Release와 Processing time이 투입자원에 종속적인 단일설비 일정계획문제의 Strong NP-completeness 분석*

이 익 선†
동아대학교 경영대학 경영학과

Strong NP-completeness of Single Machine Scheduling with Resource Dependent Release Times and Processing Times

Ik Sun Lee†
School of Business, Dong-A University

Abstract

This paper considers a single machine scheduling problem to determine release and processing times where both the release times and processing times are linearly decreasing functions of resources. The objective is to minimize the sum of the associated resource consumption cost and scheduling cost including makespan, sum of completion times, maximum lateness, or sum of lateness. This paper proves that the scheduling problem is NP-hard in the strong sense even if the release times are constant.

Keywords: Scheduling, Resource Dependent Release Times, Resource Dependent Processing Times
1. Introduction and Problem Description

This paper considers the single machine scheduling problem with resource dependent release times and processing times, where “resource dependent release times and processing times” means that both the release times and processing times are linearly decreasing functions of resources. More specifically, both the job release times and processing times depend on the amount of resource consumption. The proposed scheduling problem commonly arises in the chemical processing industry, referring to Wang and Cheng [4]. The objective of the scheduling problem is to minimize the sum of resource consumption cost and scheduling cost including makespan, sum of completion times, maximum lateness, or sum of lateness.

In the proposed problem, there are \( n \) jobs on a single machine which are initially available at time \( v \), but each job may be made available at an earlier time point by consuming extra resources that will incur additional costs. Each job \( J_j \) has a due date \( d_j \), a processing time \( p_j \) and a release time \( r_j \), \( j = 1, \ldots, n \), where both \( p_j \) and \( r_j \) depend on the amount of resource consumption. Specifically, \( p_j = a_j - \delta_j \), where \( a_j \) is the normal processing time and \( \delta_j \) is the amount of processing time compression, \( 0 \leq \delta_j \leq a_j \); \( r_j = v - u_j/w \), where \( u_j \) is the cost of resource consumed to advance the availability of \( J_j \) to \( r_j \) and \( w \) the cost per unit reduction of release time, \( 0 \leq u_j \leq wv \). Denote by \( C_j \) the completion time of job \( J_j \). Jobs are assumed to be processed in the earliest release date (ERD) order after they are released.

Referring to the classification scheme of Graham et al. [2] \( \alpha \mid \beta \mid \gamma \), the proposed problem is classified to have \( \alpha = 1 \) (only single machine problem), and “pr\(_d\)” and “rel\(_d\)” constraints for \( \beta \), where “pr\(_d\)” and “rel\(_d\)” indicate resource dependent processing times and release times, respectively. Moreover, for \( \gamma \), the objective function of the proposed problem is composed of resource consumption cost and scheduling cost. The resource consumption cost \( "RC" \) is composed of the total processing time reduction cost \( \Sigma_{j=1}^n c_j \delta_j \) and the total release time reduction cost \( \Sigma_{j=1}^n w(v-r_j) \), where \( c_j, j = 1, \ldots, n \), is the cost per unit processing time reduction, then \( RC = \Sigma_{j=1}^n c_j \delta_j + \Sigma_{j=1}^n w(v-r_j) \). The scheduling cost may be represented by one of the followings:

\[
C_{\text{max}} = \max C_j = \max_{1 \leq j \leq n} \left(r_j + \Sigma_{i=j}^{n}(a_i - \delta_i)\right) \quad \text{(makespan)}
\]

\[
\sum C_j = \sum_{i=1}^{n} \max_{1 \leq j \leq i} \left(r_j + \Sigma_{i=j}^{n}(a_i - \delta_i)\right) \quad \text{(sum of completion times)}
\]

\[
L_{\text{max}} = \max \{C_j - d_j, 0\} = \max_{1 \leq j \leq n} \left(\max_{1 \leq i \leq j} \left\{r_i + \Sigma_{i=j}^{n}(a_i - \delta_i) - d_j, 0\right\}\right) \quad \text{(maximum lateness)}
\]

\[
\Sigma_{j=1}^{n} L_j = \Sigma_{j=1}^{n} \max \{C_j - d_j, 0\} = \Sigma_{j=1}^{n} \max_{1 \leq i \leq j} \left\{r_i + \Sigma_{i=j}^{n}(a_i - \delta_i) - d_j, 0\right\} \quad \text{(sum of lateness)}
\]

