




# Single-machine scheduling with periodic due dates to minimize the total earliness and tardy penalty

Byung-Cheon Choi<sup>1</sup> · Myoung-Ju Park<sup>2</sup> 

Accepted: 15 February 2021 / Published online: 9 March 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

We consider a single-machine scheduling problem such that the due dates are assigned to each job depending on its order, and the lengths of the intervals between consecutive due dates are identical. The objective is to minimize the total penalty for the earliness and tardiness of each job. The early penalty proportionally increases according to the earliness amount, while the tardy penalty increases according to the step function. We show that the problem is strongly NP-hard, and furthermore, polynomially solvable if the two types of processing times exist.

**Keywords** Single-machine scheduling · Generalized due dates · Computational complexity

## 1 Introduction

Consider a scheduling problem such that unlike the traditional scheduling problem, the due date is assigned not to the specific job but to the job position. This due date is referred to as the *generalized due date (GDD)*.

The scheduling problem with GDD was introduced by Hall (1986). Hall (1986) and Hall et al. (1991) established the computational complexities for single-machine cases with various objective functions (e.g., maximum lateness, total weighted completion time, total weighted tardiness, and weighted number of tardy jobs). In particular, Hall (1986) showed that the problem to minimize the total tardiness is solvable in

---

✉ Myoung-Ju Park  
pmj0684@khu.ac.kr  
Byung-Cheon Choi  
polytime@cnu.ac.kr

<sup>1</sup> School of Business, Chungnam National University, 99 Daehak-ro, Yuseong-gu, Daejeon 34134, Korea

<sup>2</sup> Department of Industrial and Management Systems Engineering, Kyung Hee University, 1732, Deogyong-daero, Giheung-gu, Yongin-si, Kyunggi-do 17104, Korea

polynomial time. Srikandarajah (1990) and Gao and Yuan (2016) considered a single-machine case to minimize the total weighted tardiness, and proved its NP-hardness and strong NP-hardness, respectively. Hall and Posner (1991) considered a single-machine case with a common due date to minimize the total weighted tardiness and earliness, and proved its NP-hardness. Hall et al. (1991), Gao and Yuan (2015), and Choi and Park (2019) considered a single-machine case to minimize the total earliness and tardiness. Hall et al. (1991) showed that it remains NP-hard, even if the due dates are identical, and Gao and Yuan (2015) proved its strong NP-hardness. In particular, Choi and Park (2019) showed that it remains strongly NP-hard, even if the lengths of the intervals between the consecutive due dates are identical. Note that such due dates are referred to as *periodic due dates (PDD)*. The PDD can be found in the just-in-time (JIT) business environment such that the logistics shipping means trucks periodically visit a factory according to a planned timetable (Lee and Li 1996). Choi and Park (2018) considered a single-machine case with PDD, in which the objective is to minimize the total weighted number of early and tardy jobs. They analyzed the computational complexity for various cases. Mosheiov and Oron (2004) considered the parallel-machine case to minimize the maximum tardiness or the total tardiness, and suggested that the simple heuristic was verified as performing extremely well through the numerical experiment.

The JIT production models usually used the same kind of penalty function for earliness and tardiness—both are linear or both are stepwise (Choi and Park 2018, 2019; Gao and Yuan 2015). In this paper, we consider different penalty functions—linear for earliness and stepwise for tardiness. This is motivated from the integrated production, inventory, and delivery scheduling problems, in which companies try to attain efficient inventory and logistics management through JIT production (Han et al. 2019; Li et al. 2017). In this environment, the manufacturer tries to minimize the total inventory cost and the penalty cost for tardy jobs. This can be described concretely as follows: consider trucks that periodically visit a manufacturer according to a planned timetable. The  $i$ th truck visiting the manufacturer is responsible for delivering the  $i$ th product. If the  $i$ th product is completed before the arrival of the  $i$ th truck, then the inventory costs for a manufacturer are incurred, whereas the penalty costs for tardy jobs are incurred, otherwise. Note that no cost occurs if the completion time of the  $i$ th product is equal to the arrival time of the  $i$ th truck. The inventory cost is usually expressed as a linear function of earliness because it incurs daily from production to delivery. Also, the penalty for tardy jobs is usually expressed as a step function of tardiness. See the references Han et al. (2019) and Li et al. (2017). To the best of our knowledge, this objective function has not yet been considered in the literature on scheduling with GDD. Since the asymmetry of the objective function makes it difficult to utilize the existing results on the JIT production in Han et al. (2019) and Li et al. (2017), it is necessary to reveal the computational complexity of our problem.

Our problem can be formally stated as follows: let  $p_j$  be the processing time of job  $J_j \in \mathcal{J} = \{J_1, J_2, \dots, J_n\}$ . Let  $\sigma = (\pi; \mathcal{S})$  be a schedule such that

- $\pi = (\pi(1), \pi(2), \dots, \pi(n))$  is the job sequence, where  $J_{\pi(i)}$  is the  $i$ th job completed;

- $\mathcal{S} = (S_1(\sigma), S_2(\sigma), \dots, S_n(\sigma))$  is the vector for the job starting times, where  $S_j(\sigma)$  is the starting time of  $J_j$  in  $\sigma$ .

