UNIVERSITY OF LEEDS

This is a repository copy of *A note on reverse scheduling with maximum lateness objective*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/79237/

Version: Accepted Version

Article:

Li, SS, Brucker, P, Ng, CT et al. (2 more authors) (2013) A note on reverse scheduling with maximum lateness objective. Journal of Scheduling, 16. 417 - 422. ISSN 1094-6136

https://doi.org/10.1007/s10951-013-0314-4

Reuse
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose
university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

A Note on Reverse Scheduling with Maximum Lateness Objective

S.S. Li [a], P. Brucker [b], C.T. Ng [c], T.C.E. Cheng [c], N.V. Shakhlevich [d], J.J. Yuan [e,*]

[a]College of Science, Zhongyuan University of Technology, Zhengzhou 450007, PR China

[b]University of Osnabrueck, Faculty of Mathematics/Informatics, 49069 Osnabrueck, Germany

[c]Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

[d]School of Computing, University of Leeds, Leeds, LS2 9JT, U.K.

[e]Department of Mathematics, Zhengzhou University, Zhengzhou 450001, PR China

Abstract

The inverse and reverse counterparts of the single-machine scheduling problem $1||L_{\max}$ are studied in [2], in which the complexity classification is provided for various combinations of adjustable parameters (due dates and processing times) and for five different types of norm: $\ell_1, \ell_2, \ell_\infty, \ell_H^\Sigma$, and $\ell_H^{\max}$. It appears that the $O(n^2)$-time algorithm for the reverse problem with adjustable due dates contains a flaw. In this note we present the structural properties of the reverse model, establishing a link with the forward scheduling problem with due dates and deadlines. For the four norms $\ell_1, \ell_\infty, \ell_H^\Sigma$, and $\ell_H^{\max}$, the complexity results are derived based on the properties of the corresponding forward problems, while the case of the norm ℓ_2 is treated separately. As a by-product, we resolve an open question on the complexity of problem $1||\sum \alpha_j T_j^2$.

Keywords: Reverse scheduling; Maximum lateness

In this note we consider one of the models studied by Brucker and Shakhlevich [2] in the context of inverse/reverse optimization. The model deals with the reverse version of problem $1||L_{\max}$. Unlike the traditional forward problem, in which the exact values of the due dates are given for all the jobs and the objective is to find a job permutation minimizing the maximum lateness, in the reverse version typical values of the due dates are given and they are to be modified in order to achieve a target value of maximum lateness.

Formally, in the reverse version of problem $1||L_{\max}$, the jobs in the job set $\mathcal{N} = \{1, 2, \ldots, n\}$ are available at time 0 for processing on a single machine. Associated with each job $j \in \mathcal{N}$ are two main characteristics, namely the processing time p_j and due date d_j, both of which are integers. In a schedule induced by a job permutation π, the jobs are scheduled consecutively without idle time and their completion times are denoted by $C_j(\pi)$, $j \in \mathcal{N}$. The lateness of job j is defined as $L_j(\pi, \mathbf{d}) = C_j(\pi) - d_j$ and the maximum lateness is $L_{\max}(\pi, \mathbf{d}) = \max\{L_j(\pi, \mathbf{d}) | j \in \mathcal{N}\}$. It is required that the maximum lateness does not exceed a given target value L^*. In order to achieve the target value, one has to find an optimal permutation π and adjusted due dates $\widehat{d}_j$ belonging to

*Corresponding author. *E-mail address*: yuanjj@zzu.edu.cn (J.J. Yuan).

their variability intervals $[\underline{d}_j, \overline{d}_j]$, $\underline{d}_j \geq 0$, so that the adjustment cost $\|\widehat{\mathbf{d}} - \mathbf{d}\|$ is minimum. Thus the reverse problem can be formulated as

$$\begin{array}{lll} \text{Problem R:} & \min & \|\widehat{\mathbf{d}} - \mathbf{d}\| \\ & \text{s.t.} & L_{\max}(\pi, \widehat{\mathbf{d}}) \leq L^* \quad \text{for some permutation } \pi, \\ & & \underline{d}_j \leq \widehat{d}_j \leq \overline{d}_j. \end{array}$$

Notice that for each $j \in \mathcal{N}$, its initial due date d_j belongs to $[\underline{d}_j, \overline{d}_j]$. Clearly, there exists an optimal solution with $\widehat{d}_j \geq d_j$, $j \in \mathcal{N}$, so in what follows we consider the due date boundaries $[d_j, \overline{d}_j] \subseteq [\underline{d}_j, \overline{d}_j]$.

