ICIC Express Letters
Volume 11, Number 4, April 2017

ICIC International ©2017 ISSN 1881-803X
pp. 893–898

SINGLE AND PARALLEL MACHINE SCHEDULING TO MINIMIZE THE TOTAL STRETCH

Jang Rae Lee[1] and Suk-Hun Yoon[2,*]

[1]Department of Industrial Engineering
Seoul National University
1 Gwanak-ro, Gwanak-gu, Seoul 08826, Korea
future33@nate.com

[2]Department of Industrial and Information Systems Engineering
Soongsil University
369 Sangdo-ro, Dongjak-gu, Seoul 06978, Korea
[*]Corresponding author: yoon@ssu.ac.kr

Received October 2016; accepted January 2017

Abstract. *We consider scheduling problems in a single machine and parallel machines to minimize the total stretch. The stretch of a job is defined as the ratio of the flow time to its processing time. When jobs have different release times, the problem of minimizing the total stretch is NP-complete even for a single machine. In this paper, we assume that the release times of jobs are all zeros. We provide some optimality conditions and propose a polynomial time algorithm for the problems.*
Keywords: Non-preemptive scheduling, Parallel machines, Total stretch, Polynomial time algorithm

1. **Introduction.** Service providers frequently offer different levels of services to special customers. For instance, some counters in a supermarket are reserved for small purchase customers to reduce average waiting times by accelerating the check-out process. The stretch is defined as the ratio of a job flow time to its processing time, which relates the customers' waiting times to their demands [1].

Glass and Kellerer [2] have provided an algorithm with worst-case performance of 3/2 when processing times are restricted to 1 and 2 in parallel identical machines to minimize the makespan. Azar et al. [3] have presented an online algorithm to minimize the makespan with a competitive ratio $\lceil \log_2 m \rceil + 1$ for m machines. Legrand et al. [4] have shown that scheduling problems to minimize the total stretch in unrelated parallel machines under the divisible load are NP-complete. Muthukrishnan et al. [5] have presented online scheduling to minimize the total stretch. They provide lower bounds on the competitive ratio of online algorithms for single and parallel machines.

Xu and Nagi [6] have provided a mixed integer programming problem to minimize makespan and total weighted completion time based on the properties of an optimal schedule. Zhang et al. [7] have addressed the total weighted completion time minimization where there are m identical resources available at each time unit. They have provided a greedy algorithm with an approximation ratio 2.

In this paper, we consider scheduling problems to minimize the total stretch in a single machine and parallel machines. The rest of this paper is organized as follows. In Section 2, we provide assumptions and notations and define our models. Some dominant properties are developed. Based on these properties, we propose a polynomial time algorithm for the problems in Section 3. Finally, we provide summary and conclusions in Section 4.

2. **Dominant Properties for Single Machine Scheduling.** Let $J = \{J_1, J_2, \ldots, J_n\}$ be the job set. For job j, let p_j be the processing time of job J_j, r_j its release time, a_j its starting time, C_j its completion time, and F_j its flow time. The stretch of job J_j can be defined as $s_j = (C_j - r_j)/p_j$, equivalently $s_j = F_j/p_j$. In this paper, the release times of all jobs are assumed to be zeros and therefore, $s_j = C_j/p_j$.

Scheduling problems can be denoted by a triplet $\alpha|\beta|\gamma$ [8]. The α field describes machine environments, β processing characteristics and constraints, and γ performance measures. Using this notation, we denote our problems as $1 \,||\, \sum_{j=1}^{n} s_j$ and $Pm \,||\, \sum_{j=1}^{n} s_j$, respectively.

Lemma 2.1. *The stretch of a job can be represented of the starting time and processing time of the job.*

Proof: $s_j = \frac{C_j}{p_j} = \frac{a_j+p_j}{p_j} = \frac{a_j}{p_j} + 1.$ □

Lemma 2.2. *There exist no unforced idle times in an optimal schedule.*

Proof: If a schedule has idle times, the total stretch of the schedule can be reduced by left shift of jobs. □

Theorem 2.1. *The shortest processing time first (SPT) rule is optimal.*

Proof: By contradiction. Suppose a schedule σ, which is not SPT, is optimal. Let job J_i be followed by job J_j such that $p_i > p_j$ (Figure 1(a)). Assume job J_i starts its processing at time t. By interchanging jobs J_i and J_j, a new schedule σ' is obtained in Figure 1(b).

