

## SINGLE MACHINE GROUP SCHEDULING PROBLEM WITH MAKESPAN OBJECTIVE AND A PROPORTIONAL LINEAR SHORTENING

JI-BO WANG<sup>1,\*</sup>, XUE JIA<sup>1</sup>, JIA-XUAN YAN<sup>1</sup>, SI-HAN WANG<sup>1</sup> AND JIN QIAN<sup>2</sup>

**Abstract.** This paper considers a group scheduling problem with shorten (*i.e.*, a proportional linear shortening) job processing times and ready times on a single machine. If the jobs are in the same group, they have independent and constant ready times. The setup time of a group is independent constant setup time between each group. The goal is to determine the optimal group sequence and the job sequence within the groups such that the makespan (*i.e.*, the maximum completion time) is minimized. For the general case of the problem, an initial heuristic algorithm (an upper bound) and some lower bounds are proposed, and then a branch-and-bound algorithm can be developed to solve the problem.

**Mathematics Subject Classification.** 90B35.

Received October 11, 2021. Accepted May 14, 2022.

### 1. INTRODUCTION

In the classic scheduling, the processing time of a job is generally a specified constant. But in some actual situations, the processing time of some job is variable, which will transform around the expected value with the start time of the job. For instance, in a fire accident, the disaster situation will soar over the time. When the start time of fire extinguishing is delayed, the time of fire fighting will be prolonged and the loss of resources will increase, this is the phenomenon of deterioration effects [1, 5, 9]. On the other hand, the phenomenon of “learning effects” can be described by a non-increasing function of start time (or position) dependent function. For example, in a radar detection system, the distance between radar and target gradually decreases as the target approaches. Similarly, the time required to detect the target will be gradually shortened with the time elapsed by the radar electromagnetic wave. Extensive surveys on position dependent learning effects can be found in Biskup [4] and Azzouz *et al.* [2], and an extensive book on time dependent learning effects (*i.e.*, shortening job processing times) can be found in Gawiejnowicz [7].

Group technology is a technique that takes advantage of the similarities between things and groups into job families according to specific criteria. Due to the similarity of the jobs in one group, the processing efficiency of

---

*Keywords.* scheduling, group technology, shortening job processing times, branch-and-bound algorithm.

<sup>1</sup> School of Science, Shenyang Aerospace University, Shenyang 110136, P.R. China.

<sup>2</sup> Department of Mathematical Sciences, Northeastern University, Shenyang 110819, P.R. China.

\*Corresponding author: [wangjibo75@163.com](mailto:wangjibo75@163.com)

a job will increase and processing time will be shortened. In many countries, group technology is applied to the field of computer technology and automation technology to develop flexible manufacturing systems and provide multi-variety, small and medium-sized and highly automated production, which has great application value in actual production. Therefore, such as Potts and Van Wassenhove [19], Keshavarz *et al.* [11], Neufeld *et al.* [18], Ji and Li [10] and Lu *et al.* [17], many scholars had done lots of research on group technology. Wang *et al.* [24], Wang *et al.* [25] and Xu *et al.* [30] introduced a simple deterioration function into the single machine grouping problem. Bai *et al.* [3] studied the problem of single group scheduling with a general learning effect of deteriorating conditions. They proved that minimizing makespan was polynomial solvability in this case. The resource allocation single machine scheduling problem with the deteriorating jobs was considered by Wang *et al.* [26]. Wang and Wang [21] investigated the single machine scheduling problem with group technology and ready times, where the setup (installation) time and job processing time of each group are proportional linear deterioration functions of their start time. For the makespan minimization, they proved that it could be solved in polynomial time. Liu *et al.* [14] considered the single machine group scheduling problem with simple linear deterioration and ready times. Under independent setup time, for the makespan minimization, they proposed heuristic and branch-and-bound algorithms to solve the problem.

To our knowledge, the scheduling problems that are related to time dependent processing times are widely reflected in real production, and many scholars have studied in this field (see [7, 12, 16]). The model with time dependent learning effects (*i.e.*, shortening job processing times) is also widely reflected in the real scene (see [7]). Ho *et al.* [8] first proposed the scheduling problem of which the job processing time was a decreasing function of its starting time. Wang and Xia [22] and Wang *et al.* [27] studied scheduling with linear shortening job processing times. For the single machine problem, Wang and Xia [22] proved that some regular objective function minimizations can be solved in polynomial time respectively. For the three-machine flow shop problem of minimizing makespan, Wang *et al.* [27] proposed a branch-and-bound algorithm and a heuristic algorithm. Wang *et al.* [23] considered single machine group scheduling with shortening job processing times. Under independent setup of groups, they proved that the makespan and the total completion time minimizations can be solved in polynomial time. Wang and Sun [20] concentrated on single machine group scheduling problems with shortening job processing times. Under group setup times and job processing times are a decreasing function of their starting time, they illustrated that the makespan minimization problem can be solved in polynomial time. They also proved that the total weighted completion time minimization problem remains polynomially solvable under the proportional shortening job processing times. Lu *et al.* [15] studied the single machine group scheduling problem with proportional shortening job processing times and release dates. Under the group setup times and job processing times are both shortening job processing times, they proved that the makespan minimization was polynomial time solvable.

