Single Machine Scheduling Problem with Controllable Setup and Job Processing Times and Position-Dependent Workloads

Chengxin Luo* and Geng Zhang

Department of basic courses, Guangdong University of science and technology, Dongguan, Guangdong, China

*Corresponding author

Abstract. This paper concerns with a single-machine scheduling problem in which each job has a setup before being processed, the setup time and the processing time of jobs are controllable by allocating a continuously nonrenewable resource, respectively. This is different from that of [13], in which a job only has a processing time, but without a setup time. We assume that each setup and processing time of jobs has a workload that is both job and position-dependent. In the problem investigated, the total amount of resource for setup and processing times does not exceed an up bound, respectively. The objective is to determine the setup sequence, the job sequence and the resource allocation scheme to minimize makespan. We show that the problem can be solved in polynomial time and present an optimal algorithm.

Keywords: Scheduling; Setup; Position-dependent workload; Makespan; Resource assumption.

1. Introduction

In most classical scheduling problems, it is assumed that the processing times of jobs remain unchanged throughout the production process. However, in various real life applications, jobs processing vary either over time or as a function of the job position in a sequence. The latter model is known as learning or aging effect, depending on whether job processing times become shorter or longer in the sequence. In the case of learning effect, it models that the operators or the machines are able to refine their skills by repetition of similar operations. As a result, the later a job is sequenced, the shorter its processing time becomes. On the other hand, in the aging setting, the machine gradually loses its efficiency as it lasts a long time, so the processing time of a job becomes longer if a job is sequenced later. This case is referred as position-aging in the literature. For more details, see [1,2,3].

It is natural that processing times of jobs may be controlled by allocating some resource, such as financial budget, energy, fuel, and so on. Using resource to reduce the job processing times is a common used approach in engineering system. In the literature convex resource assumption function are most frequently used. The convex decreasing resource assumption functions can reflect the law of diminishing marginal returns, by which productivity increases at a decreasing rate with the amount of resource used. Recently, increasing attention has been given to scheduling with controllable job processing times, see [4,5,6,7,8], for example. Research on scheduling with controllable job processing times was initiated by Vickson [9]. Survey on this area of scheduling research can be found in Shabtay and Steiner [10]. For new trends in scheduling with controllable job processing times and aging effect can be found in Rudek [11].
In most of the literature, the convex resource assumption function for processing time of jobs is of the form

\[ p_j(u_j) = \left( \frac{w_j}{u_j} \right)^k, u_j > 0, j = 1, 2, \ldots, n \] (1)

where \( w_j \) is a positive parameter, which represents the workload for operation of job \( J_j \), \( u_j \) is the amount of resource allocated to job \( J_j \), \( k > 0 \) is a given constant. Note that \( w_j \) is only job-dependent, but independent of job position in a sequence.

In some real manufacturing systems, a job or a batch of jobs needs a setup before being processed. The setup times are resource-dependent. Dvir Shabtay, George Steiner [12] study a single-machine batch scheduling to minimize total completion time and resource consumption costs, in which each batch has a setup, the setup time for each batch is given by

\[ s_i(v_i) = \left( \frac{w}{v_i} \right)^k, v_i > 0, i = 1, 2, \ldots, m \] (2)

where \( w \) is a positive parameter, which represents the workload for operation of setup \( i \), \( v_i \) is the amount of resource allocated to setup \( i \), \( m \) is the number of batches, \( k \) is a positive constant. Note that \( w \) is independent of job position in a sequence.

In the above papers, the setup and the processing times of jobs are independent of the number of tasks processed previously. However, there exist many manufacturing and service systems where worker and machines acquire, develop and refine skills through the repetition of identical or similar operations. As a result, it is necessary for researchers to study scheduling problems in which both the setup times and the processing times of jobs are position-dependent. In a recent paper, Daniel Orion [13] studies a single-machine scheduling problem with controllable processing time of jobs, in which processing time of jobs has a workload that is both job and position-dependent. The objective is to determine the job sequence and the resource allocation scheme to minimize makespan. He shows that the problem can be solved in polynomial time and an optimal algorithm is presented.

In this paper we assume a setup is added to a job before the job is processed. Further assume that the workload for both job and setup are position-dependent. This is the difference between our work and that of [13]. We use convex resource assumption function to express the relationships between the amounts of resource allocated and processing times of jobs and setup times. This results in different setting. A single-machine scheduling problem is investigated, here the setup and job processing times are convex function of the amount of resource they are derived. We further assume that the parameters of this function are position-dependent, i.e. vary with the setup and job position in the sequence. We show that the problem can be solved in polynomial time and present an optimal algorithm.