The deviation $||\widehat{\mathbf{d}} - \mathbf{d}||$ is calculated in accordance with one of the following norms:

$$\begin{array}{ll} \ell_1 \text{ (Manhattan):} & \|\widehat{\mathbf{d}} - \mathbf{d}\|_{1,\alpha} = \sum_{j=1}^n \alpha_j \left(\widehat{d}_j - d_j\right), \\ \ell_2 \text{ (Euclidean):} & \|\widehat{\mathbf{d}} - \mathbf{d}\|_{2,\alpha} = \sqrt{\sum_{j=1}^n \alpha_j \left(\widehat{d}_j - d_j\right)^2}, \\ \ell_\infty: & \|\widehat{\mathbf{d}} - \mathbf{d}\|_{\infty,\alpha} = \max_{j=1,\dots,n} \left\{\alpha_j \left(\widehat{d}_j - d_j\right)\right\}, \\ \ell_H^\Sigma \text{ (Hamming):} & \|\widehat{\mathbf{d}} - \mathbf{d}\|_{H,\alpha}^\Sigma = \sum_{j=1}^n \alpha_j \mathrm{sgn}\left(\widehat{d}_j - d_j\right), \\ \ell_H^{\max} \text{ (Hamming):} & \|\widehat{\mathbf{d}} - \mathbf{d}\|_{H,\alpha}^{\max} = \max_{j=1,\dots,n} \left\{\alpha_j \mathrm{sgn}\left(\widehat{d}_j - d_j\right)\right\}, \end{array}$$

where all the coefficients α_j are non-negative.

It is stated in [2] that the reverse problem R can be solved in $O(n^2)$ time by an algorithm that iteratively increases the due dates of the critical jobs in the earliest due date (EDD) schedule. However, the proposed algorithm has a flaw, as can be seen from a two-job counter-example with parameters $p_1 = d_1 = 1$, $p_2 = d_2 = 2$, $d_1 \in [0, 10]$, $d_2 \in [0, 10]$, $\alpha_1 = 1$, $\alpha_2 = 100$, and $L^* = 0$. In this note we fix the flaw by reducing problem R to a forward scheduling problem with due dates and deadlines and by exploiting properties of that problem.

Lemma 1 *Depending on the type of the norm $\left\|\widehat{\mathbf{d}} - \mathbf{d}\right\|$, the reverse problem R is equivalent to one of the following forward scheduling problems:*

(A) $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''|\sum \alpha_j T_j$, *if norm $\ell_{1,\alpha}$ is used;*
(B) $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''|\sqrt{\sum \alpha_j T_j^2}$, *if norm $\ell_{2,\alpha}$ is used;*
(C) $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''|\max\{\alpha_j T_j\}$, *if norm $\ell_{\infty,\alpha}$ is used;*
(D) $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''|\sum \alpha_j U_j$, *if norm $\ell_{H,\alpha}^\Sigma$ is used;*
(E) $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''|\max\{\alpha_j U_j\}$, *if norm $\ell_{H,\alpha}^{\max}$ is used*

where the due dates $\mathbf{d}'$ and deadlines $\mathbf{d}''$ are defined as

$$\mathbf{d}' = \mathbf{d} + L^*, \tag{1}$$
$$\mathbf{d}'' = \overline{\mathbf{d}} + L^*. \tag{2}$$

Notice that in the equivalent forward problems, the parameters p_j and α_j, $j \in \mathcal{N}$, are the same as those in the reverse problem R.

Proof. We present a proof of Case (A) by establishing a one-to-one correspondence between a solution to the reverse problem R given by a job permutation π and a solution to problem A given by the same permutation π. In particular, we show that if π is feasible for the reverse problem, it is also feasible for problem A, and vice versa. Moreover, the optimum objective value for a fixed π is the same for both problems. This implies that the two problems are equivalent and the global optimal solutions are the same. Note that for both problems we can consider left-shifted schedules given by π.