Let  $C_j(\sigma)$  be the completion time of  $J_j$  in  $\sigma$ , that is,  $C_j(\sigma) = S_j(\sigma) + p_j$ . To describe PDD, let

$$d_i = i \Delta \text{ for } i = 1, 2, \dots, n, \tag{1}$$

be the due date of the job processed  $i$ th,  $i = 1, 2, \dots, n$ , where  $\Delta > 0$  is a given value. By relation (1),  $\mathcal{J}$  can be divided as follows:

$$\mathcal{J}_s = \{J_j \in \mathcal{J} \mid p_j \leq \Delta\}, \text{ and } \mathcal{J}_l = \mathcal{J} \setminus \mathcal{J}_s.$$

Let the jobs in  $\mathcal{J}_s$  and  $\mathcal{J}_l$  be referred to as *small* and *large* jobs, respectively. In  $\sigma = (\pi; \mathcal{S})$ , let  $E_{\pi(i)}(\sigma)$  and  $U_{\pi(i)}(\sigma)$  be the earliness and the tardy penalty of  $J_{\pi(i)} \in \mathcal{J}$ , respectively, which are defined as

$$E_{\pi(i)}(\sigma) = \max \{0, d_i - C_{\pi(i)}(\sigma)\} \tag{2}$$

and

$$U_{\pi(i)}(\sigma) = \begin{cases} 0 & \text{if } C_{\pi(i)}(\sigma) \leq d_{1,i}, \\ u_{h,i} & \text{if } d_{h,i} < C_{\pi(i)}(\sigma) \leq d_{h+1,i}, \quad h = 1, 2, \dots, g, \end{cases} \tag{3}$$

where  $g$  is a given constant. Assume that, for  $i = 1, 2, \dots, n$ ,

$$u_{1,i} \leq u_{2,i} \leq \dots \leq u_{g,i} \text{ and } d_i = d_{1,i} \leq d_{2,i} \leq \dots \leq d_{g+1,i} = \infty.$$

Note that the step function in (3) has been used in the field of transportation and semiconductor manufacturing (Chaurasia et al. 2017; Curry and Peters 2005; Detienne et al. 2011; Tseng and Chen 2013). The objective is to minimize

$$z(\sigma) = \sum_{J_j \in \mathcal{J}} (E_j(\sigma) + U_j(\sigma)).$$

Let our problem be referred to as *Problem P*.

The remainder of the paper is organized as follows. In Sect. 2, we prove the strong NP-hardness of Problem P. Section 3 introduces one polynomially solvable case. Finally, in Sect. 4, we present our concluding remarks.

## 2 Strong NP-hardness

In this section, we show that Problem P with PDD is strongly NP-hard, even if the penalty function for tardiness of each job has a single-step, that is,

$$U_{\pi(i)}(\sigma) = \begin{cases} 0 & \text{if } C_{\pi(i)}(\sigma) \leq d_i, \\ u_{1,i} & \text{otherwise.} \end{cases}$$

**Theorem 1** *Problem P is strongly NP-hard, even if the penalty function for tardiness of each job has a single-step.*

**Proof** To prove the strong NP-hardness, we use the numerical 3-dimensional matching (N3DM) problem, which is known to be strongly NP-hard (Garey and Johnson 1979). The N3DM can be stated as follows: given the disjoint sets

$$\mathcal{W} = \{w_1, w_2, \dots, w_m\}, \mathcal{X} = \{x_1, x_2, \dots, x_m\}, \text{ and } \mathcal{Y} = \{y_1, y_2, \dots, y_m\},$$

with each containing  $m$  elements, and a bound  $B$ , is there a partition  $(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m)$  of  $\mathcal{W} \cup \mathcal{X} \cup \mathcal{Y}$  such that for  $i = 1, 2, \dots, m$ ,

- Each  $\mathcal{A}_i = \{w_{a_i(1)}, x_{a_i(2)}, y_{a_i(3)}\}$  contains exactly one element from each of  $\mathcal{W}$ ,  $\mathcal{X}$ , and  $\mathcal{Y}$ ;
- $w_{a_i(1)} + x_{a_i(2)} + y_{a_i(3)} = B$ ?

Given an instance of the N3DM problem, we can construct an instance of Problem P as follows: there are  $4m$  jobs in  $\mathcal{J} = \{J_1, J_2, \dots, J_{4m}\}$  with

$$p_i = \begin{cases} M - (w_i + T) & \text{for } J_i \in \mathcal{J}_w = \{J_1, J_2, \dots, J_m\}, \\ M - (x_i + T^2) & \text{for } J_i \in \mathcal{J}_x = \{J_{m+1}, J_{m+2}, \dots, J_{2m}\}, \\ M - (y_i + T^3) & \text{for } J_i \in \mathcal{J}_y = \{J_{2m+1}, J_{2m+2}, \dots, J_{3m}\}, \\ M + T^3 + T^2 + T + B & \text{for } J_i \in \mathcal{J}_0 = \{J_{3m+1}, J_{3m+2}, \dots, J_{4m}\}, \end{cases}$$

where  $M = 3m(T^3 + T^2 + T + B) + 1$ , and  $T = 3mB$ . For  $i = 1, 2, \dots, 4m$ , let  $\Delta = M$  and  $u_{1,i} = M$ . Henceforth, we show that there exists a partition  $(\mathcal{A}_i)_{i=1,2,\dots,m}$  to the N3DM problem if and only if there exists a schedule  $\sigma$  with  $z(\sigma) \leq K$  for the reduced instance, where

$$K = 3 \sum_{i=1}^m (w_i + T) + 2 \sum_{i=1}^m (x_i + T^2) + \sum_{i=1}^m (y_i + T^3).$$

