The Binary NP-hardness of the Two-Machine Job Shop Problem with the Weighted Late Work Criterion

Jacek Blazewicz[1], Erwin Pesch[2], Malgorzata Sterna[1], Frank Werner[3]

[1] Institute of Computing Science, Poznan University of Technology
Address: Piotrowo 3A, 61-495 Poznan, Poland
Phone: +48 61 790790, fax : +48 61 8771525
Email : blazewic@sol.put.poznan.pl
Email : Malgorzata.Sterna@cs.put.poznan.pl
[2] Institute of Information Systems, FB 5 - Faculty of Economics, University of Siegen
Address : Hoelderlinstr. 3, 57068 Siegen, Germany
Phone: +49 271 740 2420
Email : pesch@fb5.uni-siegen.de
[3] Faculty of Mathematics, Otto-von-Guericke-University
Address: PSF 4120, 39016, Magdeburg, Germany
Phone: +49 391 6712025, fax : +49 391 6711171
Email: Frank.Werner@mathematik.uni-magdeburg.de

Abstract

The paper presents a dynamic programming approach for the two-machine non-preemptive job-shop scheduling problem with a total weighted late work criterion and a common due date, $J2 \mid d_i=d \mid Y_w$, which is known to be *NP*-hard. The late work performance measure estimates the quality of the obtained solution with regard to the duration of late parts of tasks not taking into account the quantity of this delay. Providing a pseudo-polynomial time method for the problem mentioned, we can classify it as binary NP-hard.

Keywords: scheduling, job-shop, late work criteria, dynamic programming

1 Introduction

Revenue management is essentially the process of allocating resources to the right customer at the right time and the right price. The focus is on maximizing profit or revenue and it has led to an increased profit in a variety of industries. Still, some of the most important applications are in the airline business, i.e. where the basis question arises whether or not to accept or reject a booking request (for a flight) within a specific booking class at a given fare [cf. 12, 17]. In this paper we are faced the situation where a set of customer demands of expected size contribute to the company's revenue in an expected amount that is a function of the demand size and importance of the particular customer. The company is supposed to answer the question how the limited resource capacity over a certain time horizon should be allocated in order to produce the customer's demand within its given due date in a way that maximizes the company's revenue. This leads to the question which customer orders are to be accepted and which orders should be rejected, i.e. the expected revenue of the latter will be lost because their production cannot be finished before the due date. Thus, rejected orders are late work which could only be produced after the due date.

Due date involving criteria are performance measures often used in practical applications [cf. 2, 7, 13]. Generally, they represent the customer point of view allowing to minimize the delay of orders realized in a system. Classical objective functions of this type, such as maximum lateness or total tardiness [cf. 2, 7, 13] are calculated with regard to the quantity of the delay, while the late work criterion allows for minimizing the amount of work executed after given due dates.

The late work objective function has not been widely investigated, although it finds many practical applications, e.g. in data collecting in control systems [1, 3], supporting agriculture technologies [4, 5, 16] or designing production plans within predefined time periods in manufacturing systems [16] or recently in revenue management [12, 17].

The late work criteria was proposed in the context of parallel machines [1, 3] and then applied to the one-machine scheduling problem [14, 15]. More recently, some general complexity results were obtained [4, 5, 16], which allow one to consider problems with the late work criterion as more complicated than analogous problems with the maximum lateness objective function. Then, the late work performance measure has been investigated in the dedicated machine environment [5, 6, 16].

2 Problem definition

In the paper, we consider a non-preemptive scheduling problem with the total weighted late work criterion and a common due date in the two-machine job-shop environment. The problem basically arises for a medium sized manufacturer (approximately 1500 employees) producing parts (e.g. power brake units, booster, etc.) on demand for most of the existing automobile companies in Europe, America, and Asia. A set of jobs (customer orders, e.g. 30000 booster of a particular type) consists of two tasks that have to be processed on two machines in a predefined order (for each job). The tasks' processing times reflect the expected order size for the two different machines. A machine can process only one job at a time and a job cannot be handled on both machines simultaneously. Within our earlier research, we have shown that analogous problems in open-shop [5] and flow-shop systems [6] are binary *NP*-hard. With regard to the hardness of the flow-shop problem, the job-shop one (being its generalization [cf. 2, 7, 13]) is also computationally hard [8]. Here, we propose a pseudopolynomial time dynamic programming method solving the problem considered (the approach was inspired by methods designed for cases with the weighted number of late jobs as an objective function [11]).That allows us to classify this case as binary *NP*-hard and to finish the research on two-machine weighted shop scheduling with a common due date.