Permutation π is feasible for the reverse problem R if there exist adjusted due dates $\widehat{d}_j$, $j \in \mathcal{N}$, for which the target $L_{\max}$-value is achieved, and the due date boundaries are satisfied:

$$C_j(\pi) - \widehat{d}_j \le L^*, \tag{3}$$

$$\underline{d}_j \le \widehat{d}_j \le \overline{d}_j. \tag{4}$$

Permutation π is feasible for problem A if the job completion times under π do not exceed their deadlines $d''_j = \overline{d}_j + L^*$ for all $j \in \mathcal{N}$:

$$C_j(\pi) \le \overline{d}_j + L^*. \tag{5}$$

Clearly, if (3)-(4) hold for the reverse problem, then (5) is satisfied for problem A.

Alternatively, if (5) holds for problem A, then by setting $\widehat{d}_j$ in the reverse problem as

$$\widehat{d}_j = \max\left\{d_j, C_j(\pi) - L^*\right\}, \ \ j \in \mathcal{N}, \tag{6}$$

we obtain a solution to the reverse problem satisfying conditions (3)-(4). Indeed, condition (3) and the left-hand-side of condition (4) immediately follow from (6). To show that the right-hand side of (4) holds, we observe that (5) implies $C_j(\pi) - L^* \le \overline{d}_j$, and together with $d_j \le \overline{d}_j$, we get

$$\widehat{d}_j = \max\left\{d_j, C_j(\pi) - L^*\right\} \le \overline{d}_j.$$

Denote by $F_{\mathrm{R}}(\pi)$ and $F_{\mathrm{A}}(\pi)$ the optimal objective values of the reverse problem R and of problem A, respectively, under the assumption that the job permutation π is fixed. For the reverse problem R with a fixed π, the optimal adjusted due dates are given by (6) since any larger values of $\widehat{d}_j$ are non-optimal: they can be reduced without violating (3)-(4), leading to a smaller adjustment cost. Hence

$$F_{\mathrm{R}}(\pi) = \sum_{j\in\mathcal{N}} \alpha_j \left(\widehat{d}_j - d_j\right) = \sum_{j\in\mathcal{N}} \alpha_j \max\left\{0, C_j(\pi) - L^* - d_j\right\}.$$

For problem A with a fixed π,

$$F_{\mathrm{A}}(\pi) = \sum_{j\in\mathcal{N}} \alpha_j T_j = \sum_{j\in\mathcal{N}} \alpha_j \max\left\{C_j(\pi) - d'_j, 0\right\} = \sum_{j\in\mathcal{N}} \alpha_j \max\left\{C_j(\pi) - d_j - L^*, 0\right\}.$$

Thus $F_{\mathrm{R}}(\pi) = F_{\mathrm{A}}(\pi)$ and case (A) is proved. The proofs of cases (B)-(E) are similar. ∎

The theorem below is based on a one-to-one correspondence between the solutions to the reverse problem R and the solutions to problems A-E.

Theorem 1 *Depending on the type of the norm $\|\widehat{\mathbf{d}} - \mathbf{d}\|$, the reverse problem R is*

(A) *NP-hard in the strong sense if norm $\ell_{1,\alpha}$ is used and NP-hard in the ordinary sense if the unit-weight norm ℓ_1 is used ($\alpha_j = 1$ for all $j \in \mathcal{N}$);*

(B) *NP-hard in the strong sense if norm $\ell_{2,\alpha}$ is used;*

(C) *solvable in $O(n \log n)$ time if norm $\ell_{\infty,\alpha}$ is used;*

(D) *NP-hard in the ordinary sense if norm $\ell_{H,\alpha}^{\Sigma}$ is used; it remains NP-hard if the unit-weight norm ℓ_H^{Σ} is used ($\alpha_j = 1$ for all $j \in \mathcal{N}$) and the upper bounds on the due dates $\overline{\mathbf{d}}$ are restrictive ($\overline{d}_j < \sum_{j=1}^{n} p_j - L^*$);*

(E) *solvable in $O(n \log n)$ time if norm $\ell_{H,\alpha}^{\max}$ is used.*

Proof. We start with the NP-hardness results and then proceed with the polynomially solvable cases.