Note that  $M > K$ . □

( $\Rightarrow$ ) Suppose that there exists a partition  $(\bar{\mathcal{A}}_i)_{i=1,2,\dots,m}$  to the N3DM problem such that  $\bar{\mathcal{A}}_i = \{w_{\bar{a}_i(1)}, x_{\bar{a}_i(2)}, y_{\bar{a}_i(3)}\}$ ,  $i = 1, 2, \dots, m$ . Then, we can construct a schedule  $\bar{\sigma} = (\bar{\pi}, \bar{S})$  such that

- $\bar{\pi} = (\bar{\pi}_1, J_{3m+1}, \bar{\pi}_2, J_{3m+2}, \dots, \bar{\pi}_m, J_{4m})$ , where

$$\bar{\pi}_i = (J_{\bar{a}_i(1)}, J_{m+\bar{a}_i(2)}, J_{2m+\bar{a}_i(3)}) \text{ for } i = 1, 2, \dots, m,$$

is the sub-sequence consisting of three jobs in  $\{J_{\bar{a}_i(1)}, J_{m+\bar{a}_i(2)}, J_{2m+\bar{a}_i(3)}\}$  corresponding to three items in  $\bar{\mathcal{A}}_i = \{w_{\bar{a}_i(1)}, x_{\bar{a}_i(2)}, y_{\bar{a}_i(3)}\}$ , respectively;

- $S_{\bar{\pi}(1)}(\bar{\sigma}) = 0$  and  $S_{\bar{\pi}(i+1)}(\bar{\sigma}) = S_{\bar{\pi}(i)}(\bar{\sigma}) + p_{\bar{\pi}(i)}$  for  $i = 1, 2, \dots, 4m - 1$ .

Note that the  $4i$ th job in  $\bar{\pi}$  is  $J_{3m+i}$ . Since  $w_{\bar{a}_i(1)} + x_{\bar{a}_i(2)} + y_{\bar{a}_i(3)} = B$ ,

$$\sum_{j=4i-3}^{4i} p_{\bar{\pi}(j)} = p_{\bar{a}_i(1)} + p_{m+\bar{a}_i(2)} + p_{2m+\bar{a}_i(3)} + p_{3m+i} = 4M \text{ for } i = 1, 2, \dots, m. \tag{4}$$

By  $p_j < M$  for  $J_j \in \mathcal{J}_w \cup \mathcal{J}_x \cup \mathcal{J}_y$  and equation (4), the jobs in  $\mathcal{J}_w \cup \mathcal{J}_x \cup \mathcal{J}_y$  and  $\mathcal{J}_0$  become the early and JIT jobs, respectively, under  $\bar{\sigma}$ . Thus, for each  $i = 1, 2, \dots, m$ ,

$$E_{\bar{a}_i(1)}(\bar{\sigma}) = d_{4(i-1)+1} - \left(4(i-1)M + p_{\bar{a}_i(1)}\right) = w_{\bar{a}_i(1)} + T, \tag{5}$$

$$\begin{aligned} E_{m+\bar{a}_i(2)}(\bar{\sigma}) &= d_{4(i-1)+2} - \left(4(i-1)M + p_{\bar{a}_i(1)} + p_{m+\bar{a}_i(2)}\right) \\ &= w_{\bar{a}_i(1)} + x_{\bar{a}_i(2)} + T + T^2, \end{aligned} \tag{6}$$

and

$$\begin{aligned} E_{2m+\bar{a}_i(3)}(\bar{\sigma}) &= d_{4(i-1)+3} - \left(4(i-1)M + p_{\bar{a}_i(1)} + p_{m+\bar{a}_i(2)} + p_{2m+\bar{a}_i(3)}\right) \\ &= w_{\bar{a}_i(1)} + x_{\bar{a}_i(2)} + y_{\bar{a}_i(3)} + T + T^2 + T^3. \end{aligned} \tag{7}$$

By equations (4)–(7),

$$z(\bar{\sigma}) = 3 \sum_{i=1}^m (w_i + T) + 2 \sum_{i=1}^m (x_i + T^2) + \sum_{i=1}^m (y_i + T^3) = K.$$

( $\Leftrightarrow$ ) Suppose that there exists a schedule  $\hat{\sigma} = (\hat{\pi}; \hat{S})$  with

$$z(\hat{\sigma}) \leq K. \tag{8}$$

*Claim 1* No tardy job exists in  $\hat{\sigma}$ .

**Proof** Since the tardy penalty cost of every job is greater than  $K$ , Claim 1 holds.  $\square$

*Claim 2* There is no idle time in  $\hat{\sigma}$ .