More formally, in the job shop scheduling problem [cf. 2, 7, 13], $J2 \mid n_i \le 2, d_i = d \mid Y_w$, we have to schedule a set of jobs $J = \{J_1, \ldots, J_i, \ldots, J_n\}$ on two dedicated machines M_1, M_2. Each job $J_i \in J$ consists of at most two tasks T_{i1} and T_{i2}, (i.e. $n_i \le 2$) described by the processing times p_{i1}, p_{i2} and machine requirements. Particular jobs have to be performed, without preemptions, on machines M_1, M_2 in the predefined order. Each job can be processed on at most one machine at the same time and each machine can perform at most one task at the same time. We have to minimize the total weighted late work in the system. The late work Y_i for job $J_i \in J$ is determined as the sum of late parts of tasks T_{i1} and T_{i2}, executed after a common due date d, on machines M_1 and M_2, respectively. Denoting as C_{i1}, C_{i2} their completion times, the late work for job J_i is given by:

$$Y_i = \sum_{j=1,2} \min\{ \max\{0, C_{ij} - d\}, p_{ij}\}.$$

To determine the total weighted late work we consider the expected revenue losses in the system, i.e. we sum up late work for all jobs (where $n=|J|$) taking into account their given weights (customer importance) w_i, i.e.:

$$Y_w = \sum_{i=1}^{n} w_i Y_i.$$

3 Dynamic programming approach

Let the set of jobs J be partitioned into two subsets J^1 and J^2 containing all jobs with the first (or only) task processed on machine M_1 and M_2, respectively. We can assume that early jobs are processed in Jackson's order [9]. Jackson's rule states that jobs from J^1 proceed J^2 on M_1, while on M_2 jobs from J^2 are executed before J^1 (for both sets jobs containing only one task are performed as the last ones). Sets J^1, J^2 are scheduled according to Johnsons's rule [10], so within sets J^1 and J^2 all jobs J_i with $p_{i1} \le p_{i2}$ are sequenced in non-decreasing order of p_{i1}, while the rest, with $p_{i1} > p_{i2}$, is scheduled in non-increasing order of p_{i2}. Jackson's order is optimal from the schedule length point of view. Thus, scheduling early jobs in this sequence, we obtain the shortest subschedule of early jobs, the maximum machine utilization and, in consequence, the maximum amount of the weighted early work for those jobs. Similarly as for the flow shop problem [6], we use the fact that maximizing the total weighted early work is equivalent to minimizing the total weighted late work, which is the criterion under consideration.

Based on the above observation, that all early jobs have to be scheduled in Jackson's order, for any subset of early jobs $J' \subseteq J^1 \cup J^2$ in an optimal solution, we can assume that jobs from $J^1 \cap J'$ precede jobs from $J^2 \cap J'$ on M_1, and oppositely jobs from $J^2 \cap J'$ precede jobs from $J^1 \cap J'$ on M_2. Moreover, we can assume that the first job of both sets $J^1 \cap J'$ and $J^2 \cap J'$ starts at time zero on machines M_1 and M_2, respectively.

Because of *NP*-hardness of the considered problem, we have to check all possible schedules to determine an optimal one. The search in the solution space is performed in a systematic way according to a dynamic programming approach. Taking into account all properties of the problem, we can restrict significantly the set of solutions explicitly analyzed. Actually, there are only three possible schemes of an optimal solution, which have to be compared. They reflect the different situations when there are (or there are no) tasks executed partially late on machines M_1 or M_2. Denoting with J^P a set of jobs with partially late tasks, we have to consider:

- $J^P = \{J_a, J_b\}$, i.e. there are on both machines partially late tasks belonging to two different jobs, where J_a denotes a job partially late on M_1 and J_b is a job partially late on M_2,
- $J^P = \{ J_x \}$, i.e. there is one partially late task, either on M_1 or on M_2, belonging to job J_x,
- $J^P = \emptyset$, i.e. there is no partially late task in a system.

For a particular set J^P, we renumber the remaining jobs $J \setminus J^P$ in Jackson's order obtaining the sequence $\hat{J} = (\hat{J}_1, \ldots, \hat{J}_u, \hat{J}_{u+1}, \ldots, \hat{J}_{\tilde{n}})$, where $\hat{J}_1, \ldots, \hat{J}_u \in J^1 \setminus J^P$, $\hat{J}_{u+1}, \ldots, \hat{J}_{\tilde{n}} \in J^2 \setminus J^P$, u denotes the number of jobs with the first (only) task on M_1 and $\tilde{n}$ denotes the number of jobs to be scheduled (besides J^P). To find an optimal order of jobs subject to set J^P, we have to choose an optimal variant of scheduling particular jobs $\hat{J}_k \in J \setminus J^P$ (i.e. $\hat{J}_k \in \hat{J}$). Job $\hat{J}_k$ may be executed early, totally late or early on its first machine and totally late on the second one. No task of job $\hat{J}_k \in J \setminus J^P$ can be performed partially late, because, in this case, $\hat{J}_k$ would have to be a member of J^P.