In σ, job J_i starts its processing at time t and is followed by job J_j. While in σ', job J_j starts its processing at time t and is followed by job J_i. All other jobs remain in their original positions. The total stretch of the jobs processed before and after jobs J_i and J_j is not affected by this interchange. Thus, we only consider the difference of the total stretch of jobs J_i and J_j in σ and σ'.

In σ,

$$s_i + s_j = \left(\frac{t}{p_i} + 1\right) + \left(\frac{t + p_i}{p_j} + 1\right) = t\left(\frac{1}{p_i} + \frac{1}{p_j}\right) + \frac{p_i}{p_j} + 2$$

In σ',

$$s_i' + s_j' = \left(\frac{t + p_j}{p_i} + 1\right) + \left(\frac{t}{p_j} + 1\right) = t\left(\frac{1}{p_i} + \frac{1}{p_j}\right) + \frac{p_j}{p_i} + 2$$

It is easily verified that if $p_i > p_j$, the sum of the stretch under σ' is strictly less than under σ. This completes the proof. □

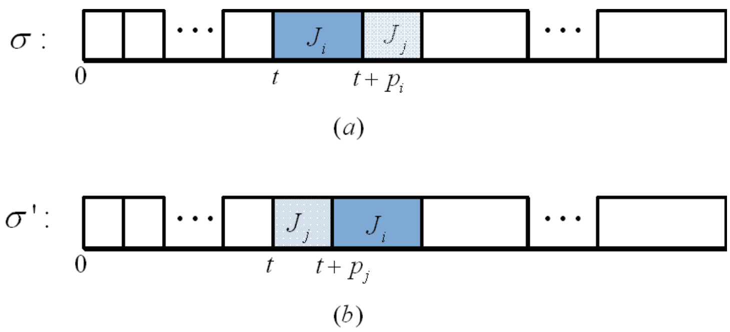


Figure 1. A pairwise interchange of jobs J_i and J_j

3. Polynomial Time Algorithm for Parallel Machine Scheduling. Let the job set J be partitioned into m disjoint sets such as $\{A_1, A_2, A_3, \ldots, A_m\}$, where A_i is the subset of J assigned to machine M_i. A feasible schedule on m parallel machines is $\sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$, where σ_i is a feasible schedule, which is constructed by A_i.

Lemma 3.1. *There exist no unforced idle times on each machine in an optimal schedule.*

Lemma 3.2. *An optimal job sequence on each machine is SPT.*

Lemma 3.3. *If starting time of a job J_j is changed by Δt, the stretch is changed by $\frac{\Delta t}{p_j}$.*

Proof: Suppose that starting time of job J_j is changed from a_j to $a_j + \Delta t$. By Lemma 2.1, the stretch of the job is changed from $\frac{a_j}{p_j} + 1$ to $\frac{a_j+\Delta t}{p_j} + 1$. Therefore, the stretch change is $\frac{\Delta t}{p_j}$. □

Theorem 3.1. *In an optimal schedule, if $p_i < p_j$, then $a_i \le a_j$.*

Proof: By contradiction. Suppose that a schedule σ is optimal, in which $p_i < p_j$ and $a_i > a_j$. Then, jobs J_i and J_j must be assigned on different machines by Lemma 3.2. Let job J_i be assigned on machine M_k and job J_j on machine M_l. Let A and B be the set of jobs that are processed before and after job J_i, respectively. Let C and D be the set of jobs that are processed before and after job J_j, respectively (Figure 2(a)).

The total stretch of the schedule σ is

$$s = \sum\nolimits_{t\in A} s_t + s_i + \sum\nolimits_{t\in B} s_t + \sum\nolimits_{t\in C} s_t + s_j + \sum\nolimits_{t\in D} s_t.$$

Consider two cases: $\sum_{t\in B} \frac{1}{p_t} < \sum_{t\in D} \frac{1}{p_t}$ and $\sum_{t\in B} \frac{1}{p_t} \ge \sum_{t\in D} \frac{1}{p_t}$

Case 1. $\sum_{t\in B} \frac{1}{p_t} < \sum_{t\in D} \frac{1}{p_t}$.