Recently, Wang *et al.* [28] examined the single machine group scheduling problem with proportional shortening job processing times and release dates. When the setup time of a group is fixed, they proved that the general case of the problem was polynomial time solvable problem when the number of groups had been determined; they also showed that some special cases of the makespan minimization can be solved in polynomial time respectively. Compared to many classic scheduling problems, most of the scheduling problems in the real world are not polynomial solvable and not special, therefore, in this paper, we continue the work of Wang *et al.* [28], we study the general case of single machine group scheduling problem with release dates and a proportional linear shortening processing time. The main contributions of this article are summarized as follows: first, this article presents an attempt to study the makespan minimisation single machine scheduling problem with a proportional linear shortening. Second, we develop a heuristic algorithm and a branch-and-bound algorithm to solve the problem. Finally, the results of numerical simulations are given to show that our algorithms can generate effective and efficient schedules.

The remaining part of this paper is organized as follows. Section 2 presents the problem formulation. Section 3 studies the scheduling problem under general conditions. And Section 4 summarizes conclusion in this study and future research directions.

## 2. PROBLEM FORMULATION

In the actual situation we are considering, the actual processing time is less than the standard processing time of the job. Generally speaking, the actual processing time of a job is related to three factors that are the standard processing time, learning factor and start-up time. On the basis of Wang *et al.* [28], we make the assumption that the actual machine processing time of a job is a linear decreasing function of its start time. The expression of the model is given as follows.

Under the group technology, a total of  $n$  jobs are divided into  $f$  groups:  $G_1, G_2, \dots, G_f$ . Assume that each group of  $G_i$  ( $1 \leq i \leq f$ ) contains  $n_i$  jobs  $\{J_{i,1}, J_{i,2}, \dots, J_{i,n_i}\}$ . A job is continuously processed on the machine. Only one job can be processed on the machine at the same time, and the machining process can not be interrupted. Let  $J_{i,j}$  denote the  $j$ th job in group  $G_i$ ,  $r_{i,j} \geq 0$  ( $p_{i,j} > 0$ ) denote the ready time (actual processing time) of job  $J_{i,j}$ ,  $i = 1, 2, \dots, f; j = 1, 2, \dots, n_i$ ,  $s_i$  ( $s_i > 0, i = 1, 2, \dots, f$ ) denote the separate installation time for each group. We assume that the jobs which are in the same group must be processed without being interrupted. For the linear shortening job processing times, the actual processing time of the job is

$$p_{i,j} = a_{i,j} - b_{i,j}t, \quad (2.1)$$

where  $a_{i,j}$  is the normal processing time of job  $J_{i,j}$  (the processing time without any shortening (learning) effects),  $b_{i,j}$  is the shortening rate of job  $J_{i,j}$  (the positive compression rate of unit time), and  $t$  is its start time. Assume that the shortening rate meets the following two conditions:

$$0 < b_{i,j} < 1, \quad i = 1, 2, \dots, f; \quad j = 1, 2, \dots, n_i, \quad (2.2)$$

and

$$b_{i,j} \left( \sum_{i=1}^f (s_i + r_{i,\max}) + \sum_{i=1}^f \sum_{h=1}^{n_i} a_{i,h} - a_{i,j} \right) < a_{i,j}, \quad (2.3)$$

where  $r_{i,\max} = \max\{r_{i,h} | h = 1, 2, \dots, n_i\}$  ( $i = 1, 2, \dots, f$ ). In this paper, we consider a special case, *i.e.*,  $b_{i,j} = \beta a_{i,j}$ , and

$$p_{i,j} = a_{i,j} (1 - \beta t), \quad (2.4)$$

where  $\beta \geq 0$  is a constant. Conditions (2.2) and (2.3) meet the following two conditions:

$$0 < \beta a_{i,j} < 1, \quad i = 1, 2, \dots, f; \quad j = 1, 2, \dots, n_i, \quad (2.5)$$

and

$$\beta \left( \sum_{i=1}^f (s_i + r_{i,\max}) + \sum_{i=1}^f \sum_{j=1}^{n_i} a_{i,j} - a_{\min} \right) < 1, \quad (2.6)$$

where  $a_{\min} = \min\{a_{i,j} | i = 1, 2, \dots, f; j = 1, 2, \dots, n_i\}$ . As in Wang *et al.* [23] and Wang *et al.* [28], conditions (2.1) and (2.4) illustrate the relationship among the actual processing time, the start time and normal processing time; conditions (2.2) and (2.5) not only guarantee that the processing time of all the jobs in each feasible order is positive, but also guarantees the formulas (2.1) and (2.4) for reasonable schedules.

Let  $C_{i,j}$  be the completion time of job  $J_{i,j}$ ,  $C_{\max} = \max\{C_{i,j} | i = 1, 2, \dots, f, j = 1, 2, \dots, n_i\}$  be the makespan of a permutation. Our goal is to minimize the makespan, *i.e.*, the maximal completion time of all jobs. This problem has two influencing factors. One of the factors is to determine the optimal job sequence  $S_i^*$  within the group  $G_i$ , and the other is to determine the optimal sequence  $S_G^*$  between the groups. The above problem is defined by the three-field notation (see [7]) as

$$1 | r_{i,j}, p_{i,j} = a_{i,j} (1 - \beta t), s_i, \text{GT} | C_{\max},$$

where GT denotes the group technology assumption. Wang *et al.* [28] considered the problem  $1 | r_{i,j}, p_{i,j} = a_{i,j} (1 - \beta t), s_i, \text{GT} | C_{\max}$ , they proved that some special cases of the problem can be solved in polynomial time.

