Information Processing Letters 114 (2014) 692–695

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Serial batch scheduling on uniform parallel machines to minimize total completion time ☆

Song-Song Li *, Yu-Zhong Zhang

School of Management, Qufu Normal University, Rizhao 276826, PR China

ARTICLE INFO

Article history:
Received 26 January 2014
Received in revised form 21 June 2014
Accepted 23 June 2014
Available online 25 June 2014
Communicated by M. Yamashita

Keywords:
Scheduling
Serial batch
Parallel machine
Total completion time
Rejection
Polynomial time algorithm

ABSTRACT

We consider two scheduling problems on m uniform serial batch machines where m is fixed. In the first problem, all jobs have to be scheduled and the objective is to minimize total completion time. In the second problem, each job may be either rejected or accepted to be scheduled and the objective is to minimize the sum of total completion time and total rejection penalty. A polynomial time procedure is presented to solve both problems with the time complexity $O(m^2n^{m+2})$ and $O(m^2n^{m+5})$, respectively.

1. Introduction

Batch scheduling problems have been extensively studied in the literature (see, e.g., [1–5]), and single-machine serial-batch scheduling problems have received much attention too (see, e.g., [6–14]). However, to the best of our knowledge, little work has been done on serial batch scheduling problems with parallel machines involved. So we study two serial batch scheduling problems on uniform parallel machines in order to attract more attention to this subject. Note that one of our problems involves rejection which is also an attractive research field (see a survey by Shabtay et al. [15]).

The first problem studied in this note can be stated as follows. There are n jobs $J_1, \cdots, J_n$, where each job J_j $(1 \le j \le n)$ has a processing time $p_j > 0$. All the jobs have to be partitioned into batches and all the batches have to be scheduled on m uniform serial batch machines $M_1, \cdots, M_m$ where m is fixed. Each machine M_i $(1 \le i \le m)$ has a speed $v_i > 0$, a batch capacity $b_i > 0$ and a setup time $s_i \ge 0$, which implies that machine M_i can process a batch B, which includes at most b_i jobs, in a processing time $p_i(B) = s_i + (\sum_{J_j \in B} p_j)/v_i$, and the completion time C_j of each job $J_j \in B$ is equal to the completion time of the batch B. The objective is to minimize the total completion time of all the jobs. Following the three-field notation introduced by Graham et al. [16], this problem can be referred to by $Q_m \mid s\text{-}batch \mid \sum C_j$.

In the second problem where rejection is involved, every job J_j $(1 \le j \le n)$ has to be either accepted to be scheduled on the m uniform serial batch machines, or rejected with a rejection penalty $e_j \ge 0$. The objective is to minimize the sum of the total completion time of the accepted jobs and the total rejection penalty of the rejected jobs. Let A be the set of the accepted jobs and $\bar{A}$ be the

☆ Research supported by the Special Funds of the National Natural Science Foundation of China (Grant Nos. 61340045 and 11201259), the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20123705110003) and Innovation Project of Shangdong Graduate Education (Grant No. SDYC13036).

* Corresponding author. Tel.: +86 13256067845.
E-mail addresses: 163lisongsong@163.com (S.-S. Li), yuzhongrz@163.com (Y.-Z. Zhang).

http://dx.doi.org/10.1016/j.ipl.2014.06.009
0020-0190/

set of the rejected jobs. Then this problem can be referred to by $Q_m \mid s\text{-}batch, rej \mid \sum_{J_j \in A} C_j + \sum_{J_j \in \bar{A}} e_j$.

We now pay attention to the papers that are related to our research. Coffman et al. [6] provide an algorithm to solve the problem $1 \mid s\text{-}batch \mid \sum C_j$ in $O(n \log n)$ time. The problems $P \mid\mid \sum C_j$ and $R \mid\mid \sum C_j$ can be solved with the time complexity $O(n \log n)$ and $O(n^3)$ respectively, where P (R) indicates identical (unrelated) parallel machines (see, [17–20]). Shabtay [14] shows that the problem $1 \mid s\text{-}batch, rej \mid \sum_{J_j \in A} C_j + \sum_{J_j \in \bar{A}} e_j$ can be solved in $O(n^5)$ time by a dynamic programming.