Interchange J_i and J_j (Figure 2(b)). The stretches of A and C are not changed. Then the difference of the total stretch after interchange is calculated below:

$$\begin{aligned}\Delta s &= \Delta s_i + \sum\nolimits_{t\in B} \Delta s_t + \Delta s_j + \sum\nolimits_{t\in D} \Delta s_t \\ &= \frac{a_j - a_i}{p_i} + (p_j - p_i)\sum\nolimits_{t\in B} \frac{1}{p_t} + \frac{a_i - a_j}{p_j} + (p_i - p_j)\sum\nolimits_{t\in D} \frac{1}{p_t} \qquad \text{(by Lemma 3.3)} \\ &= (a_i - a_j)\left(-\frac{1}{p_i} + \frac{1}{p_j}\right) + (-p_i + p_j)\left(\sum\nolimits_{t\in B} \frac{1}{p_t} - \sum\nolimits_{t\in D} \frac{1}{p_t}\right) < 0.\end{aligned}$$

Case 2. $\sum_{t\in B} \frac{1}{p_t} \ge \sum_{t\in D} \frac{1}{p_t}$.

Interchange J_i and B with J_j and D (Figure 2(c)). The stretches of A and C are not changed. Then the difference of the total stretch after interchange is calculated below:

$$\begin{aligned}\Delta s &= \Delta s_i + \sum\nolimits_{t\in B} \Delta s_t + \Delta s_j + \sum\nolimits_{t\in D} \Delta s_t \\ &= \frac{a_j - a_i}{p_i} + (a_j - a_i)\sum\nolimits_{t\in B} \frac{1}{p_t} + \frac{a_i - a_j}{p_j} + (a_i - a_j)\sum\nolimits_{t\in D} \frac{1}{p_t} \\ &= (a_i - a_j)\left(-\frac{1}{p_i} + \frac{1}{p_j}\right) + (a_i - a_j)\left(-\sum\nolimits_{t\in B} \frac{1}{p_t} + \sum\nolimits_{t\in D} \frac{1}{p_t}\right) < 0.\end{aligned}$$

In both cases, the total stretch can be decreased. This completes the proof. □

Definition 3.1. *Let $J = \{J_1, J_2, \ldots, J_n\}$ be a set of jobs, in which jobs are arranged in a nondecreasing order of processing times, that is $p_1 \le p_2 \le \cdots \le p_n$. Construct a job set $\widehat{A_i}$ by assigning job J_j to machine M_i, where $j \equiv i \pmod{m}$, for $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$. Sequence $\widehat{\sigma_i}$ is determined by SPT rule for jobs in $\widehat{A_i}$. Then, the circular SPT schedule is $\widehat{\sigma} = \{\widehat{\sigma_1}, \widehat{\sigma_2}, \ldots, \widehat{\sigma_m}\}$, where $\widehat{\sigma_i} = \left\langle J_i, J_{m+i}, J_{2m+i}, \ldots, J_{\lfloor \frac{n-i}{m} \rfloor m+i} \right\rangle$.*