Under the constraint of a common deadline, Janiak [3] has shown that the single machine scheduling problem of minimizing resource consumption (where the job release times follow a linear model, \( r_j = v - u_j/w \), \( j = 1, \ldots, n \) but the processing times are constant) is NP-hard in the ordinary sense. Referring to the Janiak’s result, Wang and Cheng [4] have just commented that the problem 1|pr\(_d\), rel\(_d\)|C\(_{\text{max}}\)+RC is NP-hard. However, they haven’t characterized whether the problem 1|pr\(_d\), rel\(_d\)|C\(_{\text{max}}\)+RC is NP-hard in the strong sense or not, which is the motivation...
of this paper. It seems to be important whether a problem is NP-hard in the strong sense or not, because “strongly NP-hard” means that the associated problem is more difficult to solve. Therefore, it is more valuable to derive any heuristic algorithms for that problem. Thus, this paper proves that the problems 1\(|pr_d, rel_d|C_{\text{max}} + RC\), 1\(|pr_d, rel_d|\sum C_j + RC\), 1\(|pr_d, rel_d|L_{\text{max}} + RC\) and 1\(|pr_d, rel_d|\sum_j L_j + RC\) are NP-hard in the strong sense even if the release times are constant. “the release times are constant” means that the release time of all the products are the same.

2. Strong NP-completeness of the Proposed Scheduling Problem

Theorem 1: The problem 1\(|pr_d, rel_d|C_{\text{max}} + RC\) is NP-hard in the strong sense even if the release times are constant.

Proof: The proof is made by reduction from the 3-Partition Problem [1] which is known to be NP-hard in the strong sense. The problem is stated as follows;

Given \(3q\) elements with integer size \(e_1, \ldots, e_{3q}\), where \(\sum_{i=1}^{3q} e_i = qB\) and \(B/4 < e_i < B/2\) for \(i = 1, \ldots, 3q\) does there exist a partition \(S_1, \ldots, S_q\) of the index set \{1, ..., 3q\} such that \(|S_j| = 3\) and \(\sum_{i \in S_j} e_i = B = B/2\) for \(j = 1, \ldots, q\).

Now, consider the following instance of the problem 1\(|pr_d, rel_d|C_{\text{max}} + RC\);

\[
\begin{align*}
\text{number of jobs } n & = 3q + 1, \\
r_{3q+1} & = r_{3q} = (j - 1)B/2, \quad j = 1, \ldots, q, \\
r_{3q+1} & = qB/2, \\
c_{3q+1} & = c_{3q} = \frac{4j}{q(q+1)B}, \quad j = 1, \ldots, q, \\
C_{\text{max}} & = 0, \\
a_j & = B/4, \quad j = 1, \ldots, 3q, \\
e_j & = e_j = B/4, \quad j = 1, \ldots, 3q, \\
\delta_j & = \delta_{3q} = 0, \\
\rho_j & = \rho_{3q+1} = 0.
\end{align*}
\]

Since \(B/4 < e_i < B/2\), then the relation \(0 < \delta_j < a_j\) holds, which is a subset of \(0 \leq \delta_j \leq a_j\); that is, \(\{0 < \delta_j < a_j\}\) is a special case of \(\{0 \leq \delta_j \leq a_j\}\). Therefore, it is reasonable to consider the relation \(0 < \delta_j < a_j\) in the proof. Moreover, define a threshold value, \(Q\) as

\[Q = qB/2 + 1/2\]

Now, it is proved that there exists a feasible schedule for the instance of the problem 1\(|pr_d, rel_d|C_{\text{max}} + RC\) with \(C_{\text{max}} + RC \leq Q\) if and only if there exists a solution to the 3-Partition Problem.

a) For the if-part: Suppose that there are \(q\) disjoint sets \(S_1, \ldots, S_q\) which comprise a solution to the 3-Partition Problem, such as \((\tau_1, \eta_1, \mu_1), \ldots, (\tau_q, \eta_q, \mu_q)\), where \((\tau_1, \eta_1, \mu_1, \ldots, \tau_q, \eta_q, \mu_q) = (e_1, \ldots, e_{3q})\) and \(\tau_k + \eta_k + \mu_k = B\) for \(k = 1, \ldots, q\). Then, the associated job sets \(S_1' = \{J_{1,1}, J_{1,2}, J_{1,3}\}, \ldots, S_q' = \{J_{q,1}, J_{q,2}, J_{q,3}\}\) have the processing times