Lu *et al.* [15] studied the problem  $1|r_{i,j}, p_{i,j} = a_{i,j}(1 - \beta t), s_i = c_i(1 - \beta t), GT|C_{\max}$  with the variable setup times, where  $c_i > 0$  is the normal setup time of the group  $G_i$ , they showed that this problem can be solved in  $O(n \log n)$  time.

### 3. GENERAL CASE

For the general case of  $1|r_{i,j}, p_{i,j} = a_{i,j}(1 - \beta t), s_i, GT|C_{\max}$ , we conjecture that this problem is NP-hard, thus, the branch-and-bound and heuristic algorithms might be a good way to obtain the optimal (near-optimal) solutions. From Wang *et al.* [28], an optimal job sequence  $S_i^*$  within the group  $G_i$  can be obtained by ordering the jobs in nondecreasing order of  $r_{i,j}$  ( $i = 1, 2, \dots, f; j = 1, 2, \dots, n_i$ , see [28], Thm. 1).

#### 3.1. A heuristic algorithm (An upper bound)

**Lemma 3.1.** *If  $s_i, \gamma_i \in R$  ( $i = 1, 2, \dots, f$ ), then the sum  $s_1\gamma_1\gamma_2 \cdots \gamma_{f-1}\gamma_f + s_2\gamma_2\gamma_3 \cdots \gamma_{f-1}\gamma_f + \cdots + s_{f-1}\gamma_{f-1}\gamma_f + s_f\gamma_f$  ( $0 < \gamma_i < 1, i = 1, 2, \dots, f$ ) is minimized if the indices are ordered by nonincreasing values of  $\frac{s_i\gamma_i}{1-\gamma_i}$  ratios ( $i = 1, 2, \dots, f$ ), i.e.,  $\frac{s_1\gamma_1}{1-\gamma_1} \geq \frac{s_2\gamma_2}{1-\gamma_2} \geq \cdots \geq \frac{s_f\gamma_f}{1-\gamma_f}$ .*

*Proof.* By using pairwise element interchanging technique. □

From the Wang *et al.* [28], and the heuristic of Framinan and Leisten [6], the following heuristic algorithm can be given.

**Algorithm 1** (The heuristic algorithm).

**Step 1.** For group  $G_i$  ( $i = 1, 2, \dots, f$ ), by ordering the jobs in non-decreasing order of  $r_{i,j}$ , i.e.,  $r_{i,(1)} \leq r_{i,(2)} \leq \cdots \leq r_{i,(n_i)}$ ,  $i = 1, 2, \dots, f$ .

**Step 2.**  $(r_{i,(\theta_i)} - \frac{1}{\beta}) \prod_{l=\theta_i}^{n_i} (1 - \beta a_{i,(l)}) = \max\{(r_{i,(1)} - \frac{1}{\beta}) \prod_{l=1}^{n_i} (1 - \beta a_{i,(l)}), (r_{i,(2)} - \frac{1}{\beta}) \prod_{l=2}^{n_i} (1 - \beta a_{i,(l)}), \dots, (r_{i,(n_i)} - \frac{1}{\beta})(1 - \beta a_{i,(n_i)})\}$ , calculate  $\theta_i$  and  $\rho(G_i) = \frac{r_{i,(\theta_i)} - \frac{1}{\beta}}{\prod_{l=1}^{\theta_i-1} (1 - \beta a_{i,(l)})}$ ,  $i = 1, 2, \dots, f$ .

**Step 3.1.** Groups are scheduled by the nondecreasing order of  $s_i$ ,  $i = 1, 2, \dots, f$ .

**Step 3.2.** Groups are scheduled by the nondecreasing order of  $\rho(G_i) = \frac{r_{i,(\theta_i)} - \frac{1}{\beta}}{\prod_{l=\theta_i}^{n_i} (1 - \beta a_{i,(l)})}$ , ( $i = 1, 2, \dots, f$ ).

**Step 3.3.** Groups are scheduled by the nonincreasing order of  $s_i \prod_{l=1}^{n_i} (1 - \beta a_{i,(l)})$ , ( $i = 1, 2, \dots, f$ ).

**Step 3.4.** Groups are scheduled by the nonincreasing order of  $\prod_{l=1}^{n_i} (1 - \beta a_{i,(l)})$ , ( $i = 1, 2, \dots, f$ ).

**Step 4.** Choose the better solution from Steps 3.1–3.4.

**Step 5.** Pick the two groups from the first and second position of the list of Step 4, and find the best sequence for these two groups by calculating  $C_{\max}$  for the two possible sequences. Do not change the relative positions of these two groups with respect to each other in the remaining steps of the algorithm. Set  $i = 3$ .

**Step 6.** Pick the group in the  $i$ th position of the list generated in Step 4 and find the best sequence by placing it at all possible  $i$  positions in the partial sequence found in the previous step, without changing the relative positions to each other of the already assigned groups. The number of enumerations at this step equals  $i$ .

**Step 7.** For the list generated in step 6, select two groups from the previous  $i$  groups and only swap the positions of the two groups. By traversing all cases of the two selectable groups, the two groups of the best swap result are found by calculating  $C_{\max}$ , and then the best sequence is updated. The number of enumerations at this step equals  $\frac{i*(i-1)}{2}$ .

