journal
computers & mathematics with applications

www.elsevier.com/locate/camwa

Single-machine scheduling problems with start-time dependent processing time

Wen-Hung Kuo, Dar-Li Yang*

Department of Information Management, National Formosa University, Yun-Lin 632, Taiwan, ROC

Received 10 April 2006; received in revised form 28 June 2006; accepted 21 July 2006

Abstract

In this paper, we study the single-machine makespan scheduling problem with start-time dependent processing time. The objectives are to minimize the makespan and to minimize the sum of the kth powers of completion times. We prove that several cases are polynomially solvable under some restrictions of the parameters. In addition, these cases still remain polynomially solvable when the restriction of a certain parameter is relaxed from a positive integer to a real number.

Keywords: Single-machine; Scheduling; Time dependent; Makespan; Completion time

1. Introduction

In classical scheduling problems, processing times of jobs are assumed to be constant. However, there are many situations where the processing time depends on the starting time of the job. For example, deterioration in processing time may occur when the machine gradually loses efficiency in the course of processing jobs. In the beginning, the machine is assumed to be at its highest level of efficiency. The efficiency loss is reflected in the fact that a job processed later has a longer processing time.

According to the notion mentioned above, many scheduling problems with linear, piecewise linear or nonlinear time-dependent processing times are studied in the literature (see Alidaee and Womer [1]). There are many interesting efficient ordering policies provided for some scheduling problems with linear or piecewise linear processing times. However, there are less efficient rules available for those with nonlinear processing times. Only a few heuristic algorithms were proposed. For example, Gupta and Gupta [2] studied the single-machine makespan scheduling problem with nonlinear processing times. In this study, the complexity of the problem was conjectured to be NP-hard. Thus, Gupta and Gupta [2] proposed two heuristic algorithms to solve the problem. Further, Alidaee [3] proposed a more efficient heuristic algorithm to solve this problem when the polynomial function is differentiable.

In this paper, we focus on the scheduling problem studied by Gupta and Gupta. In their study, the actual processing time (p_i) of job i ($i = 1, 2, \dots, n$) is a polynomial function of its starting time (t_i) and it can be expressed as follows:

$$p_i = a_{i0} + a_{i1}t_i + a_{i2}t_i^2 + \cdots + a_{im}t_i^m \quad \text{for } i = 1, 2, \dots, n, \tag{1}$$

* Corresponding author. Tel.: +886 5 6315733; fax: +886 4 22939659.
E-mail addresses: whkuo@nfu.edu.tw (W.-H. Kuo), dlyang@nfu.edu.tw (D.-L. Yang).

0898-1221/$ - see front matter
doi:10.1016/j.camwa.2006.07.012

where $a_{i0}, a_{i1}, \ldots, a_{im}$ are positive constants and m is a positive integer. We show that some special cases of this problem have polynomial time solutions. Furthermore, these cases still remain polynomially solvable when the restriction of m is relaxed from a positive integer to a real number.

2. Problem description

Assume that a set of n jobs is available to be processed on a single machine at time zero. Neither job splitting nor machine idleness is allowed. The processing time of each job depends on its starting time in the sequence and is defined as Eq. (1). The objectives are to minimize the makespan and the sum of the kth powers of completion times of all jobs, respectively. Therefore, using the three-field notation introduced by Graham et al. [4], the corresponding problems are respectively denoted by $1/p_i = a_{i0} + \sum_{k=1}^{m} a_{ik} t_i^{r_k} / C_{\max}$ and $1/p_i = a_{i0} + \sum_{k=1}^{m} a_{ik} t_i^{r_k} / \sum C_i^k$, where r_k is a real number.

2.1. The makespan minimization

In this section, we study the $1/p_i = a_{i0} + \sum_{k=1}^{m} a_{ik} t_i^{r_k} / C_{\max}$ problem, where $a_{ik} = \lambda_k$ and $a_{ik} = \lambda_k a_{i0}$ respectively. First, an elementary lemma is provided.