This note is organized as follows. In Section 2, we provide a polynomial time procedure to solve our first problem in $O(m^2 n^{m+2})$ time. In Section 3, the procedure is modified to solve our second problem in $O(m^2 n^{m+5})$ time. In Section 4, we conclude our note.

2. Minimizing the total completion time

In this section, we solve the problem $Q_m \mid s\text{-}batch \mid \sum C_j$ by a dynamic programming in $O(m^2 n^{m+2})$ time. Without loss of generality, assume that $p_1 < p_2 < \cdots < p_n$. We say batch B is regular if and only if $B = \{J_j : l_B \le j \le u_B\}$ where $l_B = \min\{j : J_j \in B\}$ and $u_B = \max\{j : J_j \in B\}$. Then we can focus on the schedules with specific properties according to the following lemma.

Lemma 1. *For problem $Q_m \mid s\text{-}batch \mid \sum C_j$, there is an optimal schedule such that the following two properties are both satisfied:*

1. *If two jobs J_j and $J_{j'}$ $(1 \le j < j' \le n)$ are both scheduled on the same machine, then $C_j \le C_{j'}$.*
2. *All the batches that are scheduled are regular.*

Proof. It is easily seen that all the optimal schedules satisfy property 1. Suppose now that none of the optimal schedules satisfy property 2, which implies that in any optimal schedule, there is at least one non-regular batch. Let $l_\pi = \min\{j : J_j$ belongs to a non-regular batch in optimal schedule $\pi\}$, $u_\pi = \max\{j : J_j$ belongs to the batch which includes J_{l_π} in optimal schedule $\pi\}$, $L = \max\{l_\pi : \pi$ is an optimal schedule$\}$, $\Pi = \{$optimal schedule $\pi : l_\pi = L\}$, $U = \min\{u_\pi : \pi \in \Pi\}$ and $\Pi' = \{\pi \in \Pi : u_\pi = U\}$. Note that $\Pi' \ne \emptyset$.

Let schedule $\sigma \in \Pi'$. Let B be the batch that includes jobs J_L and J_U in σ, which implies that batch B is not regular. Without loss of generality, assume that batch B is scheduled on machine M_i $(1 \le i \le m)$ and there is a job $J_k \notin B$ where $L < k < U$ in σ. Then according to property 1, job J_k cannot be scheduled on M_i in σ, so we can assume that $J_k \in B'$ which is scheduled on machine $M_{i'}$ $(1 \le i' \le m$ and $i' \ne i)$. Let $n_i = |N_i|$ where $N_i = \{J_j : j \ge L$ and J_j is scheduled on M_i in $\sigma\}$, and let $n_{i'} = |N_{i'}|$ where $N_{i'} = \{J_j : J_j$ is scheduled on $M_{i'}$ and $C_j \ge C_k$ in $\sigma\}$. We now claim that $n_i/v_i = n_{i'}/v_{i'}$.

In fact, if $n_i/v_i < n_{i'}/v_{i'}$, then we will let feasible schedule σ_1 be the same as σ except that job J_L will be included in batch B' and be scheduled on machine $M_{i'}$, and that job J_k will be included in batch B and be scheduled on machine M_i. Let $f(\sigma)$ and $f(\sigma_1)$ be the objective values of σ and σ_1 respectively, then we have that $f(\sigma_1) = f(\sigma) + n_i(p_k - p_L)/v_i - n_{i'}(p_k - p_L)/v_{i'} < f(\sigma)$, contradicting the fact that σ is an optimal schedule. Else if $n_i/v_i > n_{i'}/v_{i'}$, then we will let feasible schedule σ_2 be the same as σ except that job J_U will be included in batch B' and be scheduled on machine $M_{i'}$, and that job J_k will be included in batch B and be scheduled on machine M_i. Then we have that the objective value of σ_2 is equal to $f(\sigma) - n_i(p_U - p_k)/v_i + n_{i'}(p_U - p_k)/v_{i'}$ which is less than $f(\sigma)$, contradicting the fact that σ is an optimal schedule too. So $n_i/v_i = n_{i'}/v_{i'}$.