For the NP-hardness results, consider an instance of the reverse problem R with

$$\overline{d}_j \geq \sum_{j=1}^{n} p_j - L^*, \tag{7}$$

which implies

$$C_j \leq \sum_{j=1}^{n} p_j \leq \overline{d}_j + L^* = d_j''$$

for all the left-shifted schedules $\mathbf{C} = (C_j)_{j=1}^{n}$. Thus we can ignore the deadline constraints in the equivalent problems listed in Theorem 1.

The complexity results for version (A) of the reverse problem follow from the NP-hardness in the strong sense of problem $1||\sum \alpha_j T_j$ [8, 10] and the NP-hardness in the ordinary sense of problem $1||\sum T_j$ [3].

The strong NP-hardness of version (B) of the reverse problem is proved in the Appendix. To the best of our knowledge, prior to this research the complexity status of its forward counterpart $1||\sqrt{\sum \alpha_j T_j^2}$ or equivalently $1||\sum \alpha_j T_j^2$ has been open, see, e.g., [12].

The NP-hardness in the ordinary sense of version (D) of the reverse problem follows from a similar result known for problem $1||\sum \alpha_j U_j$ [6]. In the case of unit costs $\alpha_j = 1$, $j \in N$, the problem is solvable in $O(n \log n)$ time [11] if the deadlines d'' are unrestrictive, which happens, e.g., if they satisfy (7); in the case of small deadlines, problem $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''|\sum U_j$ is NP-hard in the ordinary sense [9].

We now turn to the polynomially solvable cases. Consider version (C) of the reverse problem and its equivalent counterpart $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''|\max\{\alpha_j T_j\}$. The optimal value of the objective in the latter problem is no larger than γ, where

$$\gamma = \max_{j=1,\ldots,n}\{\alpha_j\} \times \sum_{j=1}^{n} p_j, \tag{8}$$

provided that $d_j \geq 0$.

Instead of dealing with deadlines $\mathbf{d}''$, we consider an equivalent problem without deadlines but with precedence constraints between jobs, namely $1|\text{due dates } \mathbf{d}', \text{ prec}|\max\{\alpha_j T_j\}$. For this purpose, in addition to the main jobs $\{1, 2, \ldots, n\}$, we introduce n auxiliary jobs $\{n+1, n+$

$2, \ldots, 2n\}$. For each auxiliary job $n + j$, it is required that the main job j precedes it. The parameters p_{n+j} and d'_{n+j} for the auxiliary jobs are as follows:

$$\begin{aligned} p_{n+j} &= 0, \\ d'_{n+j} &= d''_j, \end{aligned}$$

where d''_j is the deadline parameter of the main job j, $1 \le j \le n$, defined by (2). The α-parameters for the auxiliary jobs are selected as sufficiently large numbers in order to force these jobs to be scheduled before their due dates d'_{n+j}. For example, if

$$\alpha_{n+j} = 2\gamma \text{ for } j = 1, \ldots, n,$$

then the optimal schedule for problem $1|\text{due dates } \mathbf{d}', \text{ prec}| \max\{\alpha_j T_j\}$ has an objective value no larger than γ only if each auxiliary job $n + j$ completes before d'_{n+j}. Due to the precedence constraints, in that schedule the associated main job j is completed before job $n + j$, so it is before its deadline d''_j, as needed.

For problem $1|\text{due dates } \mathbf{d}', \text{ prec}| \max\{\alpha_j T_j\}$ we can apply the $O(m + n \log n)$-time algorithm proposed in [4] for problem $1|\text{prec}| \max\{\alpha_j T_j\}$, where m is the number of precedence constraints. Since in our case $m = n$, we can find a solution to problem $1|\text{due dates } \mathbf{d}', \text{ prec}| \max\{\alpha_j T_j\}$ with auxiliary jobs in $O(n \log n)$ time and use it as a solution for problem $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''|$ $\max\{\alpha_j T_j\}$. Finally, (6) provides the optimal adjusted due dates $\widehat{\mathbf{d}}$ for the reverse problem. Clearly, the time complexity of this approach is $O(n \log n)$.