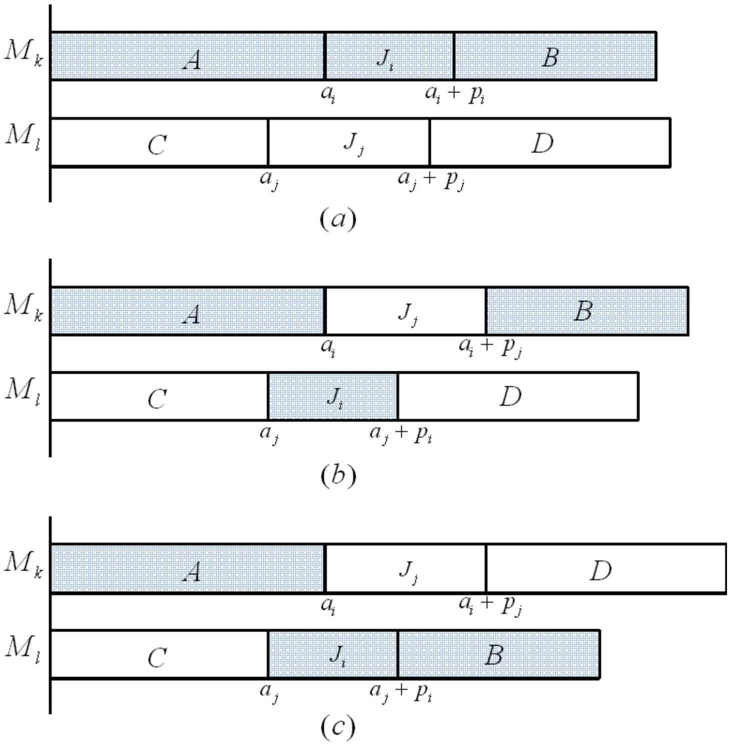


FIGURE 2. Interchanging jobs J_i and J_j

Definition 3.2. *The load l_i of machine M_i is the total job processing times assigned to the machine.*

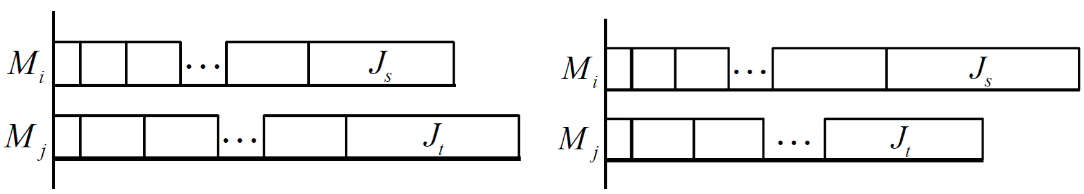
Lemma 3.4. *For the job J_k that is assigned last in the schedule $\widehat{\sigma} = \{\widehat{\sigma_1}, \widehat{\sigma_2}, \ldots, \widehat{\sigma_m}\}$, the difference of two machine loads is less than or equal to the processing time of the last job, i.e., $|l_i - l_j| \leq p_k$, $1 \leq i \neq j \leq m$.*

Proof: Let $1 \leq i < j \leq m$. The difference in number of jobs in two sequences $\widehat{\sigma_i}$ and $\widehat{\sigma_j}$ is not greater than 1, i.e., $0 \leq \left|\widehat{A_i}\right| - \left|\widehat{A_j}\right| \leq 1$. Let the last job in the sequences $\widehat{\sigma_i}$ and $\widehat{\sigma_j}$ be J_s and J_t, respectively. Then, $0 \leq l_j - l_i \leq p_t$ (Figure 3(a)) or $0 \leq l_i - l_j \leq p_s$ (Figure 3(b)). Since $p_s \leq p_k$ and $p_t \leq p_k$, $|l_i - l_j| \leq p_k$. $\square$

Corollary 3.1. *In the schedule $\widehat{\sigma} = \{\widehat{\sigma_1}, \widehat{\sigma_2}, \ldots, \widehat{\sigma_m}\}$, if $l_i = l_j$, then $\left|\widehat{A_i}\right| = \left|\widehat{A_j}\right|$ and $p_{i,[t]} = p_{j,[t]}$.*

Theorem 3.2. *The schedule $\widehat{\sigma} = \{\widehat{\sigma_1}, \widehat{\sigma_2}, \ldots, \widehat{\sigma_m}\}$ is optimal for $Pm||\sum_j s_j$.*

Proof: Let $J_{i,[k]}$ be the kth job in the sequence $\widehat{\sigma_i}$ on machine i. Let $p_{i,[k]}$ and $a_{i,[k]}$ be the processing time and starting time of job $J_{i,[k]}$, respectively. Interchange $J_{i,[u]}$ and $J_{j,[v]}$. If $p_{i,[u]} = p_{j,[v]}$, the total stretch is not changed. Therefore, we assume that $p_{i,[u]} \neq p_{j,[v]}$.