**Proof** Since  $\sum_{j=1}^{4m} p_j = 4mM$  and  $d_{4m} = 4mM$ , the existence of the idle time implies that the last job is tardy in  $\hat{\sigma}$ . Thus, Claim 2 holds immediately from Claim 1.  $\square$

By Claim 2, it is observed that

$$S_{\hat{\pi}(1)}(\hat{\sigma}) = 0 \text{ and } S_{\hat{\pi}(i+1)}(\hat{\sigma}) = S_{\hat{\pi}(i)}(\hat{\sigma}) + p_{\hat{\pi}(i)} \text{ for } i = 1, 2, \dots, 4m - 1.$$

If the last job does not belong to  $\mathcal{J}_0$ , then the second job from the last is tardy by Claim 2. Since this contradicts Claim 1, the last job belongs to  $\mathcal{J}_0$ . Without loss of generality, assume that, in  $\hat{\sigma}$ ,  $J_{3m+i}$  is processed before  $J_{3m+j}$  for  $j > i$ . Then,  $\hat{\pi}$  can be represented as

$$\hat{\pi} = (\hat{\pi}_1, J_{3m+1}, \hat{\pi}_2, J_{3m+2}, \dots, \hat{\pi}_m, J_{4m}),$$

where  $\hat{\pi}_1$  is the sub-sequence of jobs before  $J_{3m+1}$ , and  $\hat{\pi}_i$  is the sub-sequence of jobs between  $J_{3m+i-1}$  and  $J_{3m+i}$ ,  $i = 2, 3, \dots, m$ . Since every job is not tardy from Claim 1, every small job  $J_{\hat{\pi}(i)} \in \mathcal{J}_w \cup \mathcal{J}_x \cup \mathcal{J}_y$  satisfies

$$E_{\hat{\pi}(i)}(\hat{\sigma}) \geq \Delta - p_{\hat{\pi}(i)}, \tag{9}$$

and

$$E_{\hat{\pi}(i)}(\hat{\sigma}) = E_{\hat{\pi}(i-1)}(\hat{\sigma}) + (\Delta - p_{\hat{\pi}(i)}). \tag{10}$$

Then, we have the following claim: □

*Claim 3* Exactly one job in  $\mathcal{J}_w$ ,  $\mathcal{J}_x$ , and  $\mathcal{J}_y$  should be processed in  $\hat{\pi}_i$ ,  $i = 1, 2, \dots, m$ .

**Proof** See Appendix A. □

By Claim 3, let

$$\hat{\pi}_i = (J_{\hat{a}_i(1)}, J_{m+\hat{a}_i(2)}, J_{2m+\hat{a}_i(3)}).$$

It is observed from Claim 3 and inequalities (9) and (10) that in  $\hat{\sigma} = (\hat{\pi}; \hat{S})$ , for  $i = 1, 2, \dots, m$ ,

- $E_{\hat{a}_i(1)}(\hat{\sigma}) \geq w_{\hat{a}_i(1)} + T$ ;
- $E_{m+\hat{a}_i(2)}(\hat{\sigma}) \geq w_{\hat{a}_i(1)} + x_{m+\hat{a}_i(2)} + T + T^2$ ;
- $E_{2m+\hat{a}_i(3)}(\hat{\sigma}) \geq w_{\hat{a}_i(1)} + x_{m+\hat{a}_i(2)} + y_{2m+\hat{a}_i(3)} + T + T^2 + T^3$ .

By the observation above,

$$z(\hat{\sigma}) \geq \sum_{J_j \in \mathcal{J}_w \cup \mathcal{J}_x \cup \mathcal{J}_y} E_j(\hat{\sigma}) \geq K. \tag{11}$$

By inequalities (8) and (11), each job  $J_{3m+i} \in \mathcal{J}_0$  is a JIT job. Then, it is observed that

$$w_{\hat{a}_i(1)} + x_{\hat{a}_i(2)} + y_{\hat{a}_i(3)} = B \text{ for } i = 1, 2, \dots, m.$$

Let  $\hat{\mathcal{A}}_i = \{w_{\hat{a}_i(1)}, x_{\hat{a}_i(2)}, y_{\hat{a}_i(3)}\}$ ,  $i = 1, 2, \dots, m$ . Then, the partition  $(\hat{\mathcal{A}}_i)_{i=1,2,\dots,m}$  becomes a solution to the N3DM problem. □

Since the single-step model is a special case of the multi-step one, the strong NP-hardness result of the single-step model holds in the multi-step one.

### 3 Polynomiality

In this section, we show that Problem P can be solved in polynomial time if the processing times of the small and large jobs are identical.

First, we introduce some optimality properties that will be used to construct a polynomial-time algorithm. For simplicity, let  $J_{\pi(i)}$  be referred to as a *key job* under  $\sigma$ , if

$$C_{\pi(i)}(\sigma) \in \{d_{h,i} \mid h = 1, 2, \dots, g\}.$$

Suppose that  $\sigma^*$  is an optimal schedule with the greatest number of key jobs, if any, among the multiple optimal schedules. Let  $\mathcal{A}_j$  and  $\mathcal{B}_j$  be the sets of jobs processed after and before  $J_j$  in  $\sigma^*$ , respectively.

