SHORT COMMUNICATION

A note on proving the strong NP-hardness of a scheduling problem with position dependent job processing times

Radosław Rudek

Abstract   In this paper, we show that the strong NP-hardness proof of the single machine makespan minimization problem with ready times and job processing times described by a non-increasing power function dependent on a job position in a sequence presented in Bachman and Janiak (J Oper Res Soc 55:257–264, 2004) is incorrect. Namely, the applied transformation from 3- PARTITION problem to the considered scheduling problem is polynomial not pseudopolynomial. Thus, the related problem is NP-hard, but it is not proved to be strongly NP-hard.

Keywords   Computational analysis · Strong NP-hardness · Scheduling · Learning effect · Position-dependent processing time

1 Introduction

Bachman and Janiak [1] claimed that the single machine makespan minimization problem with ready times is strongly NP-hard if job processing times are described by a non-increasing power function dependent on a job position in a sequence. To prove it, the authors constructed a reduction from 3-PARTITION to the decision version of this problem. However, we show that the reduction is only polynomial not pseudopolynomial. Therefore, the authors did not prove the strong NP-hardness, but NP-hardness only. Thus, the complete computational status of this problem is still an open issue.

Furthermore, the results presented in this paper can be a useful hint: how to avoid serious mistakes during proving the strong NP-hardness.

Throughout the paper, we will keep the notation and terminology used by Bachman and Janiak [1].

R. Rudek (✉)
Wrocław University of Economics, Komandorska 118/120, 53-345 Wrocław, Poland
e-mail: radoslaw.rudek@ue.wroc.pl

2 Problem formulation

There are given a single machine and a set $J = \{1, \ldots, n\}$ of n independent and non-preemptive jobs available for processing at their ready times $r_i, i = 1, \ldots, n$. The processing time $p_i(v)$ of job i is a function dependent on its position v in a sequence. The considered model of the job processing time is characterized by position-dependent function:

$$p_i(v) = a_i v^{-b}, \tag{1}$$

where a_i and b denote a normal processing time and a learning ratio, respectively. It is assumed that a_i are positive integers and b is a non-negative rational.

The completion time $C_{\pi(i)}$ of a job placed in the ith position in the schedule π is given by the following recursive expression:

$$C_{\pi(i)} = \max\{C_{\pi(i-1)}, r_{\pi(i)}\} + p_{\pi(i)}(i), \tag{2}$$

where $C_{\pi(0)} = 0$ and $i = 1, \ldots, n$. The problem is to find a schedule π for which the criterion value $C_{\max}$ (the makespan) is minimized, i.e., $C_{\max} \triangleq C_{\pi(n)}$. The problem according to the three field notation scheme is denoted by $1|p_i(v) = a_i v^{-b}|C_{\max}$.

3 Computational complexity

In this section, we will show that the strong NP-hardness proof in [1] is incorrect. At first let us recall the proof and the definition of 3- PARTITION.

3- PARTITION [3]: Given $3m + 1$ positive integers $x_1, \ldots, x_{3m}$ and B satisfying $\sum_{j=1}^{3m} x_j = mB$ and $B/4 < x_j < B/2$ for $j = 1, \ldots, 3m$, is there a partition of the set $\{1, \ldots, 3m\}$ into m disjoint subsets $X_1, \ldots, X_m$ such that $\sum_{j \in X_i} x_j = B$ for $i = 1, \ldots, m$?

The decision version of the considered scheduling problem is given as follows: *Does there exist such a schedule π of jobs on the machine for which the criterion value $C_{\max}$ is equal or lower than the given value y?*

Based on an instance of 3- PARTITION, the authors constructed an instance of $n = m(m^4 B + 3)$ jobs with a common learning ratio $b = 1$. Among them there are $3m$ partition jobs with the following parameters:

$$a_i = x_i; \quad r_i = 0; \quad i = 1 \ldots, 3m.$$

There are $m^5 B$ enforcer jobs, which are characterized by a common normal processing time $a_{E_{(j-1)m^4 B+i}} = a_E = 1$, where $j = 1, \ldots, m$ and $i = 1, \ldots, m^4 B$. However, the enforcer jobs have different ready times, which partition them into m groups with the following parameters:

$$r_{E_{(j-1)m^4 B+i}} = r_{E_j} = (j - 1)m^4 B + \frac{B}{m^4 B + 3} \sum_{k=1}^{j-1} \frac{1}{k}, \tag{3}$$

where $j = 1, \ldots, m$, and $i = 1, \ldots, m^4 B$. Finally the parameter y, that was missing in [1], is defined as follows $y = m^5 B + \frac{B}{m^4 B + 1} \sum_{k=1}^{m} \frac{1}{k}$.