**Step 8.** If  $i = f$ , STOP; otherwise set  $i = i + 1$  and go to Step 6.

**3.2. A lower bound**

Let  $S_G = (PS_G, US_G)$  be a complete sequence, the first  $k$  groups be the scheduled part (*i.e.*, the part  $PS_G$ ), and the unscheduled groups have  $n - k$  groups (*i.e.*, the part  $US_G$ ). By definition [28], the completion time of the  $(k + 1)$ th group is

$$\begin{aligned}
 &C_{[k+1],(n_{[k+1]})}(S_G) \\
 &= \max \left\{ \left( C_{k,(n_k)} + s_{[k+1]} - \frac{1}{\beta} \right) \prod_{l=1}^{n_{[k+1]}} (1 - \beta a_{[k+1],(l)}) , \left( r_{[k+1],(1)} - \frac{1}{\beta} \right) \prod_{l=1}^{n_{[k+1]}} (1 - \beta a_{[k+1],(l)}) , \right. \\
 &\quad \left. \left( r_{[k+1],(2)} - \frac{1}{\beta} \right) \prod_{l=1}^{n_{[k+1]}} (1 - \beta a_{[k+1],(l)}) , \dots , \left( r_{[k+1],(n_{[k+1]})} - \frac{1}{\beta} \right) \prod_{l=1}^{n_{[k+1]}} (1 - \beta a_{[k+1],(l)}) \right\} + \frac{1}{\beta}. \quad (3.1)
 \end{aligned}$$

Using equation (3.1), considering the completion time and setup time, the following two specific calculation methods for the lower bound are given.

*3.2.1. Lower bound 1*

From equation (3.1), we have

$$C_{[k+1],(n_{[k+1]})}(S_G) \geq \left( C_{k,(n_k)} + s_{[k+1]} - \frac{1}{\beta} \right) \prod_{l=1}^{n_{[k+1]}} (1 - \beta a_{[k+1],(l)}) + \frac{1}{\beta}.$$

Similarly,

$$\begin{aligned}
 &C_{[k+2],(n_{[k+2]})}(S_G) \\
 &\geq \left( C_{[k+1],n_{[k+1]}} + s_{[k+2]} - \frac{1}{\beta} \right) \prod_{l=1}^{n_{[k+2]}} (1 - \beta a_{[k+2],(l)}) + \frac{1}{\beta} \\
 &\geq \left( C_{k,(n_k)} - \frac{1}{\beta} \right) \prod_{l=1}^{n_{[k+1]}} (1 - \beta a_{[k+1],(l)}) \prod_{l=1}^{n_{[k+2]}} (1 - \beta a_{[k+2],(l)}) + s_{[k+1]} \prod_{l=1}^{n_{[k+1]}} (1 - \beta a_{[k+1],(l)}) \\
 &\quad \times \prod_{l=1}^{n_{[k+2]}} (1 - \beta a_{[k+2],(l)}) + s_{[k+2]} \prod_{l=1}^{n_{[k+2]}} (1 - \beta a_{[k+2],(l)}) + \frac{1}{\beta}
 \end{aligned}$$

.....

$$\begin{aligned}
 &C_{[f],(n_{[f]})}(S_G) \\
 &\geq \left( C_{k,(n_k)} - \frac{1}{\beta} \right) \prod_{l=1}^{n_{[k+1]}} (1 - \beta a_{[k+1],(l)}) \prod_{l=1}^{n_{[k+2]}} (1 - \beta a_{[k+2],(l)}) \cdots \prod_{l=1}^{n_{[f]}} (1 - \beta a_{[f],(l)}) \\
 &\quad + s_{[k+1]} \prod_{l=1}^{n_{[k+1]}} (1 - \beta a_{[k+1],(l)}) \prod_{l=1}^{n_{[k+2]}} (1 - \beta a_{[k+2],(l)}) \cdots \prod_{l=1}^{n_{[f]}} (1 - \beta a_{[f],(l)}) \\
 &\quad + s_{[k+2]} \prod_{l=1}^{n_{[k+2]}} (1 - \beta a_{[k+2],(l)}) \cdots \prod_{l=1}^{n_{[f]}} (1 - \beta a_{[f],(l)}) + \cdots + s_{[f]} \prod_{l=1}^{n_{[f]}} (1 - \beta a_{[f],(l)}) + \frac{1}{\beta} \\
 &\geq \left( C_{k,(n_k)} - \frac{1}{\beta} \right) \prod_{i=k+1}^f \prod_{l=1}^{n_{[i]}} (1 - \beta a_{[i],(l)}) + s_{[k+1]} \prod_{i=k+1}^f \prod_{l=1}^{n_{[i]}} (1 - \beta a_{[i],(l)})
 \end{aligned}$$

$$+ s_{[k+2]} \prod_{i=k+2}^f \prod_{l=1}^{n_{[i]}} (1 - \beta a_{[i],(l)}) + \dots + s_{[f]} \prod_{l=1}^{n_{[f]}} (1 - \beta a_{[f],(l)}) + \frac{1}{\beta}.$$