(a) $\left|\widehat{A_i}\right| = \left|\widehat{A_j}\right|$ (b) $\left|\widehat{A_i}\right| = \left|\widehat{A_j}\right| + 1$

FIGURE 3. Job sequences $\widehat{\sigma_i}$ and $\widehat{\sigma_j}$

We claim that the total stretch cannot be decreased by this interchange for the cases $u = v$ and $u \neq v$.

Case 1 ($u = v$) Let $i < j$. Then $p_{i,[u]} < p_{j,[v]}$ and $a_{i,[u]} \leq a_{j,[v]}$. By interchanging $J_{i,[u]}$ with $J_{j,[v]}$, the change of the total stretch is calculated as below:

$$\begin{aligned}\Delta s &= \left(\Delta s_{i,[u]} + \Delta s_{j,[v]}\right) + \left(\sum_{t=u+1}^{|\mathrm{A}_i|} \Delta s_{i,[t]} + \sum_{t=v+1}^{|\mathrm{A}_j|} \Delta s_{j,[t]}\right) \\ &= \left(a_{i,[u]} - a_{j,[v]}\right)\left(-\frac{1}{p_{i,[u]}} + \frac{1}{p_{j,[v]}}\right) + \left(p_{i,[u]} - p_{j,[v]}\right)\left(-\sum_{t=u+1}^{|\mathrm{A}_i|} \frac{1}{p_{i,[t]}} + \sum_{t=v+1}^{|\mathrm{A}_j|} \frac{1}{p_{j,[t]}}\right)\end{aligned}$$

Since $-\sum_{t=u+1}^{|\mathrm{A}_i|} \frac{1}{p_{i,[t]}} + \sum_{t=v+1}^{|\mathrm{A}_j|} \frac{1}{p_{j,[t]}} \leq -\sum_{t=u+1}^{|\mathrm{A}_i|} \frac{1}{p_{i,[t]}} + \sum_{t=v+1}^{|\mathrm{A}_j|} \frac{1}{p_{j,[t]}} \leq 0$ and $-\frac{1}{p_{i,[u]}} + \frac{1}{p_{j,[v]}} < 0$, $\Delta s \geq 0$.

Case 2 ($u \neq v$) Without loss of generality, $u > v$. Then $p_{i,[u]} > p_{j,[v]}$ and $a_{i,[u]} \geq a_{j,[v]}$. It is evident that $u = v + 1$, since if $u - v \geq 2$, two sequences obtained by interchanging $J_{i,[u]}$ and $J_{j,[v]}$ would not be SPT. Then, by interchanging $J_{i,[u]}$ with $J_{j,[v]}$, the change of the total stretch is calculated as below:

$$\begin{aligned}\Delta s &= \left(\Delta s_{i,[u]} + \Delta s_{j,[v]}\right) + \left(\sum_{t=u+1}^{|\mathrm{A}_i|} \Delta s_{i,[t]} + \sum_{t=v+1}^{|\mathrm{A}_j|} \Delta s_{j,[t]}\right) \\ &= \left(a_{i,[u]} - a_{j,[v]}\right)\left(-\frac{1}{p_{i,[u]}} + \frac{1}{p_{j,[v]}}\right) + \left(p_{i,[u]} - p_{j,[v]}\right)\left(-\sum_{t=u+1}^{|\mathrm{A}_i|} \frac{1}{p_{i,[t]}} + \sum_{t=v+1}^{|\mathrm{A}_j|} \frac{1}{p_{j,[t]}}\right)\end{aligned}$$

If $i < j$, $-\sum_{t=u+1}^{|\mathrm{A}_i|} \frac{1}{p_{i,[t]}} + \sum_{t=v+1}^{|\mathrm{A}_j|} \frac{1}{p_{j,[t]}} \geq -\sum_{t=u+1}^{|\mathrm{A}_i|} \frac{1}{p_{i,[t]}} + \sum_{t=v+1}^{|\mathrm{A}_i|} \frac{1}{p_{j,[t]}} \geq 0$.