We treat case (E) in a similar fashion by formulating an $O(n \log n)$-time algorithm for its equivalent counterpart $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''| \max\{\alpha_j U_j\}$. Introduce an equivalent problem, namely $1|\text{due dates } \mathbf{d}', \text{ prec}| \max\{\alpha_j U_j\}$, without deadlines but with precedence constraints between the given jobs $j = 1, \cdots, n$ with due dates d'_j and auxiliary jobs $n + 1, \cdots, 2n$ such that

$$p_{n+j} = 0, \quad d'_{n+j} = d''_j, \quad \alpha_{n+j} = 2\lambda,$$

where $\lambda = \max\{\alpha_j | j = 1, \ldots, n\}$ is the largest value of the objective function. Furthermore, we add the precedence constraints $j \to n + j$ for all $j = 1, \cdots, n$.

The algorithm presented below is an adapted version of the algorithm of Lawler [7] (see, e.g., Section 4.1.1 in [1]). Considering the set S of unscheduled jobs without successors, the algorithm selects a job $j \in S$ with the smallest value $\alpha_j \times \operatorname{sgn} \max\left\{0, p - d'_j\right\}$ and schedules it to finish at time p, where p is the sum of the processing times of all the jobs that have not been scheduled yet. The scheduled job j is eliminated from S, its predecessor (if any) is added to S, p is updated, and the algorithm proceeds in a similar manner.

Algorithm for Problem (E)

1. $p := \sum_{\nu=1}^{n} p_\nu$; $S := \{n + 1, \cdots, 2n\}$; $f_{\max} := 0$;
2. While $S \ne \varnothing$ do
3. Schedule a job $j \in S$ with the smallest value $f_j := \alpha_j \times \operatorname{sgn} \max\left\{0, p - d'_j\right\}$ to finish at time p;
4. $p := p - p_j$;

5. $S := S \setminus \{j\}$;
6. If j has a predecessor $pre(j)$, then $S := S \cup \{pre(j)\}$;
7. $f_{\max} := \max\{f_{\max}, f_j\}$

Endwhile;

8. If $f_{\max} = 2\lambda$, then there exists no feasible schedule

For efficient implementation of the algorithm, we keep

- all the jobs $j \in S$ in a list L in non-increasing order of the d'_j-values, and
- all the jobs $j \in S$ with $d'_j < p$ in a second list M in non-decreasing order of the α_j-values.

To calculate f_j in Step 3, consider the first job j in list L having the largest d'_j-value. If $p \le d'_j$, then $f_j = 0$ and j is eliminated from L. Otherwise, for all the jobs $j \in S$, condition $d'_j < p$ holds and among these jobs the one with the smallest f_j-value can be found as the first job in M. It will be eliminated from L and M.

When p is decreased in Step 4, the relevant jobs have to be eliminated from M. This can be done using list L.

If a job j is added to S, then j is added to L, and in case $d'_j < p$, it is also added to M.

To perform insertion into and deletion from the lists in an efficient way, L and M are organized as doubly linked lists. Furthermore, we add a pointer from each job $j \in S$ to its position in the lists. Thus, each insertion and deletion can be executed in at most $O(\log n)$ time. Since there are at most $O(n)$ insertions and deletions, we have an $O(n \log n)$-time algorithm. ∎

Remark: As shown in the proof of case (D) of Theorem 1, NP-hardness in the ordinary sense of problem R under the norm $\ell^{\Sigma}_{H,\alpha}$ follows from the NP-hardness in the ordinary sense of problem $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''|\sum \alpha_j U_j$. Notice that no pseudo-polynomial time algorithm is known for the latter problem even if $\alpha_j = 1$ for all $j \in \mathcal{N}$, and it is an open question whether problem $1|\text{due dates } \mathbf{d}', \text{ deadlines } \mathbf{d}''|\sum \alpha_j U_j$ is NP-hard in the strong sense. This implies that the same open question remains for the reverse problem under the norm $\ell^{\Sigma}_{H,\alpha}$.

Appendix

The proof of Theorem 1, Case (B). The decision version of problem R is clearly in NP. We perform a reduction from the strongly NP-complete problem 3-PARTITION [5].