Now, we consider the feasible schedule σ_2 again. It is easily seen that σ_2 is an optimal schedule and $l_{\sigma_2} = L$ (if $l_{\sigma_2} < L$, then l_σ should be less than L too; else if $l_{\sigma_2} > L$, then it contradicts the definition of L), which imply that $\sigma_2 \in \Pi$, which together with the fact that $u_{\sigma_2} < U$ contradicts the definition of U. □

Recall that our goal is to find an optimal schedule of the n jobs. Now, we can define $\Pi(j, r, k, q_i\ (i = 1, \cdots, m))$ as an optimal partial solution on job set $\{J_j, \cdots, J_n\}$, with jobs J_j and J_k starting and ending the first batch on machine M_r, and with q_i jobs scheduled on machine M_i.

We say a partial schedule $\Pi(j, r, k, q_i)$ is proper if and only if $1 \le j \le n$, $1 \le r \le m$, $j \le k \le \min\{n, b_r + j - 1\}$, $0 \le q_i \le n - k$ $(i = 1, \cdots, r-1, r+1, \cdots, m)$, $k - j + 1 \le q_r \le n - j + 1$ and $\sum_{i=1}^m q_i = n - j + 1$. Note that the number of proper partial schedules $\Pi(j, r, k, q_i)$ is $O(mn^{m+1})$ as there may be $O(n)$ different j values, $O(m)$ different r values, $O(n)$ different k values and $O(n)$ different q_i $(i = 1, \cdots, m-1)$ values (note that $q_m = n - j + 1 - \sum_{i=1}^{m-1} q_i$).

Let $F(j, r, k, q_i\ (i = 1, \cdots, m))$ denote the objective value of partial schedule $\Pi(j, r, k, q_i)$. Then $F(j, r, k, q_i) = +\infty$ when $\Pi(j, r, k, q_i)$ is not proper. And the initial conditions are given by

$$F(j, r, n, q_i) = q_r \left(s_r + \frac{\sum_{h=j}^n p_h}{v_r} \right),$$

where $j = j', \cdots, n$ $(j' = \max\{1, n + 1 - b_r\})$, $r = 1, \cdots, m$, $q_i = 0$ $(i = 1, \cdots, r-1, r+1, \cdots, m)$ and $q_r = n - j + 1$.

Note that any proper partial schedule $\Pi(j, r, k, q_i)$ can be obtained by adding jobs $J_j, \cdots, J_k$ as a batch to machine M_r in the optimal one of partial schedules $\Pi(k+1, r', k', q_i')$, where $q_i' = q_i$ $(i = 1, \cdots, r-1, r+1, \cdots, m)$ and $q_r' = q_r - (k - j + 1)$. So we have the following dynamic programming formula $(DP1)$:

$$F(j, r, k, q_i) = q_r \left(s_r + \frac{\sum_{h=j}^k p_h}{v_r} \right) + \min\left\{ F\left(k+1, r', k', q_i'\right) \right\}$$

where $r' = 1, \cdots, m$, $k' = k+1, \cdots, \min\{n, k + b_r\}$, $q_i' = q_i$ $(i = 1, \cdots, r-1, r+1, \cdots, m)$ and $q_r' = q_r - (k - j + 1)$. Note that $(DP1)$ requires $O(mn)$ time.

Finally, the optimal objective value of scheduling the n jobs is equal to

$$\min\left\{ F(1, r, k, q_i) : \Pi(1, r, k, q_i) \text{ is proper} \right\},$$