2. Problem Formulation

The single-machine scheduling problem under consideration can be stated as follows: \( n \) independent jobs \( J = \{J_1, J_2, \ldots, J_n\} \) are processed on a machine. All the jobs arrive at time \( t=0 \), the machine can handle at most one job at a time, job preemption is not allowed. Each job has a setup before being processed. The setup time of job \( J_j \), \( s_j \) if scheduled in the position \( r \) in a sequence, which is a decreasing convex function of the amount of a continuous and nonrenewable resource allocated to the setup, which is denoted by \( v_j \) is given by

\[ s_j(v_j) = \left( \frac{\theta_j}{v_j} \right)^k, v_j > 0, j = 1, 2, \ldots, n \] (3)

where \( \theta_j \) is a positive parameter, standing for the workload of the setup operation for setup \( j \) if scheduled in the \( r \) position in a sequence, \( l > 0 \) is a given constant.
The processing time of job $J_j$, $p_j$ is also a decreasing convex function of the amount of a continuous and nonrenewable resource allocated to the job, denoted by $u_j$. The resource consumption functions for processing time of job $J_j$ is as follows

$$p_{jr}(u_j) = \left( \frac{w_{jr}}{u_j} \right)^k, u_j > 0, j = 1, 2, \ldots, n$$

(4)

where $w_{jr}$ is a positive parameter, representing the workload of the processing operation for job $J_j$ if scheduled in the $r$ position in a sequence, $k > 0$ is a given constant.

The problem we study is to determine the setup sequence and job sequence, and the resource allocation scheme $\{v_j, u_j\}$ for setups and jobs to minimize $C_{\text{max}}$, the makespan, subject to the total amount of resource is bounded by a given value $V, U$, respectively, i.e. the resource used for setups and jobs are subject to $\sum_{j=1}^{n} v_j \leq V, \sum_{j=1}^{n} u_j \leq U$. Using the three field notation [14], this problem can be denoted as

$$\text{P1: } 1|\text{setup, conv}, \sum_{j=1}^{n} v_j \leq V, \sum_{j=1}^{n} u_j \leq U | C_{\text{max}}$$

where $\text{conv}$ in the second field represents the convex resource assumption functions are used, the makespan $C_{\text{max}}$ is defined as $C_{\text{max}} = \max_{1 \leq j \leq n} \{C_j\}$, $C_j$ is the completion time of job in a sequence.

3. An Optimal Algorithm for Minimizing the Makespan

This section investigates the problem of minimizing the makespan subject to limited resource availability. Firstly, we will derive the objective value assume that the job sequence and the setup sequence are given in advance. We first define a set of binary decision variables. Then we will present a Mixed Integer Non-Linear Programming (MINLP) model for our problem. Let $0-1$ variables $\{X_{jr}\}$ be $X_{jr} = 1$, if $J_j$ is scheduled in the $r$ position, $X_{jr} = 0$, otherwise, $j, r = 1, 2, \ldots, n$.

Note that $C_{\text{max}} = \max_{1 \leq j \leq n} \{C_j\}$, so problem P1 can be changed to the following problem

$$\text{P2: } \begin{cases} \min f = C_{\text{max}}(u, v) = \sum_{j=1}^{n} \sum_{r=1}^{n} \left( \theta_{jr} / v_j \right)^{j} X_{jr} + \sum_{j=1}^{n} \sum_{r=1}^{n} \left( w_{jr} / u_j \right)^{k} X_{jr} \\ \sum_{j=1}^{n} v_j \leq V, v_j > 0; \sum_{j=1}^{n} u_j \leq U, u_j > 0, j = 1, 2, \ldots, n \\ s.t.: \sum_{j=1}^{n} X_{jr} = 1, r = 1, 2, \ldots, n; \sum_{j=1}^{n} X_{jr} = 1, j = 1, 2, \ldots, n \\ X_{jr} \in \{1, 0\}, j, r = 1, 2, \ldots, n \end{cases}$$

(5)

Lemma 1 For a given $\{X_{jr} = 1, j, r = 1, 2, \ldots, n\}$, the optimal resource allocation policies for the setups and jobs, $v = (v_1, v_2, \ldots, v_n)$, $u = (u_1, u_2, \ldots, u_n)$ are given by

$$v_j^* = \frac{\sum_{r=1}^{n} \theta_{jr} X_{jr}^{j(i+1)}}{\sum_{j=1}^{n} \sum_{r=1}^{n} \theta_{jr} X_{jr}^{j(i+1)}}, \quad u_j^* = \frac{\sum_{r=1}^{n} w_{jr} X_{jr}^{k(i+1)}}{\sum_{j=1}^{n} \sum_{r=1}^{n} w_{jr} X_{jr}^{k(i+1)}}, \quad j = 1, 2, \ldots, n$$