3-PARTITION: Given a set of $3t$ positive integers $a_1, a_2, \ldots, a_{3t}$ and an integer B such that $\sum_{i=1}^{3t} a_i = tB$ and $B/4 < a_i < B/2$ for $1 \le i \le 3t$, can the index set $I = \{1, 2, \ldots, 3t\}$ be split into t disjoint 3-element subsets $I_1, I_2, \ldots, I_t$ such that $\sum_{i \in I_j} a_i = B$, $1 \le j \le t$?

Given an instance $(a_1, a_2, \ldots, a_{3t}; B)$ of 3-PARTITION, let $M = \sqrt{\frac{1}{6}t(t+1)(2t+1)B}$ and $L = \frac{1}{2}t(t+1)(2t+1)B^2 = 3BM^2$. An instance of the reverse problem R is characterized by the following parameters:

- job set $\mathcal{N} = \{1, 2, \ldots, 4t\}$ consisting of normal jobs $\{1, 2, \ldots, 3t\}$ and partition jobs $\{3t + 1, 3t + 2, \ldots, 4t\}$;
- for the normal jobs, $\alpha_j = p_j = a_j$, $\underline{d}_j = 0$, $\overline{d}_j = t(L + B), 1 \leq j \leq 3t$;
- for the partition jobs, $\alpha_j = (L + B)M + 1$, $p_{3t+j} = L$, $\underline{d}_{3t+j} = jL + (j - 1)B$, $\overline{d}_{3t+j} = \underline{d}_{3t+j}$, $1 \leq j \leq t$;
- the target value of the maximum lateness is $L^* = 0$;
- the threshold value of the due date adjustment cost is $Y = (L + B)M$.

In the decision version of the reverse problem R, we are required to find out whether there exists a job permutation π and adjusted due dates $\widehat{\mathbf{d}}$ such that $L_{\max}(\pi, \widehat{\mathbf{d}}) \leq L^*$ and $||\widehat{\mathbf{d}} - \mathbf{d}||_{2,\alpha} \leq Y$.

Suppose the constructed instance of 3-PARTITION has a solution $I_1, I_2, \ldots, I_t$. Without loss of generality, we assume that $I_j = \{3j - 2, 3j - 1, 3j\}$, $1 \leq j \leq t$. We show that the permutation

$$\pi = (3t + 1,\ 1, 2, 3,\ 3t + 2,\ 4, 5, 6,\ \ldots,\ 3t + k,\ 3k - 2, 3k - 1, 3k,\ \ldots,\ 4t,\ 3t - 2, 3t - 1, 3t)$$

and the vector $\widehat{\mathbf{d}}$ of adjusted due dates,

$$\begin{aligned} \widehat{d}_j &= C_j(\pi), & 1 \leq j \leq 3t, \\ \widehat{d}_{3t+j} &= d_{3t+j}, & 1 \leq j \leq t, \end{aligned}$$

define a feasible solution to the decision version of the reverse problem R.

Indeed, the due dates of all the jobs satisfy the boundaries $\left[\underline{d}_j, \overline{d}_j\right]$, $1 \leq j \leq 4t$. The total processing time of each triple of normal jobs $3k-2, 3k-1, 3k$ positioned in π between two partition jobs is B so that

$$L_j(\pi, \widehat{\mathbf{d}}) = C_j(\pi) - \widehat{d}_j = 0, \quad 1 \leq j \leq 4t,$$

and the target value of $L_{\max}$ is achieved .

To demonstrate that $||\widehat{\mathbf{d}} - \mathbf{d}||_{2,\alpha} \leq Y$ we use the following conditions

$$\begin{aligned} \widehat{d}_j - d_j &= C_j(\pi) \leq \left\lceil \tfrac{j}{3} \right\rceil (L + B), & 1 \leq j \leq 3t, \\ \widehat{d}_j - d_j &= 0, & 3t + 1 \leq j \leq 4t. \end{aligned}$$