At first recall that the computational complexity theory requires "reasonable and concise" encoding scheme for the problem to represent values of its parameters on Deterministic Turing Machine (see [3]). Therefore, the values of parameters of the considered problem have to be expressed as integer numbers. If it is not the case, then the values of parameters have to be transformed (by multiplying) to the integer numbers. Note that only r_{E_j} in (3) are fraction numbers by the term $\frac{1}{m^4 B + 3} \sum_{k=1}^{j-1} \frac{1}{k}$ and y by $\frac{1}{m^4 B + 1} \sum_{k=1}^{m} \frac{1}{k}$. Therefore, to use the computational complexity theory, we have to express each r_{E_j} and y as the integer numbers. Without loss of generality the value of y and each parameter a_l and r_l for $l = 1, \ldots, m(m^4 B + 3)$ have to be multiplied by the number $A' = (m^4 B + 3)(m^4 B + 1) \cdot A$ such that $r'_{E_j} = A' \cdot r_{E_j}$ and $y' = A' \cdot y$ are integer numbers for $j = 1, \ldots, m$. On this basis, the constructed instance of the decision version of the considered scheduling problem has the following integer parameters:

$$a'_i = A' \cdot x_i; \quad r'_i = 0; \quad i = 1 \ldots, 3m,$$

$b' = 1, a'_{E_{(j-1)m^4 B + i}} = a'_E = A'$, where $j = 1, \ldots, m$ and $i = 1, \ldots, m^4 B$ and

$$r'_{E_{(j-1)m^4 B + i}} = r'_{E_j} = A' \cdot \left[(j-1)m^4 B + \frac{B}{m^4 B + 3} \sum_{k=1}^{j-1} \frac{1}{k} \right]$$

$$= (j-1)m^4 B \cdot (m^4 B + 3)(m^4 B + 1)A + B \cdot (m^4 B + 1)A \cdot \sum_{k=1}^{j-1} \frac{1}{k}$$

and

$$y' = A' \cdot \left[m^5 B + \frac{B}{m^4 B + 1} \sum_{k=1}^{m} \frac{1}{k} \right]$$

$$= m^5 B \cdot (m^4 B + 3)(m^4 B + 1)A + B \cdot (m^4 B + 3)A \cdot \sum_{k=1}^{j-1} \frac{1}{k}.$$

Since the values of the parameters have to be integers, therefore, A is such number that all terms $A \cdot \sum_{k=1}^{j} \frac{1}{k}$ for $j = 1, \ldots, m$ are integers. Note that the values of all terms $A \cdot \sum_{k=1}^{j} \frac{1}{k}$ (where $j = 1, \ldots, m$) are the least integers if A is the least common multiple (lcm) of the following numbers $1, 2, \ldots, m$, i.e., $A = \mathrm{lcm}(1, 2, \ldots, m)$. Following [2,4], we know that $\log \mathrm{lcm}(1, 2, \ldots, m) \backsim m$, thus, A can be bounded below by 2^m as m tends to infinity.

Recall now a condition from a definition of a pseudopolynomial reduction [3,5,6]. Let π_1 and π_2 are two decision problems. Let D_{π_1} and D_{π_2} denote their sets of all possible instances, $Max(I)$ denotes the maximum value of the numbers for an instance

I and $N(I)$ is the size of I. Let $f : D_{\pi_2} \longrightarrow D_{\pi_1}$ denote the reduction from π_2 to π_1. One of the requirements for f to be pseudopolynomial is such that there must exist a polynomial Q of two variables that holds:

$$\forall I \in D_{\pi_2} : Max(f(I)) \leq Q(Max(I), N(I)). \qquad (4)$$

It means that the values of any instance I of the problem π_2 cannot increase in an exponential manner if π_2 is reduced to π_1.

Let π_2 denote 3- PARTITION and π_1 is the considered scheduling problem. It is obvious that for the given reduction and $I \in D_{\pi_2}$, we have $Max(I) = B$, $N(I) = 3m$, $Max(f(I)) > A \geq 2^m$ and $N(f(I)) = m(m^4 B + 3)$. Note that the values of the problem π_1 increase in an exponential manner depending on the values of π_2, thus, there does not exist such Q for which (4) holds, thereby the reduction cannot be pseudopolynomial. Therefore, the proof of the strong NP-hardness is incorrect.

Nevertheless, the constructed reduction is polynomial one, thus, the problem $1|p_i(v) = a_i v^{-b}|C_{max}$ is at least NP-hard, but the strong NP-hardness is not determined.

4 Conclusion

In this note, we showed that the strong NP-hardness proof of the single machine makespan minimization problem with ready times and job processing times described by a non-increasing power function dependent on a job position in a sequence provided by Bachman and Janiak [1] is incorrect. Thus, the problem is at least NP-hard. However, the complete computational complexity of this problem is still an open issue: strongly or ordinary NP-hard?

References

1. Bachman, A., Janiak, A.: Scheduling jobs with position dependent processing times. J. Oper. Res. Soc. **55**, 257–264 (2004)
2. Farhi, B., Kane, D.: New results on the least common multiple of consecutive integers. P. Am. Math. Soc. **137**, 1933–1939 (2009)
3. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
4. Hardy, G.H., Wright, E.M.: An Introduction to the Theory of Numbers. Oxford University Press, New York (1979)
5. Lee, C.-Y., Vairaktarkis, G. : Complexity of single machine hierarchical scheduling: a survey. In: Pardalos, P.M. (ed.) Complexity in Numerical Optimization, pp. 269–298. World Scientific, Singapore (1993)
6. Pardalos, P.M., Resende, M. : Combinatorial optimization. In: Pardalos, P.M., Resende, M. (eds.) Handbook of Applied Optimization, pp. 51–52. Oxford University Press, Oxford (2002)