To find an optimal solution of the problem, we have to analyze all possible sets of jobs with partially late tasks J^P. For a particular set J^P, we calculate initial conditions ($f_{\tilde{n}+1}$) determining the amount of weighted early work corresponding to this set. Then, we consider the remaining jobs $\hat{J}_k \in \hat{J}$ calculating for them recurrence relations (f_k) denoting the amount of weighted early work obtained for set $\{J_k, \ldots, J_{\tilde{n}}\} \cup J^P$. First, we analyze all jobs with the first (only) task executed on machine M_2, i.e. $k = \tilde{n}, \ldots, u+1$, and then, using slightly different recurrence relations, all jobs with the first (only) task executed on machine M_1, i.e. $k = u, \ldots, 1$. The value obtained for the first job $\hat{J}_1$ (f_1) denotes the weighted early work for all jobs $\{J_1, \ldots, J_{\tilde{n}}\} \cup J^P$ subject to set J^P.

After analyzing all possible sets J^P, we determine the optimal weighted early work for the problem under consideration. Then, restoring decisions taken during dynamic programming calculations for an optimal set J^P, we schedule optimally particular tasks from $J \setminus J^P$. All early jobs have to be executed before a common due date in Jackson's order, while remaining jobs are performed between those early ones and J^P in an arbitrary order.

Initial conditions

To find an optimal solution of the problem, we have to analyze all possible sets of jobs with partially late tasks – J^P. The weighted early work corresponding to this set is determined by initial conditions defined as $f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2)$, where $\tilde{n} = |J \setminus J^P|$. Function $f_{\tilde{n}+1}$ denotes the maximum amount of the weighted early work provided that the totally early tasks of jobs from J^P (if any) start exactly at time A on M_1, and exactly at time B on M_2. Moreover, there are exactly t_1, t_2 units of early tasks and exactly L_1, L_2 units of partially late tasks on machines M_1 and M_2, respectively. In general, as we have mentioned there are 3 possible cases, when set J^P contains two jobs, one or no job.

To illustrate the meaning of function parameters, let us consider, for example, $|J^P|=2$ and a job J_a partially late on M_1, belonging to J^2 (i.e. executed first on M_2 then on M_1 with regard to the predefined job precedence constraints). For such a job, we analyze among others the situation, when the first early task of J_a starts at time B on M_2 and it is executed for $t_2 = p_{a2}$ units. The second operation of J_a is partially late on M_1 and only t_1 units from p_{a1} units of this task are executed early, before the common due date d.

In general, if set J^P contains two jobs, i.e. $J^P = \{J_a, J_b\}$, then we have to analyze four subcases corresponding to the different types of jobs (they may belong to J^1 or J^2). For a set J^P containing only one job, i.e. $J^P = \{J_x\}$, only two subcases are possible depending on the type of job J_x. Finally, we have to analyze the case when no partially late task exists in the system. Such a situation occurs, when on a particular machine a task finishes/starts exactly at time d or there is idle time around a common due date.

Taking into account the fact that all parameters of function $f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2)$ are bounded by $O(d)$ the calculation of the initial conditions for any set J^P takes $O(d^6)$ time.

Recurrence relations

After determining initial conditions for a particular set J^P, we calculate the recurrence relations for the remaining jobs $\hat{J}_k \in J \setminus J^P$, numbered according to Jackson's rule as $\hat{J}_1, \ldots, \hat{J}_u, \hat{J}_{u+1}, \ldots, \hat{J}_{\tilde{n}}$. As we have mentioned, first, we analyze jobs with the first (only) task on M_2 ($k = \tilde{n}, \ldots, u+1$). Then, jobs with the first (only) task on M_1 are taken into account ($k = u, \ldots, 1$). For job $\hat{J}_k$, we determine the amount of the weighted early work for jobs $\{\hat{J}_k, \ldots, \hat{J}_{\tilde{n}}\} \cup J^P$ based on the recurrence relation $f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$. The meaning of the parameters changes slightly