It follows that

$$\begin{aligned} \left(||\widehat{\mathbf{d}} - \mathbf{d}||_{2,\alpha}\right)^2 &= \sum_{j=1}^{3t} \alpha_j C_j^2(\pi) \leq \sum_{k=1}^{t} (a_{3k-2} + a_{3k-1} + a_{3k}) \left[k(L + B)\right]^2 = B(L + B)^2 \sum_{k=1}^{t} k^2 \\ &= B(L + B)^2 \times \frac{1}{6} t(t + 1)(2t + 1) = Y^2. \end{aligned}$$

On the other hand, suppose that $(\pi, \widehat{\mathbf{d}})$ is a solution to the instance of the reverse problem with $L_{\max}(\pi, \widehat{\mathbf{d}}) \leq L^*$ and $||\widehat{\mathbf{d}} - \mathbf{d}||_{2,\alpha} \leq Y$. We denote by $\mathcal{N}_k$ the subset of normal jobs that appear in π after the partition job $3t + k$ and by P_k their total processing time. For completeness, we define $P_{t+1} = 0$. The following sequence of statements proves that 3-PARTITION has a solution.

1. There are no idle times in the schedule given by π.
2. The partition jobs satisfy $C_j(\pi) \leq d_j$, $3t + 1 \leq j \leq 4t$.

3. The partition jobs appear in permutation π in the order of their numbering.
4. The total processing time of the jobs in $\mathcal{N}_k$ satisfies $P_k \geq (t-k+1)B$.
5. The total processing time of the jobs in $\mathcal{N}_k$ satisfies $P_k \leq (t-k+1)B$.
6. Between the two partition jobs $3t+k$ and $3t+k+1$, there are three normal jobs $\mathcal{N}_k \backslash \mathcal{N}_{k+1}$, and their total processing time is B.

Statement 1 is satisfied since the last job completes at time $\sum_{j=1}^{4t} p_j = t(L+B)$ and it cannot exceed its adjusted due date bounded by $\max_{1\leq j\leq 4t}\left\{\overline{d}_j\right\} = t(L+B)$.

Statement 2 holds since $\widehat{d}_j$ cannot exceed $\overline{d}_j$ and $\overline{d}_j = d_j$ for any partition job.

To prove Statement 3, suppose that for $u < v$, a partition job $3t+v$ appears before a partition job $3t+u$. Let $3t+v$ be the first partition job with this property. Then, taking into account that all the partition jobs are of length L, $C_{3t+u}(\pi) \geq (u+1)L$, which exceeds the maximum allowed due date $\overline{d}_{3t+u} = uL + (u-1)B$:

$$C_{3t+u}(\pi) - \overline{d}_{3t+u} \geq (u+1)L - uL - (u-1)B = L - (u-1)B > L - tB > 0.$$

To prove Statement 4, we consider the fragment of the schedule starting with the partition job $3t+k$. Job $3t+k$ is followed by $t-k$ partition jobs of total length $(t-k)\,L$ and by the normal jobs $\mathcal{N}_k$ of total length P_k. Due to Statement 1, the completion time of the last job is $t(L+B)$:

$$C_{3t+k}(\pi) + (t-k)\,L + P_k = t(L+B).$$

Since job $3t+k$ should be completed no later than $\overline{d}_{3t+k} = kL + (k-1)B$, we obtain:

$$P_k = tB + kL - C_{3t+k}(\pi) \geq tB + kL - (kL + (k-1)B) = (t-k+1)B.$$

To prove Statement 5, we use the estimate:

$$\left(||\widehat{\mathbf{d}} - \mathbf{d}||_{2,\alpha}\right)^2 \geq \sum_{j=1}^{3t} \alpha_j C_j^2(\pi) \geq \sum_{k=1}^{t}(P_k - P_{k+1})(kL)^2 = L^2 \sum_{k=1}^{t}(2k-1)P_k.$$

Suppose that $P_z \geq (t-z+1)B + 1$ for some $1 \leq z \leq t$. Since for the remaining values, $P_k \geq (t-k+1)B$ due to Statement 4, we obtain:

$$\left(||\widehat{\mathbf{d}} - \mathbf{d}||_{2,\alpha}\right)^2 \geq L^2 \left(B \sum_{k=1}^{t}(2k-1)(t-k+1) + (2z-1)\right).$$