(6)

Proof Note that $f$ is inversely proportional to the amount of resource allocated, so in any optimal solution in (5) are tight. In fact, if there is any amount resource left for setup, allocating it to any setup will reduce the objective function value, hence any optimal policy for setup will always consist of allocating all available resource. Similarly, any optimal policy for processing time will always
consist of allocating all available resource. Hence the resource constraints can be rewritten as equations. As a result, we can use Lagrangian multiplier method to solve the problem \textbf{P2}. The Lagrangian function is as follow:

\[
L(v, u, \lambda, \eta) = \sum_{j=1}^{n} \sum_{r=1}^{n} \left( \theta_{jr} v_j X_{jr} \right)^i \left( \sum_{j=1}^{n} \sum_{r=1}^{n} w_{jr} X_{jr} \right)^k \lambda \left( \sum_{j=1}^{n} v_j - V \right) + \eta \left( \sum_{j=1}^{n} u_j - U \right)
\]

where \( \lambda, \eta \) are the Lagrangian multipliers. Since \( L(v, u, \lambda, \eta) \) is a convex function of all the decision variables, the following necessary conditions for optimality are also sufficient:

\[
\frac{\partial L(v, u, \lambda, \eta)}{\partial \lambda} = \sum_{j=1}^{n} v_j - V = 0, \quad \frac{\partial L(v, u, \lambda, \eta)}{\partial \eta} = \sum_{j=1}^{n} u_j - U = 0
\]

(7)

\[
\frac{\partial L(v, u, \lambda, \eta)}{\partial v_j} = \frac{\partial f}{\partial v_j} = \frac{\partial \bar{f}}{\partial v_j} + \lambda, \quad \frac{\partial L(v, u, \lambda, \eta)}{\partial u_j} = \frac{\partial f}{\partial u_j} + \eta = 0
\]

(8)

where \( j = 1, 2, \ldots, n \). From (7)-(8), we derive

\[
\frac{\partial \bar{f}}{\partial v_j} = \frac{\partial \bar{f}}{\partial v_i}, \quad \frac{\partial \bar{f}}{\partial u_j} = \frac{\partial \bar{f}}{\partial u_i}, \quad j = 1, 2, \ldots, n
\]

(9)

So from (7)-(9) we obtain

\[
U_j = u_j \left( \sum_{j=1}^{n} w_{jr} X_{jr} \right)^{i(k+1)} \left( \sum_{j=1}^{n} X_{jr} \right)^{i} \lambda, \quad V_j = v_j \left( \sum_{j=1}^{n} \theta_{jr} X_{jr} \right)^{i(k+1)} \left( \sum_{j=1}^{n} \theta_{jr} X_{jr} \right)^{i} \lambda
\]

(10)

where \( j = 2, \ldots, n \). Substituting (10) in (7), we have

\[
U_j = U \left( \sum_{j=1}^{n} w_{jr} X_{jr} \right)^{i(k+1)} \left( \sum_{j=1}^{n} X_{jr} \right)^{i} \lambda, \quad V_j = V \left( \sum_{j=1}^{n} \theta_{jr} X_{jr} \right)^{i(k+1)} \left( \sum_{j=1}^{n} \theta_{jr} X_{jr} \right)^{i} \lambda
\]

(11)

Finally insert (11) in (10), respectively, we obtain (6). This completes the proof.

**Lemma 2** Under the optimal resource allocation polices (6), as a function of the job sequence and the setup sequence, the function value of \( f \) is given by

\[
f = V^{-1} \left( \sum_{j=1}^{n} \left( \sum_{r=1}^{n} \theta_{jr} X_{jr} \right)^{i(k+1)} \right)^{i} + U^{-1} \left( \sum_{j=1}^{n} \left( \sum_{r=1}^{n} w_{jr} X_{jr} \right)^{i(k+1)} \right)^{i}
\]

(12)

**Proof** Substituting (6) into the first equation in (3), we have (12). This completes the proof. The next theorem shows that the makespan minimization problem can be reduced to linear assignment problems. As a result the problem cab be solved in polynomial time.

