Weighted Flow-Shop Problem with a Common Due Date and Late Work Criterion

Jacek Błażewicz
Institute of Computing Science, Poznań University of Technology, Poland.
blazewic@sol.put.poznan.pl

Erwin Pesch
Institute of Economics and Business Administration, BWL 3, University of Bonn, Germany.
E.Pesch@uni-bonn.de

Małgorzata Sterna [1]
Institute of Computing Science, Poznań University of Technology, Poland.
Malgorzata.Sterna@cs.put.poznan.pl

Frank Werner
Faculty of Mathematics, Otto-von-Guericke-University, Germany.
Frank.Werner@mathematik.uni-magdeburg.de

Abstract

The paper concerns two-machine flow-shop scheduling problem with a common due date and the late work criterion. It presents an NP-hardness proof based on a transformation from the partition problem together with a pseudo-polynomial time dynamic programming approach. The results made it possible to classify the problem as binary NP-hard.

1. Introduction

The late work criterion can be applied for all scheduling cases where the size of late parts of particular activities should be minimised without taking into account the quantity of the delay. This performance measure finds its applications in control processes [1, 3, 12], agriculture processes [10, 11, 12] or manufacturing planning [12].

The late work objective function was first proposed for the parallel machines scheduling problem [1, 3] and then applied for one-machine environment [10, 11]. The criteria minimising the amount of late work have been recently applied for shop scheduling, particularly to two-machine cases with a common due date [4, 5, 12].

Introducing the following notation [2]:

J - the set of jobs, $| J | = n$,
J_i - the i-th job,
d_i - the due date for a job J_i (in a common due date case, $d_i = d$ for all J_i),
w_i - the weight of a job J_i,
m - the number of machines M_j (in the case considered, $m = 2$),
T_{ij} - the task representing processing a job J_i on a machine M_j,
p_{ij} - the processing time of a task T_{ij}
C_{ij} - the completion time of a task T_{ij},

the late work value for a particular job J_i is defined in the non-preemptive case as:

$$Y_i = \sum_{T_{ij} \in J_i} \min\{\max\{0, C_{ij} - d_i\}, p_{ij}\}$$

If preemptions are allowed, then the late work for a job J_i, which particular task T_{ij} is executed in k_{ij} parts, where the k-th part starts at the time S_{ij}^{k} and finishes at the time C_{ij}^{k} is defined as:

$$Y_i = \sum_{T_{ij} \in J_i} \sum_{k=1}^{k_{ij}} \max\left\{C_{ij}^{k} - \max\left\{d_i, S_{ij}^{k}\right\} 0\right\}$$

Based on the parameter defined above, the late work criteria are defined as:

$$Y = \sum_{J_i \in J_i} Y_i, \quad Y_w = \sum_{J_i \in J_i} w_i Y_i$$

2. Problem F2 | d_i =d | Y_w

The two-machine flow-shop problem with a common due date and the weighted late work criterion is binary NP-hard. Its complexity status has been determined based on a transformation of the partition problem [7] to the decision counterpart of this scheduling case and on a formulation of a pseudo-polynomial time dynamic programming approach solving it.

2.1. NP-hardness proof

The NP-hardness of the problem F2 | $d_i = d$ | Y_w is shown by transforming to it the partition problem defined as follows.

Does there exists a subset $A' \subseteq A$, where A is a set of elements $a_i \in A$ described by a positive integer size $s(a_i)$, such that $\sum_{a_i \in A'} s(a_i) = \sum_{a_i \in A \backslash A'} s(a_i)$?

For a given instance of the partition problem, we construct an instance of the scheduling problem with a due date d equal to 2B, where $B = \sum_{a_i \in A} s(a_i)$ and n = | A | +1 jobs. Jobs J_1, ... J_{n-1} have the unary weights $w_i = 1$ and consist of tasks with the processing times $p_{i1} = 0$ and $p_{i2} = s(a_i)$. The remaining job J_n is executed on both machines for B time units (i.e. $p_{n1} = B$, $p_{n2} = B$) and have a big weight $w_n = B^2$.

We proved that the partition problem has a solution if and only if there exists a schedule for the problem F2 | $d_i = d$ | Y_w with the criterion value not exceeding B.

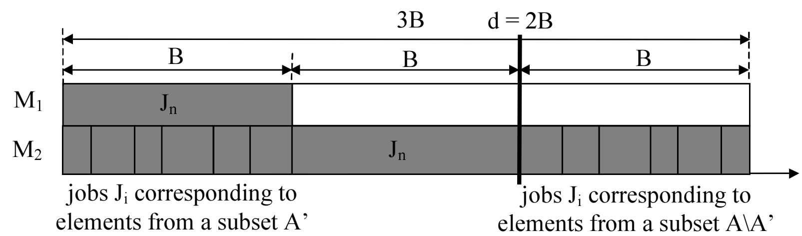

Figure 1. A schedule corresponding to a solution of the partition problem

If the partition problem has a solution then we construct the schedule, as it is shown in Figure 1, which is optimal from the late work criterion point of view. If we have an optimal schedule of the scheduling problem, i.e. with the criterion value $Y_w \leq B$, then the job J_n has to be processed early because each late unit of this job will increase the criterion value of $w_n = B^2$. Moreover, at most B time units of remaining jobs can be processed after J_n on M_2. The division of the job set into a subset executed before and after J_n on M_2 (cf. Figure 1) determines a solution of the partition problem.