\[
\left(\frac{B}{2} - \tau_1, \frac{B}{2} - \eta_1, \frac{B}{2} - \mu_1\right), \ldots, \left(\frac{B}{2} - \tau_q, \frac{B}{2} - \eta_q, \frac{B}{2} - \mu_q\right),
\]

respectively, where \(J_{1,1}, J_{1,2}, J_{1,3} = \ldots, J_{q,1}, J_{q,2}, J_{q,3}\). The total processing time of the three jobs in \(S_j'\) is \(\frac{B}{2}, k = 1, \ldots, q\). Therefore, the total resource consumption cost is \(B/4 \sum_{j=1}^{q} \frac{4j}{q(q+1)B} = \sum_{j=1}^{q} \frac{j}{q(q+1)} = 1/2\). Job \(J_{3q+1}\) is sequenced at the last position, so that the makespan associated with the schedule is \(qB/2\). Thus, the total objective cost is \(qB/2 + 1/2\).

b) For the only if-part: Suppose that there is a feasible schedule \(\pi\) satisfying the relation \(C_{\text{max}}\)
Since the release time $r_{3q+1}$ is $qB/2$ and the threshold value is $Q = qB/2 + 1/2$, job $J_{3q+1}$ is sequenced at the last position. This paper assumes that job $J_{3q+1}$ is immediately processed at time $r_{3q+1} = qB/2$ without waiting, since $c_j < 1$, which is more cost-effective. Then, the makespan associated with the schedule $\pi$ is $qB/2$, which implies that the total resource consumption cost is not greater than $1/2$.

Referring to Lemma 1 in [4], the schedule $\pi$ needs to have no machine idle time. Therefore, by subtracting the makespan of $qB/2$ from the term $\sum_{j=1}^{3q+1} a_j = 3qB/4$, the total processing time reduction amount associated with the schedule $\pi$ is $qB/4$.

Now, $\delta_j$'s (the amount of processing time compression of each job $j$) are found by use of the following mathematical programming:

\[
\begin{align*}
\text{minimize} & \quad \left( (\delta_1 + \delta_2 + \delta_3) + 2(\delta_1 + \delta_4 + \delta_5) + \cdots + q(\delta_{3q-2} + \delta_{3q-1} + \delta_{3q}) \right) \frac{4}{q(q+1)B} \\
\text{subject to} & \quad \begin{align*}
(\delta_1 + \delta_2 + \delta_3) + (\delta_4 + \delta_5 + \delta_6) + \cdots + (1) \\
(\delta_{3q-2} + \delta_{3q-1} + \delta_{3q}) = qB/4 \\
(\delta_1 + \delta_2 + \delta_3) \leq B/4 \quad (2) \\
(\delta_1 + \delta_2 + \delta_3) + (\delta_4 + \delta_5 + \delta_6) \leq 2(B/4) \quad (3) \\
& \quad \vdots \\
(\delta_1 + \delta_2 + \delta_3) + (\delta_4 + \delta_5 + \delta_6) + \cdots + (q) \\
(\delta_{3(q-1)-2} + \delta_{3(q-1)-1} + \delta_{3q-1}) & \leq (q-1)(qB/4)
\end{align*}
\]

Constraint (1) represents that the total processing time reduction amount is $qB/4$. Constraints (2)~(q) represent that the schedule $\pi$ has no machine idle time, since $a_{3q-2} + a_{3q-1} + a_{3q} = 3B/4$ and $r_{3q-2} = r_{3q-1} = r_{3q} = (j-1)B/2$, $j = 1, \ldots, q$. The above mathematical programming achieves a minimum at $(\delta_1 + \delta_2 + \delta_3) = (\delta_4 + \delta_5 + \delta_6) = \cdots = (\delta_{3q-2} + \delta_{3q-1} + \delta_{3q}) = B/4$, and the associated minimum objective value is equal to $1/2$. The relation $\delta_{3q-2} + \delta_{3q-1} + \delta_{3q} = B/4$ is equivalent to the relation $(c_{3q-2} + c_{3q-1} + c_{3q}) = B$, $j = 1, \ldots, q$. This implies the existence of a solution to the 3-Partition Problem.

This completes the proof.

Theorem 2: The problem $1|\text{pr}_d, \text{rel}_d| \sum C_j + RC$ is NP-hard in the strong sense even if the release times are constant.