**Lemma 1** *If job  $J_l$  is the last key job in  $\sigma^*$ , then every job in  $\mathcal{A}_l$  is tardy, and there is no idle time after  $J_l$ .*

**Proof** Suppose that an early job exists in  $\mathcal{A}_l$ . Then, we can construct a new schedule  $\bar{\sigma} = (\pi^*; \bar{\mathcal{S}})$  by letting  $S_j(\bar{\sigma}) = S_j(\sigma^*) + \epsilon$  for each  $J_j \in \mathcal{A}_l$ , where  $\epsilon > 0$  is a sufficiently small value. Then, since

$$\sum_{J_j \in \mathcal{J}} E_j(\bar{\sigma}) < \sum_{J_j \in \mathcal{J}} E_j(\sigma^*), \quad \text{and} \quad \sum_{J_j \in \mathcal{J}} U_j(\bar{\sigma}) = \sum_{J_j \in \mathcal{J}} U_j(\sigma^*),$$

we have  $z(\bar{\sigma}) < z(\sigma^*)$ . Since this is a contradiction, every job in  $\mathcal{A}_l$  is tardy. In turn, the idle time after  $J_l$  is unnecessary. □

By Lemma 1, if no key job exists in  $\sigma^*$ , then every job is tardy, and no idle time exists. Hence, a sequence with a non-increasing order of  $p_j$  is an optimal schedule. To exclude this trivial case, we assume that  $\sigma^*$  has at least one key job.

**Lemma 2** *If  $J_{\pi^*(k)}$  and  $J_{\pi^*(l)}$  are the consecutive key jobs in an optimal schedule  $\sigma^*$ , then no idle time exists between jobs  $J_{\pi^*(k)}$  and  $J_{\pi^*(l)}$ .*

**Proof** Suppose that the idle time first exists between jobs  $J_{\pi^*(i')}$  and  $J_{\pi^*(i'+1)}$  for some  $i' \in \{k + 1, k + 2, \dots, l - 1\}$  in the sub-schedule consisting of the jobs in  $\{J_{\pi^*(k+1)}, J_{\pi^*(k+2)}, \dots, J_{\pi^*(l)}\}$ . Consider two cases. □

(i) An early job exists in  $\{J_{\pi^*(k+1)}, J_{\pi^*(k+2)}, \dots, J_{\pi^*(i')}\}$ . We can construct a new schedule  $\bar{\sigma} = (\pi^*; \bar{\mathcal{S}})$  by letting  $S_{\pi^*(i)}(\bar{\sigma}) = S_{\pi^*(i)}(\sigma^*) + \epsilon$  for  $i = k + 1, k + 2, \dots, i'$ , where  $\epsilon > 0$  is a sufficiently small value. Then,

$$z(\bar{\sigma}) \leq z(\sigma^*) - \epsilon < z(\sigma^*).$$

This is a contradiction.

(ii) No early job exists in  $\{J_{\pi^*(k+1)}, J_{\pi^*(k+2)}, \dots, J_{\pi^*(i')}\}$ . We can construct a new schedule  $\hat{\sigma} = (\pi^*; \hat{\mathcal{S}})$  by letting  $S_{\pi^*(i)}(\hat{\sigma}) = S_{\pi^*(i)}(\sigma^*) + \delta$  for  $i = k + 1, k + 2, \dots, i'$ , where  $\delta$  is the idle time between jobs  $J_{\pi^*(i')}$  and  $J_{\pi^*(i'+1)}$ . From the assumption that  $\sigma^*$  is an optimal schedule with the greatest number of key jobs,

$$U(\hat{\sigma}) = U(\sigma^*),$$

which implies that  $z(\hat{\sigma}) = z(\sigma^*)$ . By applying this argument repeatedly, Lemma 2 is obtained.

From two cases, Lemma 2 holds. □

**Theorem 2** *Problem P can be solved in strongly polynomial time  $O(n^6)$  if  $p_j = p_s$  for  $J_j \in \mathcal{J}_s$  and  $p_j = p_l$  for  $J_j \in \mathcal{J}_l$ .*

**Proof** For simplicity, let

$$\mathcal{J}_s = \{J_1, J_2, \dots, J_{n_s}\} \text{ and } \mathcal{J}_l = \{J_{n_s+1}, J_{n_s+2}, \dots, J_n\}.$$

We prove this theorem by reducing it to the shortest path (SP) problem, which can be stated as follows: in a directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with a node set  $\mathcal{V} = \{1, 2, \dots, n\}$  and an edge set  $\mathcal{E}$  such that each arc  $(i, j) \in \mathcal{E}$  has a length  $l_{i,j}$ , the objective is to find the shortest path from the source to the sink nodes.

For  $0 \leq a \leq n_s, 0 \leq b \leq n_l, 1 \leq h \leq g$  such that  $d_{h,a+b} \geq a \cdot p_s + b \cdot p_l$ , let  $N(a, b, h)$  be the node representing sub-schedule such that

- The jobs in  $\mathcal{J}_{a,b}$  have been scheduled, where

$$\mathcal{J}_{a,b} = \{J_1, J_2, \dots, J_a\} \cup \{J_{n_s+1}, J_{n_s+2}, \dots, J_{n_s+b}\};$$

- The last job in  $\mathcal{J}_{a,b}$  is a key job whose completion time and tardy penalty are  $d_{h,a+b}$  and  $u_{h-1,a+b}$ , respectively. Note that for consistency of notation, let  $d_{h,0} = 0$  and  $u_{0,a+b} = 0$ .

