Discrete Applied Mathematics 189 (2015) 49–52

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Note

Unary NP-hardness of minimizing the total deviation with generalized or assignable due dates

Yuan Gao, Jinjiang Yuan *

School of Mathematics and Statistics, Zhengzhou University, Zhengzhou, Henan 450001, People's Republic of China

ARTICLE INFO

Article history:
Received 26 December 2012
Received in revised form 25 February 2015
Accepted 25 March 2015
Available online 22 April 2015

Keywords:
Single machine scheduling
Generalized due dates
Assignable due dates
Unary NP-hard

ABSTRACT

In this paper, we study the single machine scheduling to minimize the total earliness and tardiness (i.e., the total deviation) with generalized or assignable due dates. Under the assumption of assignable due dates, the due dates are treated as variables and must be assigned to the individual jobs in a schedule. The assumption of generalized due dates is a special version of assignable due dates, in which the due dates are sequenced in the EDD order and assigned to the jobs by the increasing order of their completion times so that the ith completed job receives the ith due date. The exact complexity of the two problems has been reported open in the literature. We show in this paper that the two problems are unary NP-hard.

1. Introduction

Scheduling under the assumption of generalized due dates (GDD) was first introduced by Hall [3]. In this scheduling model, there are n jobs $J_1, \ldots, J_n$ with processing times $p_1, \ldots, p_n$, respectively, and n due dates $d_{[1]}, \ldots, d_{[n]}$ with $d_{[1]} \leq \cdots \leq d_{[n]}$. Denote by $C_j(\pi)$ the completion time of job J_j in a schedule π on a single machine. We order the jobs in the order $J_{\pi(1)}, \ldots, J_{\pi(n)}$ so that $C_{\pi(1)}(\pi) < \cdots < C_{\pi(n)}(\pi)$, that is, $J_{\pi(i)}$ is the ith completed job in π. Then we assign the due date $d_{[i]}$ to job $J_{\pi(i)}$. Thus, a due-dated-oriented criterion of job $J_{\pi(i)}$ under the schedule π can be calculated by $C_{\pi(i)}(\pi)$ and $d_{[i]}$. The complexities of various generalized due date scheduling problems were studied in Hall et al. [5].

Gordon and Kubiak [2] introduced the scheduling under the assumption of assignable release dates and due dates. In their research, there are n pairs (r_j, d_j), $j = 1, 2, \ldots, n$, of release dates and due dates in the restricted form $(r_j, d_j) = ((j-1)P/n, jP/n)$ to be assigned to the n jobs separately, where P is the total processing time of the n jobs. Qi et al. [6] studied the general version of the assignable due dates (ADD) in which the due dates to be assigned have no restriction, i.e., the n due dates to be assigned are independent. In this paper, we accept the problem formulation in Qi et al. [6]. Under the assumption of assignable due dates (ADD), the n previously given due dates $d_1, \ldots, d_n$ are assigned to the n jobs $J_1, \ldots, J_n$, separately. In a schedule π on a single machine, we use $d_{\pi[i]}$ to denote the due date assigned to the ith job $J_{\pi(i)}$. Then a due-dated-oriented criterion of job $J_{\pi(i)}$ under the schedule π can be calculated by $C_{\pi(i)}(\pi)$ and $d_{\pi[i]}$. Note that GDD assumption is a special version of ADD assumption. Then $d_{\pi[i]} = d_{[i]}$, $1 \leq i \leq n$, in every schedule π under the GDD assumption.

As described in Hall [3], the research on scheduling under GDD assumption was motivated by its applications in public utility planning, survey design, and flexible manufacturing. By Qi et al. [6], the ADD assumption provides more flexibility than the GDD assumption in some applications. Furthermore, an application of the scheduling under ADD assumption in the airline industry was also provided in [6].

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

http://dx.doi.org/10.1016/j.dam.2015.03.011
0166-218X/

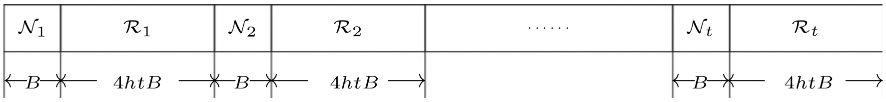

Fig. 1. The schedule π.

For a schedule π of the n jobs $J_1, \ldots, J_n$ on a single machine, we use $E_{\pi(i)}(\pi) = \max\{d_{\pi[i]} - C_{\pi(i)}(\pi),\ 0\}$ and $T_{\pi(i)}(\pi) = \max\{C_{\pi(i)}(\pi) - d_{\pi[i]},\ 0\}$ to denote the earliness and tardiness of job $J_{\pi(i)}$, respectively, under π. Then the total earliness and tardiness of schedule π is given by $\sum_{1\le i\le n}(E_i(\pi) + T_i(\pi)) = \sum_{1\le i\le n}(E_{\pi(i)}(\pi) + T_{\pi(i)}(\pi)) = \sum_{1\le i\le n}|C_{\pi(i)}(\pi) - d_{\pi[i]}|$. By the scheduling notation, we use $1|\text{GDD}|\sum(E_i + T_i)$ to denote the single machine scheduling to minimize the total earliness and tardiness with generalized due dates, and use $1|\text{ADD}|\sum(E_i + T_i)$ to denote the single machine scheduling to minimize the total earliness and tardiness with assignable due dates. For convenience, we use GDD and ADD to denote the above two scheduling problems, respectively.