Therefore, the maximum completion time of the sequence  $S_G$  satisfies the condition

$$C_{\max}(S_G) \geq \left( C_{k,(n_k)} - \frac{1}{\beta} \right) \prod_{i=k+1}^f \prod_{l=1}^{n_{[i]}} (1 - \beta a_{[i],(l)}) + s_{[k+1]} \prod_{i=k+1}^f \prod_{l=1}^{n_{[i]}} (1 - \beta a_{[i],(l)}) + s_{[k+2]} \prod_{i=k+2}^f \prod_{l=1}^{n_{[i]}} (1 - \beta a_{[i],(l)}) + \dots + s_{[f]} \prod_{l=1}^{n_{[f]}} (1 - \beta a_{[f],(l)}) + \frac{1}{\beta}. \tag{3.2}$$

Obviously, term  $\left( C_{k,(n_k)} - \frac{1}{\beta} \right) \prod_{i=k+1}^f \prod_{l=1}^{n_{[i]}} (1 - \beta a_{[i],(l)})$  is a fixed number ( $k$  is a given number), from the Wang *et al.* [24], the term  $s_{[k+1]} \prod_{i=k+1}^f \prod_{l=1}^{n_{[i]}} (1 - \beta a_{[i],(l)}) + s_{[k+2]} \prod_{i=k+2}^f \prod_{l=1}^{n_{[i]}} (1 - \beta a_{[i],(l)}) + \dots + s_{[f]} \prod_{l=1}^{n_{[f]}} (1 - \beta a_{[f],(l)})$  is minimized in which the unscheduled groups are in the non-increasing order of  $\frac{s_i \prod_{l=1}^{n_i} (1 - \beta a_{i,(l)})}{1 - \prod_{l=1}^{n_i} (1 - \beta a_{i,(l)})}$ ,  $i = k + 1, k + 2, \dots, f$ . Hence the first lower bound can be obtained as follows:

$$\begin{aligned} \overline{LB}_1 = & \left( C_{k,(n_k)} - \frac{1}{\beta} \right) \prod_{i=k+1}^f \prod_{l=1}^{n_i} (1 - \beta a_{i,(l)}) + s_{\langle k+1 \rangle} \prod_{i=k+1}^f \prod_{l=1}^{n_{\langle i \rangle}} (1 - \beta a_{\langle i \rangle,(l)}) \\ & + s_{\langle k+2 \rangle} \prod_{i=k+2}^f \prod_{l=1}^{n_{\langle i \rangle}} (1 - \beta a_{\langle i \rangle,(l)}) + \dots + s_{\langle f \rangle} \prod_{l=1}^{n_{\langle f \rangle}} (1 - \beta a_{\langle f \rangle,(l)}) + \frac{1}{\beta}, \end{aligned} \tag{3.3}$$

where  $\frac{s_{\langle k+1 \rangle} \prod_{l=1}^{n_{\langle k+1 \rangle}} (1 - \beta a_{\langle k+1 \rangle,(l)})}{1 - \prod_{l=1}^{n_{\langle k+1 \rangle}} (1 - \beta a_{\langle k+1 \rangle,(l)})} \geq \frac{s_{\langle k+2 \rangle} \prod_{l=1}^{n_{\langle k+2 \rangle}} (1 - \beta a_{\langle k+2 \rangle,(l)})}{1 - \prod_{l=1}^{n_{\langle k+2 \rangle}} (1 - \beta a_{\langle k+2 \rangle,(l)})} \geq \dots \geq \frac{s_{\langle f \rangle} \prod_{l=1}^{n_{\langle f \rangle}} (1 - \beta a_{\langle f \rangle,(l)})}{1 - \prod_{l=1}^{n_{\langle f \rangle}} (1 - \beta a_{\langle f \rangle,(l)})}$ .

3.2.2. Lower bound 2

When considering the setup time of the job is long, the calculation process of the lower bound only considers the completion time of the job which is not rigorous. So considering the setup time of the job, the second lower bound of the problem can be obtained. According to formula (3.1),

$$C_{[k+1],(n_{k+1})}(S_G) \geq \left( r_{[k+1],(\theta_{k+1})} - \frac{1}{\beta} \right) \prod_{l=\theta_{k+1}}^{n_{[k+1]}} (1 - \beta a_{[k+1],(l)}) + \frac{1}{\beta},$$

where  $\theta_{k+1} \in \{1, 2, \dots, n_{k+1}\}$ .

Similarly,

$$C_{[k+2],(n_{k+2})}(S_G) \geq \left( r_{[k+2],(\theta_{k+2})} - \frac{1}{\beta} \right) \prod_{l=\theta_{k+2}}^{n_{[k+2]}} (1 - \beta a_{[k+2],(l)}) + \frac{1}{\beta},$$

where  $\theta_{k+2} \in \{1, 2, \dots, n_{k+2}\}$ .

Similarly, the completion time of the last job in the  $f$ th group in the sequence  $S_G$  is

$$C_{[f],(n_f)}(\pi) \geq \left( r_{[f],(\theta_f)} - \frac{1}{\beta} \right) \prod_{l=\theta_f}^{n_{[f]}} (1 - \beta a_{[f],(l)}) + \frac{1}{\beta},$$

where  $\theta_f \in \{1, 2, \dots, n_f\}$ .