Let  $N(0, 0, 1)$  and  $t$  be the source and sink nodes, respectively. For  $0 \leq \alpha \leq n_s - a, 0 \leq \beta \leq n_l - b$  and  $1 \leq \gamma \leq g$ , let  $N(a, b, h)$  be connected to  $N(a + \alpha, b + \beta, \gamma)$  if  $\alpha + \beta \geq 1$  and  $d_{h,a+b} \leq S$ , where

$$S = d_{\gamma,a+b+\alpha+\beta} - \alpha \cdot p_s - \beta \cdot p_l.$$

Note that  $S$  is the start time of the first job in  $\mathcal{J}_{a+\alpha,b+\beta} \setminus \mathcal{J}_{a,b}$  by Lemma 2. This edge shows that the last jobs in  $\mathcal{J}_{a,b}$  and  $\mathcal{J}_{a+\alpha,b+\beta}$  are two consecutive key jobs at  $d_{h,a+b}$  and at  $d_{\gamma,a+b+\alpha+\beta}$ , respectively. The length of the edge from  $N(a, b, h)$  to  $N(a + \alpha, b + \beta, \gamma)$ , denoted  $L(e; \alpha, \beta; S)$ , is calculated as follows, where  $e = a + b$ .

*Procedure*  $L(e; \alpha, \beta; S)$

*Step 1* (Initialization) Set  $L_{0,0} = 0$  and  $L_{i,j} = \infty$  if  $i < 0$  or  $j < 0$ .

*Step 2* For  $i = 0, 1, \dots, \alpha$  and  $j = 0, 1, \dots, \beta$ ,

$$L_{i,j} = \min \{L_{i-1,j}, L_{i,j-1}\} + E_{\pi(e+i+j)}(\sigma) + U_{\pi(e+i+j)}(\sigma).$$

Note that  $E_{\pi(e+i+j)}(\sigma)$  and  $U_{\pi(e+i+j)}(\sigma)$  are calculated by Eqs. (2) and (3) with

$$C_{\pi(e+i+j)}(\sigma) = S + i \cdot p_s + j \cdot p_l, \text{ for } 0 \leq i \leq \alpha \text{ and } 0 \leq j \leq \beta.$$



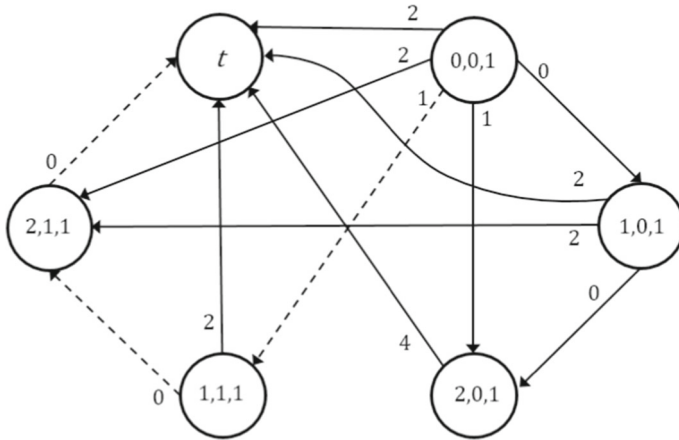


Fig. 1 Reduced graph for Example 1

Step 3 Return  $L(e; \alpha, \beta; S) = L_{\alpha,\beta}$ .

Note that the above procedure can be done in  $O(n^2)$ . Finally, for  $0 \leq a \leq n_s$ ,  $0 \leq b \leq n_l$  and  $1 \leq h \leq g$ , let  $N(a, b, h)$  be connected to the sink node  $t$ , which shows that the last job in  $\mathcal{J}_{a,b}$  is the last key job at  $d_{h,a+b}$  in the schedule. The length of this edge is  $L(a + b; n_s - a, n_l - b; \bar{S})$ , where  $\bar{S} = d_{h,a+b}$  is the start time of the first job in  $\mathcal{J} \setminus \mathcal{J}_{a,b}$  by Lemma 1. The objective is to find the shortest path between the source and sink nodes. See the reduced SP for Example 1 in Fig. 1 and calculating the length of edge from  $N(0, 0, 1)$  to  $N(2, 1, 1)$  in Fig. 2. Note that the shortest path consists of dashed edges. It is observed from the reduction scheme above that in the constructed SP problem,

- The total number of arcs is  $O(g^2n^4)$ , since the number of nodes is  $O(gn^2)$ , and the number of arcs coming from each node is at most  $O(gn^2)$ ;
- The length of each arc can be obtained in  $O(n^2)$  by Procedure L, which implies that the reduced graph is constructed in  $O(g^2n^6)$ ;
- The reduced graph is acyclic, which implies that the corresponding SP problem can be solved in  $O(g^2n^4)$  by the algorithm in Ahuja et al. (1990).

By these observations and the fact that  $g$  is a constant, Theorem 2 holds. □

**Example 1** An instance of Problem P is as follows:

- $n_s = 2, n_l = 1, p_s = 3, p_l = 7$  and  $\Delta = 5$ ;
- $(d_{1,1}, d_{1,2}, d_{1,3}) = (5, 10, 15)$ ;
- $(u_{1,1}, u_{1,2}, u_{1,3}) = (1, 3, 4)$ .