We calculate the sum on the right hand side:

$$\begin{aligned}
\sum_{k=1}^{t}(2k-1)(t-k+1) &= \sum_{k=1}^{t}(2kt - 2k^2 + 2k - t + k - 1) = \sum_{k=1}^{t}((2t+3)\,k - 2k^2) - t^2 - t \\
&= (2t+3) \times \frac{1}{2}t(t+1) - \frac{2}{6}t\,(t+1)\,(2t+1) - t^2 - t \\
&= \frac{1}{6}t\,(t+1)\,(2t+1) + \frac{1}{6}t\,(t+1) \times 6 - t^2 - t = \frac{1}{6}t(t+1)(2t+1).
\end{aligned}$$

It follows that

$$\begin{aligned}\left(\|\widehat{\mathbf{d}}-\mathbf{d}\|_{2,\alpha}\right)^2 &\geq L^2\left(\frac{1}{6}t(t+1)(2t+1)B+(2z-1)\right)=L^2\left(M^2+(2z-1)\right)\\ &\geq L^2\left(M^2+1\right)=L^2M^2+3BLM^2>(L+B)^2M^2=Y^2,\end{aligned}$$

a contradiction to the assumption that $\|\widehat{\mathbf{d}}-\mathbf{d}\|_{2,\alpha}\leq Y$.

As a consequence of Statements 4 and 5, we conclude that $P_j=(t-j+1)B$, $1\leq j\leq t$. Hence the normal jobs between the partition jobs $3t+j$ and $3t+j+1$ have a total processing time B, $1\leq j\leq t-1$. Since $B/4<p_j=a_j<B/2$ for $1\leq j\leq 3t$, each such set must contain exactly three jobs. Thus the splitting of the normal jobs into triples defines a solution to the instance of 3-PARTITION. ■

Notice that the proof can be easily extended for the case of equal upper bounds for all the due dates, i.e.,

$$\overline{d}_j=t(L+B)=\sum_{j=1}^{4t}p_j,\quad 1\leq j\leq 4t.$$

In spite of the large $\overline{d}_j$, each partition job j, $3t+1\leq j\leq 4t$, is forced to be completed no later than d_j since completing it at time d_j+1 or later incurs a high cost for adjusting $\widehat{d}_j$ and results in $\|\widehat{\mathbf{d}}-\mathbf{d}\|_{2,\alpha}\geq(L+B)M+1>Y$. Thus the equivalent problem $1\|\sum\alpha_jT_j^2$ with unrestrictive deadlines is strongly NP-hard as well.

Acknowledgement

This research was supported in part by NSFC (11171313) and NSFC (11271338).

References

[1] Brucker, P. (2004). *Scheduling Algorithms*, Springer, Berlin.

[2] Brucker, P., & Shakhlevich, N.V. (2009). Inverse scheduling with maximum lateness objective. *Journal of Scheduling, 12*, 475-488.

[3] Du J., & Leung, J.Y.-T. (1990). Minimizing total tardiness on one machine is NP-hard. *Mathematics of Operations Research, 15*, 483-495.

[4] Fields, M.C., & Frederickson, G.N. (1990). A faster algorithm for the maximum weighted tardiness problem. *Information Processing Letters, 36*, 39-44.

[5] Garey, M.R. and Johnson, D.S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA.

[6] Karp, R.M. (1972). Reducibility among combinatorial problems. In R.E. Miller & J.W. Thatcher (Eds) *Complexity of Computer Computations*, Plenum Press, 85-103.

[7] Lawler, E.L. (1973). Optimal sequencing of a single machine subject to precedence constraints. *Management Science, 19*, 544-546.

[8] Lawler, E.L. (1977). A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics, 1*, 331-342.

[9] Lawler, E.L. (1982). Scheduling a single machine to minimize the number of late jobs. Preprint, Computer Science Division, University of California, Berkeley.

[10] Lenstra, J.K., Rinnooy Kan, A.H.G., & Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics, 1*, 343-362.

[11] Moore, J.M. (1968). An n job, one machine sequencing algorithm for minimizing the number of late jobs, *Management Science, 15*, 102-109.

[12] Valente, J.M.S., & Schaller, J.E. (2012). Dispatching heuristics for the single machine weighted quadratic tardiness scheduling problem. *Computers & Operations Research, 39*, 2223-2231.