If $i > j$, $-\sum_{t=u+1}^{|\mathrm{A}_i|} \frac{1}{p_{i,[t]}} + \sum_{t=v+1}^{|\mathrm{A}_j|} \frac{1}{p_{j,[t]}} \geq -\sum_{t=u+1}^{|\mathrm{A}_j|} \frac{1}{p_{i,[t]}} + \sum_{t=v+1}^{|\mathrm{A}_j|} \frac{1}{p_{j,[t]}} \geq 0$.

Therefore, $\Delta s \geq 0$. This completes the proof. □

Algorithm *SPT-LS* for parallel machines

Step 1. Construct a list of jobs by the SPT-rule.

Step 2. (List scheduling) Assign jobs to the least loaded machine according to a list of jobs, where jobs are arranged by the specific criteria in the list (ties are broken arbitrarily).

Numerical Example. Consider three parallel identical machines with ten jobs. The processing times of jobs are provided in Table 1.

Table 1. Three machines with ten jobs

Job	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}
p_j	5	4	5	1	2	2	6	10	2	7

First, construct a list of jobs according to SPT-rule such as

$$List = \{J_4, J_5, J_6, J_9, J_2, J_1, J_3, J_7, J_{10}, J_8\}.$$

Second, make an assignment by the list scheduling.

Then, an optimal schedule is shown in the Gantt chart in Figure 4.

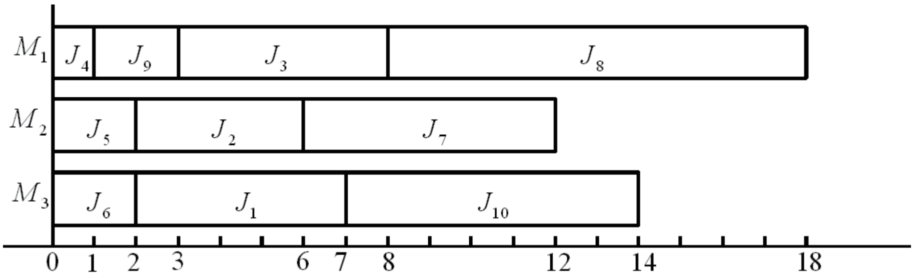


FIGURE 4. Optimal schedule

4. **Conclusion.** We consider scheduling problems in a single machine and parallel machines, in which the objective is to minimize the total stretch. The objective is a special case of the total weighted completion times, which is known to be NP-hard. We provide some dominant properties and propose a polynomial time algorithm for the problems. The problem complexity of general parallel machine problems with release dates would be studied for future research.

REFERENCES

[1] H. C. Hwang, S. Y. Chang and K. Lee, Parallel machine scheduling under a grade of service provision, *Computers & Operations Research*, vol.31, pp.2055-2061, 2004.
[2] C. A. Glass and H. Kellerer, Parallel machine scheduling with job assignment restrictions, *Naval Research Logistics*, vol.54, pp.250-257, 2007.
[3] Y. Azar, J. Naor and R. Rom, The competitiveness of on-line assignments, *Journal of Algorithms*, vol.18, no.2, pp.221-237, 1995.
[4] A. Legrand, A. Su and F. Vivien, Minimizing the stretch when scheduling flows of divisible requests, *Journal of Scheduling*, vol.11, pp.381-404, 2008.
[5] S. Muthukrishnan, R. Rajaraman, A. Shaheen and J. Gehrke, Online scheduling to minimize average stretch, *SIAM Journal on Computing*, vol.34, no.2, pp.433-452, 2005.
[6] J. Xu and R. Nagi, Identical parallel machine scheduling to minimise makespan and total weighted completion time: A column generation approach, *International Journal of Production Research*, vol.51, nos.23-24, pp.7091-7104, 2013.
[7] Q. Zhang, W. Wu and M. Li, Minimizing the total weighted completion time of fully parallel jobs with integer parallel units, *Theoretical Computer Science*, vol.57, pp.34-40, 2013.
[8] M. Pinedo, *Scheduling: Theory, Algorithms and Systems*, 4th Edition, Prentice-Hall, Englewood Cliffs, NJ, 2012.