Therefore, the maximum completion time of the sequence  $S_G$  satisfies the condition

$$C_{\max}(S_G) \geq \left(r_{[f],(\theta_f)} - \frac{1}{\beta}\right) \prod_{l=\theta_f}^{n_{[f]}} (1 - \beta a_{[f],(l)}) + \frac{1}{\beta}.$$

Consequently, we obtain the second lower bound

$$\begin{aligned} \overline{LB}_2 = \max & \left\{ \left(r_{k+1,(\theta_{k+1})} - \frac{1}{\beta}\right) \prod_{l=\theta_{k+1}}^{n_{k+1}} (1 - \beta a_{k+1,(l)}), \left(r_{k+2,(\theta_{k+2})} - \frac{1}{\beta}\right) \prod_{l=\theta_{k+2}}^{n_{k+2}} (1 - \beta a_{k+2,(l)}), \right. \\ & \left. \dots, \left(r_{f,(\theta_f)} - \frac{1}{\beta}\right) \prod_{l=\theta_f}^{n_f} (1 - \beta a_{f,(l)}) \right\} + \frac{1}{\beta}, \end{aligned} \tag{3.4}$$

where  $(r_{i,(\theta_i)} - \frac{1}{\beta}) \prod_{l=\theta_i}^{n_i} (1 - \beta a_{i,(l)}) = \max\{(r_{i,(1)} - \frac{1}{\beta}) \prod_{l=1}^{n_i} (1 - \beta a_{i,(l)}), (r_{i,(2)} - \frac{1}{\beta}) \prod_{l=2}^{n_i} (1 - \beta a_{i,(l)}), \dots, (r_{i,(n_i)} - \frac{1}{\beta}) \prod_{l=n_i}^{n_i} (1 - \beta a_{i,(n_i)})\}$ ,  $\theta_i \in \{1, 2, \dots, n_i\}$ .

In order to make the lower bound tighter, we choose the maximum value of equations (3.3) and (3.4) as the lower bound, *i.e.*,

$$\overline{LB} = \max \{ \overline{LB}_1, \overline{LB}_2 \}. \tag{3.5}$$

### 3.3. Branch-and-bound algorithm

**Algorithm 2. Step 1.** Determine an initial solution (*i.e.*, the upper bound) by using Algorithm 1.

**Step 2.** A depth-first search is used in the branching procedure until all the nodes are explored or eliminated. Start the assignment of groups at the beginning of a sequence and move forward one step at a time.

**Step 3.** In the  $k$ th level node, the first  $k$  positions are occupied by  $k$  specific groups. Select one of the remaining  $f - k$  groups for the node at level  $k + 1$ .

**Step 4.** Calculate the lower bound of the unfathomed partial sequences by using (3.5). During the search, any unfathomed branch that has a lower bound which is larger than or equal to the initial solution is fathomed, *i.e.*, eliminated from further consideration. If the value of the completed sequence is less than the initial solution, use it as the new initial solution. Otherwise, eliminate it.

**Step 5.** Continue to search all the nodes, and the remaining initial solution is optimal.

### 3.4. Computational experiments

Algorithms 1 and 2 (the Branch-and-bound algorithm) were programmed in VC++ 6.0 and tests were run on CPU Intel® Core i5 2.5 GHz and 8GB RAM for several sizes of the problem ( $n = 1000, 1250, 1500, 1750, 2000$ ,  $f = 250, 275, 300, 325, 350$  and each group must contain at least one job). The following parameters are randomly generated:

- (1)  $\beta = 0.00005$ ;
- (2)  $r_{ij}$  are uniformly distributed over  $[1, 100]$ ;
- (3)  $s_i$  are uniformly distributed over  $[1, 100]$ ;
- (4)  $a_{i,j}$  are uniformly distributed over  $(0.01, 0.5)$  and  $(0.5, 1)$  such that 
$$\beta \left( \sum_{i=1}^f (s_i + r_{i,\max}) + \sum_{i=1}^f \sum_{j=1}^{n_i} a_{i,j} - a_{\min} \right) < 1.$$

As a consequence, 25 experimental conditions were examined and 40 replications were randomly generated for each condition. A total of 1000 problems were tested. The percentage error of the solution produced by Algorithm 2 is calculated as

$$\frac{C_{\max}(A1) - C_{\max}^*}{C_{\max}^*} \times 100\%,$$

TABLE 1. Results of Algorithm 1 and the B&B algorithm for  $r_{ij} \sim [1, 100]$ .