### 4 Concluding remarks and future works

We considered a single-machine scheduling problem with periodic due dates in which the objective is to minimize the total early and tardy penalties. Assume that the early

$C_{\pi(i+j)}(\sigma)$			$E_{\pi(i+j)}(\sigma)$			$U_{\pi(i+j)}(\sigma)$			$L_{i,j}$		
$i \setminus j$	0	1	$i \setminus j$	0	1	$i \setminus j$	0	1	$i \setminus j$	0	1
0		9	0		0	0		1	0	0	1
1	5	12	1	0	0	1	0	3	1	0	3
2	8	15	2	2	0	2	0	0	2	2	2

Fig. 2 Procedure  $L(0; 2, 1; 2)$  for the edge from  $N(0, 0, 1)$  to  $N(2, 1, 1)$

and tardy penalties increase according to the linear and step functions, respectively. We showed that the problem is strongly NP-hard, and can be solved in strongly polynomial time if the processing times of the small and large jobs are identical.

For future research, it would be interesting to analyze the computational complexity of the problem such that each job has an identical step function for tardiness.

**Funding** This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2018S1A5B8070344).

**Availability of data and material** Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

**Declarations**

**Conflict of interest** The authors have no conflicts of interest to declare that are relevant to the content of this article.

**Appendix A. The proof of Claim 3 in Theorem 1**

(i) Suppose that two jobs  $J_k$  and  $J_l$  in  $\mathcal{J}_y$  are processed in  $\hat{\pi}_i$  for some  $i \in \{1, 2, \dots, m\}$ . We assume that  $J_l$  is processed later than  $J_k$ . Then, by inequalities (9) and (10), we have

$$E_l(\hat{\sigma}) \geq E_k(\hat{\sigma}) + (\Delta - p_l) \geq (\Delta - p_k) + (\Delta - p_l) > 2T^3. \tag{12}$$

By inequality (9) and  $p_j < M - T^3$  for  $J_j \in \mathcal{J}_y$ , we have

$$E_j(\hat{\sigma}) > T^3 \text{ for } J_j \in \mathcal{J}_y. \tag{13}$$

By inequalities (12) and (13),

$$z(\hat{\sigma}) \geq \sum_{J_j \in \mathcal{J}_y} E_j(\hat{\sigma}) > (m + 1)T^3 > K.$$

This is a contradiction. Thus, we have the following result.

*Result 1* Exactly one job in  $\mathcal{J}_y$  is processed in  $\hat{\pi}_i$ .

For simplicity, let  $J_{2m+\hat{a}_i(3)}$  be that job. By equation (10), furthermore, the later the small job is processed, the better. Thus, since  $J_{2m+\hat{a}_i(3)}$  has the smallest processing time in  $\hat{\pi}_i$ , it is processed last.

(ii) Suppose that two jobs  $J_k$  and  $J_l$  in  $\mathcal{J}_x$  are processed in  $\hat{\pi}_i$  for some  $i \in \{1, 2, \dots, m\}$ . We assume that  $J_l$  is processed later than  $J_k$ . Then, by inequalities (9) and (10), we have

$$E_l(\hat{\sigma}) \geq E_k(\hat{\sigma}) + (\Delta - p_l) \geq (\Delta - p_k) + (\Delta - p_l) > 2T^2. \tag{14}$$

By inequality (9) and  $p_j < M - T^2$  for  $J_j \in \mathcal{J}_x$ , we have

$$E_j(\hat{\sigma}) > T^2 \text{ for } J_j \in \mathcal{J}_x. \tag{15}$$

By inequalities (14) and (15),

$$\sum_{J_j \in \mathcal{J}_x} E_j(\hat{\sigma}) > (m + 1)T^2. \tag{16}$$

Also, by inequalities (9) and (10) and Result 1, we have

$$E_{2m+\hat{a}_i(3)}(\hat{\sigma}) > n_i T^2 + T^3 \text{ for } J_{2m+\hat{a}_i(3)} \in \mathcal{J}_y, \tag{17}$$

where  $n_i$  is the number of jobs of  $\mathcal{J}_x$  in  $\hat{\pi}_i$ . By Result 1, and inequalities (16) and (17),

$$z(\hat{\sigma}) \geq \sum_{J_j \in \mathcal{J}_x} E_j(\hat{\sigma}) + \sum_{i=1}^m E_{2m+\hat{a}_i(3)}(\hat{\sigma}) > (m + 1)T^2 + \sum_{i=1}^m (n_i T^2 + T^3) > K.$$

This is a contradiction. Thus, we have the following result.

**Result 2** Exactly one job in  $\mathcal{J}_x$  is processed in  $\hat{\pi}_i$ .

For simplicity, let  $J_{m+\hat{a}_i(2)}$  be that job. Since  $J_{2m+\hat{a}_i(3)}$  has the smallest processing time in  $\hat{\pi}_i$ , it is processed immediately before job  $J_{m+\hat{a}_i(2)}$ .