Proof: The proof is made by reduction from the Numerical 3-Dimensional Matching Problem [1] which is NP-hard in the strong sense. The problem is stated as follows:

Given three sets $X$, $Y$, $Z$ of $q$ positive integers $X = \{x_1, x_2, \ldots, x_q\}$, $Y = \{y_1, y_2, \ldots, y_q\}$ and $Z = \{z_1, z_2, \ldots, z_q\}$ with $\sum_{i=1}^{q} (x_i + y_i + z_i) = qB$, decide if there exist one-to-one functions $\phi$ and $\psi$ on the sets $S_1, S_2, \ldots, S_q$ such that $x_i + y_{\phi(i)} + z_{\psi(i)} = B$, for all $S_i = \{x_i, y_{\phi(i)}, z_{\psi(i)}\}$, $i = 1, 2, \ldots, q$.

Now, consider the following instance of the problem $1|\text{pr}_d, \text{rel}_d| \sum C_j + RC$:

\[
\begin{align*}
n &= 3q+1, w = 0, \\
r_j &= r_{q+j} = r_{2q+j} = 5(j-1)B, \quad j = 1, \ldots, q, \quad r_{3q+1} = 5qB, \\
c_j &= c_{q+j} = c_{2q+j} = \frac{j}{q(q+1)B}, \quad j = 1, \ldots, q, \quad c_{3q+1} = 0, \\
a_j &= B, \quad j = 1, \ldots, q, \\
a_j &= 2B, \quad j = q+1, \ldots, 2q, \\
a_j &= 3B, \quad j = 2q+1, \ldots, 3q, \\
a_{3q+1} &= 0, \\
\delta_j &= x_j, \quad \delta_{j+q} = y_j, \quad \delta_{j+2q} = z_j, \quad j = 1, \ldots, q, \quad \delta_{3q+1} = 0.
\end{align*}
\]

Then, $p_j = B - x_j, p_{j+q} = 2B - y_j, p_{j+2q} = 3B - z_j, \quad j = 1, \ldots, q$. Since $0 < x_j, y_j, z_j < B$, the relation $0 < \delta_j < a_j$ holds, which is a subset of $\{0 \leq \delta_j \leq a_j\}$; that is, $\{0 < \delta_j < a_j\}$ is a special case of
\(0 \leq \delta_j \leq a_j\). Therefore, it is reasonable to consider the relation \(0 < \delta_j < a_j\) in the proof. Moreover, define a threshold value, \(Q\), as

\[
Q = 15Bq^2/2 + 15Bq/2 + 1/2 - \sum_{i=1}^{l-1} (3x_i + 2y_i + z_i)
\]

Now, it is proved that there exists a feasible schedule for the instance of the problem 1\(|\text{pr}_d, \text{rel}_d|\ \Sigma C_j + RC\) with \(\Sigma C_j + RC \leq Q\) if and only if there exists a solution to the Numerical 3-Dimensional Matching Problem.

a) For the if-part: Suppose that there are \(q\) disjoint sets \(S_1, \ldots, S_q\) which comprise a solution to the Numerical 3-Dimensional Matching Problem, i.e., \(\{x_1, y_{0(1)}, z_{\varphi(1)}\}, \{x_2, y_{0(2)}, z_{\varphi(2)}\}, \ldots, \{x_q, y_{0(q)}, z_{\varphi(q)}\}\) such as \(x_1 + y_{0(j)} + z_{\varphi(j)} = B, j = 1, \ldots, q\). The solution values \(x_j, y_{0(j)}\) and \(z_{\varphi(j)}\) are assigned to \(\delta_j, \delta_j + \gamma\), respectively. Jobs with identical release time are processed in SPT (Shortest Processing Time) order, such as \(\pi_j = \pi_{j+1}\). The schedule \(\pi\) of the proposed problem is represented by the job sequence \(~\{J_1, J_{q+1}, J_{2q+1}, \ldots, J_q, J_{2q}, J_{3q}\}\) without machine idle time. Then, it follows that the schedule \(\pi\) has the sum of completion times \(\Sigma_{j=1}^{l-1} C_j = 15Bq^2/2 + 15Bq/2 - \sum_{i=1}^{l-1} (3x_i + 2y_i + z_i)\), and the total resource consumption cost \(\sum_{j=1}^{l-1} (c_j\delta_j + c_j + \gamma\delta_j + c_{j+2}\delta_j + c_{j+2y}z_j) = B\sum_{j=1}^{l-1} j/q(q+1)B = \sum_{j=1}^{l-1} j/q(q+1) = 1/2\), so that \(\Sigma C_j + RC = Q\).