$n$	$m$	$a_{ij} \sim (0.01, 0.5)$				$\alpha_{ij} \sim (0.5, 1)$				Error percentage (%)		Error percentage (%)	
		A1-CPU time (s) Mean	A1-CPU time (s) Max	B&B-CPU time (s) Mean	B&B-CPU time (s) Max	Error percentage (%) Mean	Error percentage (%) Max	A1-CPU time (s) Mean	A1-CPU time (s) Max	B&B-CPU time (s) Mean	B&B-CPU time (s) Max	Mean	Max
1000	250	54.68	55.36	96.65	181.31	0	0	54.25	54.64	9.14	89.1	0	0
	275	73.95	75.27	130.35	308.37	0	0	72.82	73.76	95.67	289.9	0	0
	300	96.45	97.86	98.27	365.88	0	0	95.45	96.22	144.7	416.36	0	0
	325	124.5	126.43	210.51	496.79	0	0	122.3	122.94	63.76	331.63	0	0
	350	156.86	158.83	176.8	582.11	0	0	154.17	155.24	129.4	391.43	0	0
1250	250	67.17	67.86	91.91	222.42	0	0	66.8	67.31	51.76	230.51	0	0
	275	90.28	90.89	71.6	268.48	0	0	89.72	89.95	78.71	315.78	0	0
	300	121.4	122.96	162.17	371.41	0	0	117.56	118.61	177.95	368.81	0	0
	325	153.27	157.91	186.1	445.02	0	0	151.22	152.64	162.42	363.39	0	0
	350	191.36	193.14	82.15	539.24	0	0	189.84	191.36	130.15	399.37	0	0
1500	250	80.1	80.64	63.58	250.35	8.72E-11	8.72E-10	79.14	80.2	54.11	142.71	0	0
	275	107	108.55	63.21	264.75	0	0	106.21	106.65	60.12	256.38	0	0
	300	139.83	141.61	43.81	350.52	0	0	139.47	140.01	96.42	329.91	0	0
	325	179.36	180.11	153.96	507.19	0	0	179.63	181.43	79.08	426.58	0	0
	350	225.95	228.32	162.52	601.29	0	0	225.19	227.38	193.22	464.73	0	0
1750	250	91.89	92.46	67.27	276.38	0	0	91.53	92.42	34.05	169.27	0	0
	275	122.45	123.44	130.71	382.06	0	0	122.71	123.35	30.46	102.61	0	0
	300	160.87	161.68	93.07	299.4	0	0	160.82	162.34	104.96	317.4	0	0
	325	206.97	209.79	212.4	463.51	0	0	206.65	207.83	149.22	281.6	0	0
	350	261.94	268.53	210.28	504.08	0	0	260.9	263.52	166.48	495.17	0	0
2000	250	103.92	104.57	26.44	146.42	0	0	104.67	106.71	65.94	176.34	0	0
	275	138.96	140.42	88.27	268.41	0	0	140.13	141.34	107.98	333.02	0	0
	300	181.62	182.65	156.2	340.37	0	0	184.57	185.41	165.5	414.24	0	0
	325	234.81	237.95	95.25	372.88	5.01E-12	5.01E-11	237.56	240.53	89.5	365.25	0	0
	350	294.21	296.61	119.15	387.12	0	0	298.93	304.21	223.75	544.31	0	0



where  $C_{\max}(A1)$  ( $C_{\max}^*$ ) is the makespan value generated by using Algorithm 1 (the optimal sequence can be obtained by Algorithm 2).

In addition, we define “A1-CPU time (s)” (“B&B CPU time (s)”) as the running time of Algorithm 1 (the B&B algorithm, *i.e.*, Algorithm 2). The corresponding results are summarized in Table 1. From Table 1, we can see that the performance of Algorithm 1 performs very well in terms of the error percentages. Moreover, Table 1 reveals that a relatively large number of instances (2000 jobs and 350 groups) were solved by B&B algorithm (a maximum CPU time less than 602 s).

#### 4. CONCLUSIONS

This study researches the single-machine group scheduling problem with ready times and shortened job processing times. The processing time of a job is a proportional linear shortening function of its start time, the setup time of each group is a constant, and the objective function is the maximum completion time of all jobs. The goal is to find the job sequence within each group and the group sequence such that the makespan is minimized. For the general case of the problem, we propose a branch-and-bound algorithm and a heuristic algorithm to solve this problem. From the experimental results, it can be seen that the effect of heuristic algorithm is significant. In future research, we can extend it to multi-machine problems such as parallel machines or flow shop settings. It is also interesting to extend the model to the single-machine group scheduling problem with general shortened job processing times, due-date assignment (see [29]) or positional-dependent learning effects (see [2, 4, 13]).

*Acknowledgements.* This Work Was Supported by the Natural Science Foundation of Liaoning Province, China (2020-MS-233). This Work Was Also Supported by the LiaoNing Revitalization Talents Program (grant no. XLYC2002017).

*Conflict of interest.* The authors declare that they have no conflict of interest.