**Theorem 1** The makespan minimization problem can be solved in \( O(n^3) \) time.

**Proof** We see that in (12) \( k, l, V, U \) are all positive constants, and \( f \) is separated in decision variables \( X_{jr} \), so we only need to minimize the following functions under some certain conditions:

\[
f_1 = \sum_{j=1}^{n} \left( \sum_{r=1}^{n} \theta_{jr} X_{jr} \right)^{i(k+1)}, \quad f_2 = \sum_{j=1}^{n} \left( \sum_{r=1}^{n} w_{jr} X_{jr} \right)^{i(k+1)}
\]

(13)

Note that \( f_1, f_2 \) are independent of the optimal resource allocation to setups and jobs, respectively. On the other hand, we have

\[
f_1 = \sum_{j=1}^{n} \left( \sum_{r=1}^{n} \theta_{jr} X_{jr} \right)^{i(k+1)} = \sum_{j=1}^{n} \sum_{r=1}^{n} \theta_{jr} X_{jr}
\]

This is because \( \sum_{j=1}^{n} X_{jr} = 1, j = 1, 2, \ldots, n \), and the fact that \( \{ X_{jr}, j, r = 1, 2, \ldots, n \} \) are binary. With a similar argument we have

\[
f_2 = \sum_{j=1}^{n} \sum_{r=1}^{n} w_{jr}^{i(k+1)} X_{jr}
\]

Hence to minimize that \( f_1, f_2 \) it suffices to solve the following linear assignment problems, respectively
To solve LAP 1, LAP 2, we need $O(n^3)$ time. After the optimal setup sequence and the optimal job sequence are obtained, we can calculate the optimal resource policies to the setups and jobs by (6) in $O(n^2)$ time. So the overall effort is $O(n^3)$ time. This completes the proof.

To summarize, we present the following optimal algorithm to solve our problem P1.

**Algorithm 1** Input $k, I, V, U, \theta_j, w_{jr}, j, r = 1, 2, \ldots, n$.

Step 1. Solve LAP 1, LAP 2 to obtain the optimal setup sequence, the optimal job sequence, respectively.

Step 2. Allocate the optimal setup resource $v_j$, the optimal job resource $u_j$ ($j = 1, 2, \ldots, n$) according to (6), respectively.

Step 3. Compute the optimal makespan value by (12). (In fact, the optimal makespan $C_{\text{max}}^*$ can be represented as $C_{\text{max}}^* = A/V^l + B/U^k$, where $A, B$ are the optimal objective values of LAP 1, LAP 2, respectively.)

**Acknowledgement**

This research was financially supported by the National Science Foundation (under Grant:11171050).

**References**

[1] Browne, S., Yechiali, U., Scheduling deteriorating jobs on a single processor, Operations Research. 38 (1990) 495-498.

[2] Gawiejnowics, Time-Dependent Scheduling problem, Springer-Verlag, Berlin, Heidelberg. 2008.

[3] D. Biskup, A state-of-the-art review on scheduling with learning effects, European Journal of Operational Research. 188 (2008) 315-329.

[4] Yang, D.-L., T.C.E. Cheng, and S.-J. Yang, Parallel-machines scheduling with controllable processing times and rate-modifying activities to minimise total cost involving total completion time and job compressions. International Journal of Production Research. 52 (2014) 1133-1141.

[5] Cheng, T. C. E., Janiak, A. and Kovalyov, M. Y. Single machine batch scheduling with resource dependent setup processing times, European Journal of Operational Research. 135 (2001) 177-183.

[6] Gang Li, Mei-Ling Luo, Wen-Jie Zhang, Xiao-Yuan Wang, Single machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation. International Journal of Production Research. 53 (4) (2015) 1228-1241.

[7] Ji-Bo Wang, Jian-Jun Wang, Research on scheduling with job-dependent learning effect and convex resource-dependent processing times, International Journal of Production Research. 53 (19) (2015) 5826-5836.
[8] Dvir Shabtay, Single and a two-resource allocation algorithms for minimizing the maximal lateness in a single machine-scheduling problem, Computers and Operations Research. 31 (8) (2004) 1303-1315.

[9] Vickson, R.G., Two single-machine sequencing problems involving controllable job processing times, AIIE Transactions. 3 (1980) 258-262.

[10] Shabtay, D. and Steiner G., A survey of scheduling with controllable processing times, Discrete Applied Mathematics. 13(2007) 1643-1666.

[11] Rudek A. and Rudek R., A note on optimization in deteriorating systems using scheduling problems with the aging effect and resource allocation models, Computer and Mathematics with Applications. 4 (2011) 1870-1878.

[12] Dvir Shabtay, George Steiner, Single machine batch scheduling to minimize total completion time and resource consumption costs, Journal of Scheduling. 10 (2007) 255-261.

[13] D. Oron, Scheduling controllable processing time jobs with position-dependent workloads, International Journal of Production Economics. 173 (2016) 153-160.

[14] Graham, R. L., et al., Optimization and approximation in deterministic sequencing and scheduling: a survey, Annals of Discrete Mathematics. 3 (1979) 287-326.