Lemma 1. (a) $\lambda(1-(1+\tau)^{\alpha}) - (1-(1+\lambda\tau)^{\alpha}) \geq 0$ *if* $\lambda \geq 1, \tau \geq 0$ *and* $\alpha \leq 0$.
(b) $\lambda(1-(1+\tau)^{\alpha}) - (1-(1+\lambda\tau)^{\alpha}) \leq 0$ *if* $\lambda \geq 1, \tau \geq 0$ *and* $0 < \alpha < 1$.
(c) $\lambda(1-(1+\tau)^{\alpha}) - (1-(1+\lambda\tau)^{\alpha}) \geq 0$ *if* $\lambda \geq 1, \tau \geq 0$ *and* $\alpha \geq 1$.

Proof. (a) See the proof of Lemma 2 in Kuo and Yang [5,6].
(b) Similar to the proof of (a).
(c) Similar to the proof of (a). □

Proposition 1. *If* $a_{i1} = \lambda_1, a_{i2} = \lambda_2, \ldots, a_{im} = \lambda_m$ *for all* $i = 1, 2, \ldots, n$ *and* $r_i \in [0, \infty)$ *for all* $i = 1, 2, \ldots, m$, *then for the scheduling problem* $1/p_i = a_{i0} + \sum_{k=1}^{m} \lambda_k t_i^{r_k} / C_{\max}$, *there exists an optimal schedule in which the job sequence is in non-decreasing order of* a_{i0}.

Proof. Let $S_1 = (\pi_1, J_h, J_j, J_i, \pi_2)$ denote a sequence with job J_j processed immediately before job J_i. π_1 and π_2 are the partial sequences of S_1. In addition, π_1 or π_2 may be empty. We will show that it does not increase the makespan to interchange J_j and J_i in S_1. Thus, we deduce that Proposition 1 holds.

Let S_2 be the same sequence with J_j and J_i mutually exchanged. In addition, let $C_l(S_1)$ denote the completion time of J_l in sequence S_1 and $C_l(S_2)$ denote the completion time of J_l in sequence S_2. Without loss of generality, we assume that $C_h(S_1) = C_h(S_2) = t$. Thus, we have

$$C_i(S_1) = C_h(S_1) + \left(a_{j0} + \sum_{k=1}^{m} \lambda_k t^{r_k}\right) + \left(a_{i0} + \sum_{l=1}^{m} \lambda_l \left(t + a_{j0} + \sum_{k=1}^{m} \lambda_k t^{r_k}\right)^{r_l}\right)$$

and

$$C_j(S_2) = C_h(S_2) + \left(a_{i0} + \sum_{k=1}^{m} \lambda_k t^{r_k}\right) + \left(a_{j0} + \sum_{l=1}^{m} \lambda_l \left(t + a_{i0} + \sum_{k=1}^{m} \lambda_k t^{r_k}\right)^{r_l}\right).$$

Let $T = t + \sum_{k=1}^{m} \lambda_k t^{r_k} \geq 0$, then we have

$$C_j(S_2) - C_i(S_1) = \sum_{l=1}^{m} \lambda_l \left((a_{i0} + T)^{r_l} - \left(a_{j0} + T\right)^{r_l}\right). \tag{2}$$

Therefore, $C_j(S_2) \leq C_i(S_1)$ if $a_{i0} \leq a_{j0}$. Thus, repeating this interchange argument for all jobs not sequenced according to non-decreasing order of a_{i0} completes the proof of Proposition 1. □

Proposition 2. *If* $a_{i1} = \lambda_1, a_{i2} = \lambda_2, \ldots, a_{im} = \lambda_m$ *for all* $i = 1, 2, \ldots, n$ *and* $r_i \in (-\infty, 0]$ *for all* $i = 1, 2, \ldots, m$, *then for the scheduling problem* $1/p_i = a_{i0} + \sum_{k=1}^{m} \lambda_k t_i^{r_k} / C_{\max}$, *there exists an optimal schedule in which the job sequence is in non-increasing order of* a_{i0}.