#### REFERENCES

- [1] B. Alidaee and N.K. Womer, Scheduling with time dependent processing times: Review and extensions. *J. Oper. Res. Soc.* **50** (1999) 711–720.
- [2] A. Azzouz, M. Ennigrou and L.B. Said, Scheduling problems under learning effects: Classification and cartography. *Int. J. Prod. Res.* **56** (2018) 1642–1661.
- [3] J. Bai, Z.R. Li and X. Huang, Single-machine group scheduling with general deterioration and learning effects. *Appl. Math. Model.* **36** (2012) 1267–1274.
- [4] D. Biskup, A state-of-the-art review on scheduling with learning effects. *Eur. J. Oper. Res.* **188** (2008) 315–329.
- [5] T.C.E. Cheng, Q. Ding and B.M.T. Lin, A concise survey of scheduling with time-dependent processing times. *Eur. J. Oper. Res.* **152** (2004) 1–13.
- [6] J.M. Framinan and R. Leisten, An efficient constructive heuristic for flowtime minimization in permutation flow shops. *Omega* **31** (2003) 311–317.
- [7] S. Gawiejnowicz, *Models and Algorithms of Time-Dependent Scheduling*. Springer, Berlin (2020).
- [8] K.I.-J. Ho, J.Y.-T. Leung and W.-D. Wei, Complexity of scheduling tasks with time-dependent execution times. *Inf. Process. Lett.* **48** (1993) 315–320.
- [9] X. Huang, N. Yin, W.-W. Liu and J.-B. Wang, Common due window assignment scheduling with proportional linear deterioration effects. *Asia-Pac. J. Oper. Res.* **37** (2020) 1950031.
- [10] P. Ji and L. Li, Single-machine group scheduling problems with variable job processing times. *Math. Prob. Eng.* **2015** (2015) 758919 (9 pages).
- [11] T. Keshavarz, M. Savelsbergh and N. Salmasi, A branch-and-bound algorithm for the single machine sequence-dependent group scheduling problem with earliness and tardiness penalties. *Appl. Math. Model.* **39** (2015) 6410–6424.
- [12] L. Li and J.-J. Wang, Scheduling jobs with deterioration effect and controllable processing time. *Neural Comput. App.* **29** (2018) 1163–1170.
- [13] S.-S. Lin, A note on parallel-machine scheduling with controllable processing times and job-dependent learning effects. *RAIRO-Oper. Res.* **55** (2021) 561–569.
- [14] F. Liu, J. Yang and Y.-Y. Lu, Solution algorithms for single-machine group scheduling with ready times and deteriorating jobs. *Eng. Optim.* **51** (2019) 862–874.
- [15] Y.-Y. Lu, J.-J. Wang and J.-B. Wang, Single machine group scheduling with decreasing time-dependent processing times subject to release dates. *Appl. Math. Comput.* **234** (2014) 286–292.
- [16] Y.-Y. Lu, J. Jin, P. Ji and J.-B. Wang, Resource-dependent scheduling with deteriorating jobs and learning effects on unrelated parallel machine. *Neural Comput. App.* **27** (2016) 1993–2000.

- [17] Y.-Y. Lu, J.-B. Wang, P. Ji and H. He, A note on resource allocation scheduling with group technology and learning effects on a single machine. *Eng. Optim.* **49** (2017) 1621–1632.
- [18] J.S. Neufeld, J.N.D. Gupta and U. Buscher, A comprehensive review of flowshop group scheduling literature. *Comput. Oper. Res.* **70** (2016) 56–74.
- [19] C.N. Potts and L.N. Van Wassenhove, Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. *J. Oper. Res. Soc.* **43** (1992) 395–406.
- [20] J.-B. Wang and L.-Y. Sun, Single-machine group scheduling with decreasing linear deterioration setup times and job processing times. *Int. J. Adv. Manuf. Technol.* **49** (2010) 765–772.
- [21] J.-B. Wang and J.-J. Wang, Single machine group scheduling with time dependent processing times and ready times. *Inf. Sci.* **275** (2014) 226–231.
- [22] J.-B. Wang and Z.-Q. Xia, Scheduling jobs under decreasing linear deterioration. *Inf. Process. Lett.* **94** (2005) 63–69.
- [23] J.-B. Wang, A.-X. Guo, F. Shan, B. Jiang and L.-Y. Wang, Single machine group scheduling under decreasing linear deterioration. *J. Appl. Math. Comput.* **24** (2007) 283–293.
- [24] J.-B. Wang, L. Lin and F. Shan, Single-machine group scheduling problems with deteriorating jobs. *Int. J. Adv. Manuf. Technol.* **39** (2008) 808–812.
- [25] J.-B. Wang, X. Huang, Y.-B. Wu and P. Ji, Group scheduling with independent setup times, ready times, and deteriorating job processing times. *Int. J. Adv. Manuf. Technol.* **60** (2012) 643–649.
- [26] D. Wang, Y. Huo and P. Ji, Single-machine group scheduling with deteriorating jobs and allotted resource. *Optim. Lett.* **8** (2014) 591–605.
- [27] Z. Wang, C.-M. Wei and Y.-Y. Lu, Permutation flow shop problem with shortening job processing times. *Asia-Pac. J. Oper. Res.* **33** (2016) 1650032 (14 pages).
- [28] J.-B. Wang, L. Liu, J.-J. Wang and L. Li, Makenspan minimization scheduling with ready times, group technology and shortening job processing times. *Comput. J.* **61** (2018) 1422–1428.
- [29] L.-Y. Wang, X. Huang, W.-W. Liu, Y.-B. Wu and J.-B. Wang, Scheduling with position-dependent weights, due-date assignment and past-sequence-dependent setup times. *RAIRO-Oper. Res.* **55** (2021) S2747–S2758.
- [30] Y.-T. Xu, Y. Zhang and X. Huang, Single-machine ready times scheduling with group technology and proportional linear deterioration. *Appl. Math. Model.* **38** (2014) 384–391.

## Subscribe to Open (S2O)

A fair and sustainable open access model



This journal is currently published in open access under a Subscribe-to-Open model (S2O). S2O is a transformative model that aims to move subscription journals to open access. Open access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment. We are thankful to our subscribers and sponsors for making it possible to publish this journal in open access, free of charge for authors.

### Please help to maintain this journal in open access!

Check that your library subscribes to the journal, or make a personal donation to the S2O programme, by contacting [subscribers@edpsciences.org](mailto:subscribers@edpsciences.org)

More information, including a list of sponsors and a financial transparency report, available at: <https://www.edpsciences.org/en/maths-s2o-programme>