With regard to the fact that the flow shop scheduling problem is a special case of the job shop scheduling one, where all jobs have exactly the same sequence of their execution on particular machines, we can state that the problem $J2 \mid d_i = d \mid Y_w$ is also NP-hard.

2.2. Dynamic programming approach

The problem $F2 \mid d_i = d \mid Y_w$ can be solved in pseudo-polynomial time by a dynamic programming method inspired by an approach to the similar problem of minimising the weighted number of late jobs [9].

Taking into account that in any optimal solution the early jobs have to be scheduled in Johnson's order [8, 12] and minimising the weighted late work is equivalent to maximising the weighted early work [12], one has to select the first late job and the set of early jobs to solve the case. Thus, we consider each job as the first tardy job J_x and determine an optimal solution by calculating a recurrence function $f_k(A, B, t, a)$. It denotes the maximum amount of the weighted early work of jobs $J_k, J_{k+1}, ..., J_x$ assuming that the first job from this set starts on the machine M_1 at the time A and not earlier than at the time B on the machine M_2. Additionally, t time units are reserved for performing jobs among $J_1, ..., J_{k-1}$ on M_1 after J_x before the common due date d. Finally, we distinguish with the value $a \in \{0,1\}$ the situations where no job or exactly one job among $J_1, ..., J_{k-1}$ is only partially executed on M_1 within these t time units.

To solve the problem one has to calculate $f_k(A, B, t, a)$ for all jobs J_k ($k = n, ..., 1$) and for all possible values of A, B, t ($0 \leq A, B, t \leq d$) and $a \in \{0,1\}$. The optimal value of the weighted early work is determined by the value $f_1(0,0,0,0)$

The calculations of the recurrence relation presented above and the construction of an optimal solution requires $O(n^2d^4)$ time. The existence of a pseudo-polynomial time method allowed one to classify the problem $F2 \mid d_i = d \mid Y_w$ as NP-hard in the ordinary sense.

3. Conclusions

In the paper the problem $F2 \mid d_i = d \mid Y_w$ was studied. We showed the binary NP-hardness of this scheduling case by providing the transformation from the partition problem and formulating the pseudo-polynomial time dynamic programming approach.

The results obtained for the late work criteria in the shop environment constitute the theoretical part of the studies on a flexible manufacturing system located at the Poznań University of Technology [13] and support the analysis of a production planning problem in this system.

Endnotes

1. The author has been awarded the Annual Stipend for Young Scientists of the Foundation for Polish Science.

References

[1] Błażewicz, J. (1984). Scheduling Preemptible Tasks on Parellel Processors with Information Loss, *Recherche Technique et Science Informatiques,*3 (6), 415-420.

[2] Błażewicz, J., Ecker, K., Pesch, E., Schmidt, G. and Węglarz, J. (2001). *Scheduling Computer and Manufacturing Processes*, Springer, Heidelberg.

[3] Błażewicz, J. and Finke, G. (1987). Minimizing Mean Weighted Execution Time Loss on Identical and Uniform Processors, *Information Processing Letters* 24, 259-263.

[4] Błażewicz, J., Pesch, E., Sterna, M. and Werner, F. (1999). Total Late Work Criteria for Shop Scheduling Problems, in: *Operations Research Proceedings 1999*, Springer, Heidelberg.

[5] Błażewicz, J., Pesch, E., Sterna, M. and Werner, F. (2000). Shop Scheduling Problems with Late Work Criteria, *Otto-von-Guericke-Universität Magdeburg Preprint* 35.

[6] Brucker, P. (1995). *Scheduling Algorithms*. Springer, Heidelberg.

[7] Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractability*, W.H. Freeman and Co., San Francisco.

[8] Johnson, S.M. (1954). Optimal Two- and Three-stage Production Schedules, *Naval Research Logistics Quarterly* 1, 61-68.

[9] Józefowska, J., Jurisch, B. and Kubiak, W. (1994). Scheduling Shops to Minimize the Weighted Number of Late Jobs, *Operation Research Letters* 16 (5), 277-283.

[10] Potts, C.N. and Van Wassenhove, L.N. (1991). Single Machine Scheduling to Minimize Total Late Work, *Operations Research* 40 (3), 586-595.

[11] Potts, C.N. and Van Wassenhove, L.N. (1992). Approximation Algorithms for Scheduling a Single Machine to Minimize Total Late Work, *Operations Research Letters* 11, 261-266.

[12] Sterna M. (2000). *Problems and Algorithms in Non-Classical Shop Scheduling*, Scientific Publishers of the Polish Academy of Sciences, Poznań.