depending on the job type, whether $\hat{J}_k \in J^2 \setminus J^P$ or $\hat{J}_k \in J^1 \setminus J^P$. In general, calculating the recurrence function value for a job $\hat{J}_k$, we assume that this job can be added to a partial schedule already obtained (i.e. it cannot start earlier than at time A on M_1 and at time B on M_2). Moreover, we reserve some time (r_1 or r_2) for tasks of early jobs of another job type, which have to be executed after a task of job $\hat{J}_k$ scheduled early. If $\hat{J}_k$ is scheduled partially late, then its early task has to be executed within a certain time window (i.e. t_1 or t_2). Additionally, some time interval (i.e. T_1 or T_2) is reserved for early tasks of the remaining jobs scheduled only partially late. Parameter F denotes always, for any $\hat{J}_k$, the completion time of the last early job from $\{\hat{J}_k, \ldots, \hat{J}_u\} \setminus J^P$ on M_1. For jobs $\hat{J}_k \in J^2 \setminus J^P$ (analyzed first) this value is not known yet and it has to be considered as a variable. For jobs from $\hat{J}_k \in J^1 \setminus J^P$, F is calculated based on a current partial solution. Parameter F is necessary to determine a proper initial condition value during the construction of an optimal solution (F becomes A for set J^P). Similarly, parameters L_1, L_2 are equal, for any $\hat{J}_k$, to the number of early units of the partially late tasks of jobs from initial set J^P.

For job $\hat{J}_k \in J^2 \setminus J^P$ processed first on M_2 then on M_1 (i.e. for $k = \tilde{n}, \ldots, u+1$), the recurrence function value $f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$ denotes the maximum amount of the weighted early work of jobs $\{\hat{J}_k, \ldots, \hat{J}_{\tilde{n}}\} \cup J^P$ provided that:

- the first job from this set starts processing exactly at time B on M_2 and not earlier than at time A on M_1 (jobs from $J^1 \setminus J^P$ will be scheduled within time A in the following stages of the DP method),
- there are at least r_2 time units in the interval $[B, d]$ not used for processing jobs from $J^2 \setminus J^P$ on M_2 (within this time second tasks of jobs from J^1 will be scheduled in the following DP stages),
- there are exactly r_1 time units in interval $[A, d]$ reserved for processing jobs from $J^2 \setminus J^P$ on M_1 (all tasks of early jobs from $J^2 \setminus J^P$ have to be executed within this interval),
- the first tasks of tardy jobs from $\{\hat{J}_k, \ldots, \hat{J}_{\tilde{n}}\} \cup J^P$ are processed exactly t_2 time units on M_2 before d and exactly T_2 units are reserved on M_2 before d for the first tasks of tardy jobs $\hat{J}_i$ from $J^2 \setminus J^P$ for $i < k$,
- there are exactly L_1 (L_2) units of partially late tasks on M_1, M_2 (they belong to jobs J_a, J_b or J_x).

Parameters t_1, T_1 are not important at this stage of the analysis (those intervals are embedded within A from $\hat{J}_k$ point of view). They play analogous roles as t_2, T_2 for jobs from J^1 in the following stages of DP. Parameter F denotes the assumed completion time of the last early job from $\{\hat{J}_1, \ldots, \hat{J}_u\} \setminus J^P$ on M_1.
In the presented recurrence relations, all parameters of function $f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$ are bounded by $O(d)$. Thus, determining the recurrence relations for jobs $\hat{J}_k \in J^2 \setminus J^P$ takes $O(d^{11})$ time.

For job $\hat{J}_k \in J^1 \setminus J^P$, processed first on M_1 then on M_2 (i.e. for $k = u, \ldots, 1$), the recurrence function value $f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$ denotes the maximum amount of the weighted early work of jobs $\{\hat{J}_k, \ldots, \hat{J}_{\tilde{n}}\} \cup J^P$ provided that:

- the first job from this set starts processing exactly at time A on M_1 and not earlier than at time B on M_2 (jobs from $J^2 \setminus J^P$ have been scheduled within interval B in DP stages described above),
- there are at least r_1 time units in the interval $[A, d]$ not used for processing jobs from $J^1 \setminus J^P$ on M_1 (within this interval second tasks of jobs from J^2 have been scheduled),
- there are exactly r_2 time units in interval $[B, d]$ reserved for processing jobs $\hat{J}_i$ from $J^1 \setminus J^P$ for $i < k$ on M_2,
- the first tasks of tardy jobs from $\{\hat{J}_k, \ldots, \hat{J}_{\tilde{n}}\} \cup J^P$ are processed exactly t_1 time units on M_1 before d and exactly T_1 units are reserved on M_1 before d for the first tasks of tardy jobs $\hat{J}_i$ from $J^1 \setminus J^P$ for $i < k$,
- there are exactly L_1 (L_2) units of partially late tasks on M_1, M_2 (they belong to jobs J_a, J_b or J_x).