(iii) Suppose that two jobs  $J_k$  and  $J_l$  in  $\mathcal{J}_w$  are processed in  $\hat{\pi}_i$  for some  $i \in \{1, 2, \dots, m\}$ . We assume that  $J_l$  is processed later than  $J_k$ . Then, by inequality (9) and (10), we have

$$E_l(\hat{\sigma}) \geq E_k(\hat{\sigma}) + (\Delta - p_l) \geq (\Delta - p_k) + (\Delta - p_l) > 2T. \tag{18}$$

By inequality (9) and  $p_j < M - T$  for  $J_j \in \mathcal{J}_w$ , we have

$$E_j(\hat{\sigma}) > T \text{ for } J_j \in \mathcal{J}_w. \tag{19}$$

By inequalities (18) and (19),

$$\sum_{J_j \in \mathcal{J}_w} E_j(\hat{\sigma}) > (m+1)T. \quad (20)$$

Also, by inequalities (9) and (10), and Results 1 and 2, we have

$$E_{m+\hat{a}_i(2)}(\hat{\sigma}) > l_i T + T^2, \text{ for } J_{m+\hat{a}_i(2)} \in \mathcal{J}_x, \quad (21)$$

and

$$E_{2m+\hat{a}_i(3)}(\hat{\sigma}) > l_i T + T^2 + T^3, \text{ for } J_{2m+\hat{a}_i(3)} \in \mathcal{J}_y, \quad (22)$$

where  $l_i$  is the number of jobs of  $\mathcal{J}_w$  in  $\hat{\pi}_i$ . By Results 1 and 2, and inequalities (20)–(22),

$$\begin{aligned} z(\hat{\sigma}) &\geq \sum_{J_j \in \mathcal{J}_w} E_j(\hat{\sigma}) + \sum_{i=1}^m (E_{m+\hat{a}_i(2)}(\hat{\sigma}) + E_{2m+\hat{a}_i(3)}(\hat{\sigma})) \\ &> (m+1)T + \sum_{i=1}^m (2l_i T + 2T^2 + T^3) > K. \end{aligned}$$

This is a contradiction. Thus, we have the following result.

**Result 3** Exactly one job in  $\mathcal{J}_w$  is processed in  $\hat{\pi}_i$ .

For simplicity, let  $J_{\hat{a}_i(1)}$  be that job. Thus, by Results 1–3, Claim 3 holds.  $\square$

## References

- Ahuja RA, Mehlhorn K, Orlin JB (1990) Faster algorithm for the shortest path problem. *J ACM* 37:213–223
- Chaurasia SN, Sundar S, Singh A (2017) Hybrid metaheuristic approaches for the single machine total stepwise tardiness problem with release dates. *Oper Res* 17:275–295
- Choi BC, Park MJ (2018) Just-in-time scheduling with generalized due dates and identical due date intervals. *Asia-Pacific J Oper Res* 35:1850046
- Choi BC, Park MJ (2019) Strong NP-hardness of minimizing total deviation with generalized and periodic due dates. *Oper Res Lett* 47:433–437
- Curry J, Peters B (2005) Rescheduling parallel machines with stepwise increasing tardiness. *Int J Prod Res* 43:3231–3246
- Detienne B, Dauzère-Pères S, Yugma C (2011) Scheduling jobs on parallel machines to minimize a regular step total cost function. *J Sched* 14:523–538
- Garey MR, Johnson DS (1979) *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco
- Gao Y, Yuan J (2015) Unary NP-hardness of minimizing the total deviation with generalized or assignable due dates. *Discrete Appl Math* 189:49–52
- Gao Y, Yuan J (2016) Unary NP-hardness of minimizing total weighted tardiness with generalized due dates. *Oper Res Lett* 44:92–95
- Hall NG (1986) Scheduling problems with generalized due dates. *IEE Trans* 18:220–222
- Hall NG, Posner ME (1991) Earliness-tardiness scheduling problems I: weighted deviation of completion times about a common due date. *Oper Res* 39:836–846
- Hall NG, Kubiak W, Sethi SP (1991) Earliness-tardiness scheduling problems II: deviation of completion times about a restrictive common due date. *Oper Res* 39:847–856

- Hall NG, Sethi SP, Srikandarajah S (1991) On the complexity of generalized due date scheduling problems. *Eur J Oper Res* 51:100–109
- Hall NG, Lesaoana M, Potts CN (2001) Scheduling with fixed delivery dates. *Oper Res* 49:134–144
- Han D, Yang Y, Wang D, Cheng TCE, Yin Y (2019) Integrated production, inventory, and outbound distribution operations with fixed departure times in a three-stage supply chain. *Transp Res Part E Logist Transp Rev* 125:334–347
- Lee CY, Li CL (1996) On the fixed interval due-date scheduling problem. *Discrete Appl Math* 68(1–2):101–117
- Li F, Chen ZL, Tang L (2017) Integrated production, inventory and delivery problems: complexity and algorithms. *INFORMS J Comput* 29(2):232–250
- Mosheiov G, Oron D (2004) A note on the SPT heuristic for solving scheduling problems with generalized due dates. *Comput Oper Res* 31:645–655
- Srikandarajah S (1990) A note on the generalized due dates scheduling problem. *Naval Res Logist* 37:587–597
- Tseng CT, Chen KH (2013) An electromagnetism-like mechanism for the single machine total stepwise tardiness problem with release dates. *Eng Optim* 45:1431–1448
- Yang X (2000) Scheduling with generalized batch delivery dates and earliness penalties. *IIE Trans* 32:735–741

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.