By Qi et al. [6], there is an optimal schedule for ADD so that the due dates are assigned according to the EDD order. Therefore, an optimal solution for the GDD assignment is also optimal for the ADD assignment. As shown by Hall et al. [4], the problem $1|d_i = d|\sum(E_i + T_i)$ (under the traditional due dates) is binary NP-hard. Since $1|d_i = d|\sum(E_i + T_i)$ is a special version of both GDD and ADD, the two problems GDD and ADD are binary NP-hard. It was reported in Qi et al. [6], the exact complexity (unary NP-hard or pseudo-polynomial-time solvability) is still open.

Computational complexity is an important direction in scheduling research. As both GDD and ADD are binary NP-hard and no pseudo-polynomial-time algorithms were presented until now, the exact complexity of the two problems motivated the research in this paper.

We show in this paper that both GDD and ADD are unary NP-hard.

2. NP-hardness proof

We show the unary NP-hardness of GDD by using the unary NP-complete 3-Partition (Garey and Johnson, [1]) for the reduction.

3-Partition: Given a set of $3t$ positive integers $\{a_1, a_2, \ldots, a_{3t}\}$ and a positive integer B such that $B/4 < a_i < B/2$ for each i with $1 \le i \le 3t$ and $\sum_{i=1}^{3t} a_i = tB$, does there exist a partition of the index set $\{1, \ldots, 3t\}$ into subsets $A_j, j = 1, \ldots, t$, such that $|A_j| = 3$ and $\sum_{i\in A_j} a_i = B$ for each j with $1 \le j \le t$?

Theorem 1. *GDD (and so ADD) is unary NP-hard.*

Proof. Given an instance $(a_1, a_2, \ldots, a_{3t}; B)$ of 3-Partition, we construct an instance of the decision version of GDD as follows.

- We have $n = 3t + 2t^2B = (h + 3)t$ jobs $J_1, \ldots, J_n$, where $h = 2tB$.
- The first $3t$ jobs, called normal jobs, have processing times $p_i = a_i$ for $i = 1, \ldots, 3t$.
- The next ht jobs, called restricted jobs, have processing times $p_i = 4tB$ for $i = 3t + 1, \ldots, n$.
- The n due dates $d_{[i]}$, $1 \le i \le n$, to be assigned, are given by

$$d_{[(j-1)(h+3)+1]} = d_{[(j-1)(h+3)+2]} = d_{[(j-1)(h+3)+3]} = (j-1)(4ht+1)B + B, \quad \text{for } 1 \le j \le t,$$

and

$$d_{[(j-1)(h+3)+3+i]} = (j-1)(4ht+1)B + B + 4tBi, \quad \text{for } 1 \le j \le t,\ 1 \le i \le h.$$

- The threshold value is given by $y = tB$.
- The decision asks whether there is a feasible schedule π such that $\sum_{i=1}^{n}(E_i(\pi) + T_i(\pi)) \le y$.

The above construction can be done in a pseudo-polynomial time, which is polynomial in the unary encode. By the definition of $d_{[i]}$, we have $d_{[1]} \le \cdots \le d_{[n]}$. Denote by $\mathcal{N}$ the set of the $3t$ normal jobs and $\mathcal{R}$ the set of the $n - 3t = ht$ restricted jobs. Then we partition $\mathcal{R}$ into t subsets $\mathcal{R}_1, \ldots, \mathcal{R}_t$ of equal sizes such that $\mathcal{R}_j = \{J_i : 3t + (j-1)h + 1 \le i \le 3t + jh\}$ for each j with $1 \le j \le t$. Note that each $\mathcal{R}_j$ consists of exactly h restricted jobs. We show in the following that the instance of 3-Partition has a solution if and only if the GDD instance has a feasible schedule π such that $\sum_{i=1}^{n}(E_i(\pi) + T_i(\pi)) \le y$.

Suppose first that the instance of 3-Partition has a solution. Then there is a partition $(A_1, \ldots, A_t)$ of $\{1, \ldots, 3t\}$ such that $|A_j| = 3$ and $\sum_{i\in A_j} a_i = B$ for each j with $1 \le j \le t$. By setting $\mathcal{N}_j = \{J_i : i \in A_j\}$ for $j = 1, \ldots, t$, we obtain a partition $\mathcal{N}_1, \ldots, \mathcal{N}_t$ of the normal jobs so that each $\mathcal{N}_j$ has exactly three jobs with total processing time B. Let π be the feasible schedule in which the n jobs are processed consecutively from time 0 in the order $\mathcal{N}_1 \prec \mathcal{R}_1 \prec \mathcal{N}_2 \prec \mathcal{R}_2 \prec \cdots \prec \mathcal{N}_t \prec \mathcal{R}_t$ so that the three jobs in each $\mathcal{N}_j$ are processed in the LPT order. Fig. 1 displays the structure of schedule π.

For each j with $1 \le j \le t$, the three jobs in $\mathcal{N}_j$, which receive the common due date $(j-1)(4ht+1)B + B$, are processed in the time interval $[(j-1)(4ht+1)B,\ (j-1)(4ht+1)B + B]$ in π. Since the three jobs in $\mathcal{N}_j$ are processed in the LPT order, the first job in $\mathcal{N}_j$ has earliness at most $2B/3$, the second one has earliness at most $B/3$ and the third one has earliness 0.