b) For the only if-part: Suppose that there is a feasible schedule \(\pi\) satisfying the relation \(\Sigma C_j + RC \leq Q\). Due to the relation \(\max_{1 \leq j \leq 2q} C_j < 1\), the processing of each earlier arrived job is completed before arrival of any next jobs, since the relation implies more cost-effective. Thus, any one of simultaneously arrived jobs can be immediately processed without waiting. Therefore, three jobs arrive simultaneously except for \(J_{3q+1}\), so that the three jobs need to be processed in SPT order so as to be more cost-effective. This leads to the processing order of \((J_j, J_{q+j}, J_{2q+j}), j = 1, \ldots, q\), among the three simultaneously arrived jobs and their processing times of \((B - \delta_j, 2B - \delta_{j+\gamma}, 3B - \delta_{j+2\gamma})\). However, the last job \(J_{3q+1}\) having the release time of \(5qB\) arrives last by itself, so that it should be immediately processed without waiting at the last position in the sequence. Moreover, in the schedule \(\pi\), the processing of each earlier arrived job is completed before arrival of any next jobs, so that

\[
\delta_j + \delta_{j+\gamma} + \delta_{j+2\gamma} \geq B, j = 1, \ldots, q. \quad (q+1)
\]

since \(r_j = r_j + 1 - r_{j+\gamma} = r_{j+2\gamma} + 1 - r_{j+2\gamma} = 5B, j = 1, \ldots, q-1\).

Referring to Lemma 1 in [4], the schedule \(\pi\) needs to have no machine idle time, so that

\[
\delta_j + \delta_{j+\gamma} + \delta_{j+2\gamma} \leq B, j = 1, \ldots, q. \quad (q+2)
\]

From Eqs. (q+1) and (q+2), it follows that

\[
\delta_j + \delta_{j+\gamma} + \delta_{j+2\gamma} = B, j = 1, \ldots, q. \quad (q+3)
\]

Based on Eq. (q+3), the sum of completion times associated with the schedule \(\pi\) is \(15Bq^2/2 + 15Bq/2 - \sum_{i=1}^{l-1} (3x_i + 2y_i + z_i)\), and the total resource consumption cost is \(\sum_{j=1}^{l-1} (c_j\delta_j + c_j + \gamma\delta_j + c_{j+2}\delta_j + c_{j+2y}z_j) = B\sum_{j=1}^{l-1} j/q(q+1)B = \sum_{j=1}^{l-1} j/q(q+1) = 1/2\), so that \(\Sigma C_j + RC = Q\), which satisfies the hypothesis. Moreover, the relation \((\delta_j + \delta_{j+\gamma} + \delta_{j+2\gamma} = B)\) is equivalent to the relation \((x_j + y_j + z_j) = B, j = 1, \ldots, q\). This
implies the existence of a solution to the Numerical 3-Dimensional Matching Problem. This completes the proof.

**Theorem 3** : The problems $1|pr_d, rel_d|L_{max} + RC$ and $1|pr_d, rel_d|\Sigma_{j=1}^{n} L_j + RC$ are NP-hard in the strong sense even if the release times are constant.

**Proof** : If the value 0 is assigned to $d_j$‘s, $j = 1, ..., n$, then the problems $1|pr_d, rel_d|L_{max} + RC$ and $1|pr_d, rel_d|\Sigma_{j=1}^{n} L_j + RC$ are equivalent to the problems $1|pr_d, rel_d|C_{max} + RC$ and $1|pr_d, rel_d|\Sigma C_j + RC$. Therefore, the problems $1|pr_d, rel_d|L_{max} + RC$ and $1|pr_d, rel_d|\Sigma_{j=1}^{n} L_j + RC$ are NP-hard in the strong sense even if the release times are constant, since their special cases are NP-hard in the strong sense.

This completes the proof.

In summary, this paper proves that the problems $1|pr_d, rel_d|C_{max} + RC$, $1|pr_d, rel_d|\Sigma C_j + RC$, $1|pr_d, rel_d|L_{max} + RC$ and $1|pr_d, rel_d|\Sigma_{j=1}^{n} L_j + RC$ are NP-hard in the strong sense even if the release times are constant.

**References**