Similarly as in the previous case, parameters t_2, T_2 are not important at this stage of analysis (those intervals are embedded within B from $\hat{J}_k$ point of view). Parameter F denotes the completion times of the last early jobs from $\{\hat{J}_k, \ldots, \hat{J}_u\} \cup J^P$ on M_1.
In the case of jobs from set $J^1 \setminus J^P$, calculating recurrence relations $f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$ takes $O(d^{10})$ time (F is not a variable as for $\hat{J}_k \in J^2 \setminus J^P$).

To determine the maximum weighted late work subject to a given set J^P, one has to select the maximum value of $f_1(0, t_1, 0, r_1, L_1, 0, B, t_2, 0, 0, L_2)$ for $0 \le t_1, r_1, L_1, B, t_2, L_2 \le d$. Function f_1 denotes the weighted early work for all jobs $\{\hat{J}_1, \ldots, \hat{J}_{\tilde{n}}\} \cup J^P$. Changing parameters t_1, L_1, t_2, L_2, we check solutions obtained for all possible amounts of early tasks of late jobs, while changing r_1 and B, we reserve different amounts of time on M_1 and M_2 for jobs from $J^2 \setminus J^P$. Determining the maximal total weighted early work for a particular set J^P takes $O(d^6)$ time.

Complexity of dynamic programming approach

As we have mentioned, initial conditions are calculated in $O(d^6)$ time. Fixing recurrence relations for a single job requires at most $O(d^{11})$ time, but this stage of the algorithm has to be repeated for all jobs from $J \setminus J^P$, i.e. $O(n)$ times. The maximum criterion value can be found in $O(d^6)$ time. Thus, the overall complexity of the described DP stages is

$O(nd^{11})$. These calculations have to be performed for all possible sets J^P, i.e. containing two, one or none job with a task partially late on machine M_1 or M_2, in order to find an optimal solution of the problem under consideration. Consequently, dynamic programming calculations have to be repeated for all $O(n^2)$ two-job sets, all $O(n)$ one-job sets and for an empty set J^P. That gives the complexity $O(n^3d^{11})$. The construction of an optimal schedule does not increase the overall complexity of the dynamic programming approach.

The presented method allows us to find an optimal solution of problem $J2 \mid n_i \leq 2,\ d_i = d \mid Y_w$ in pseudo-polynomial time. Thus, we can classify this scheduling case as binary *NP*-hard [8].

4 Conclusions

The paper presents a dynamic programming approach for the job-shop scheduling problem with the total weighted late work criterion and a common due date $J2 \mid n_i \leq 2,\ d_i = d \mid Y_w$. The *NP*-hardness of the flow-shop problem, $F2 \mid d_i = d \mid Y_w$, being a special case of $J2 \mid n_i \leq 2,\ d_i = d \mid Y_w$, resulted in the *NP*-hardness of the job-shop case. But, it was not settled, whether the latter problem is binary or unary *NP*-hard.
Proposing a solution method with pseudo-polynomial time complexity, we have proven the binary *NP*-hardness of the problem considered. The problem describes a basic situation in revenue management where expected customer orders are known. Ongoing work will consider cases where future demand is uncertain and we are faced the situation to set aside capacity for potentially attractive orders on the risk of loosing the revenue of currently rejected jobs. Customer segmentation policies will change over time as closer the due date approaches.

References

[1] J. Błażewicz, "Scheduling preemptible tasks on parallel processors with information loss". Recherche Technique et Science Informatiques, 3/6, 415-420, 1984.
[2] J. Błażewicz, K. Ecker, E. Pesch, G. Schmidt, J. Węglarz, "Scheduling Computer and Manufacturing Processes". 2nd edn., Springer, Berlin Heidelberg New York, 2001.
[3] J. Błażewicz, G. Finke, " Minimizing mean weighted execution time loss on identical and uniform processors". Information Processing Letters, 24, 259-263, 1987.
[4] J. Błażewicz, E. Pesch, M. Sterna, F. Werner, "Total late work criteria for shop scheduling problems". in: K. Inderfurth, G. Schwödiauer, W. Domschke, F. Juhnke, P. Kleinschmidt, G. Wäscher (Eds.), Operations Research Proceedings 1999, Springer, Berlin, 354-359, 1987.
[5] J. Błażewicz, E. Pesch, M. Sterna, F. Werner, "Open shop scheduling problems with late work criteria". Discrete Applied Mathematics, to appear.
[6] J. Błażewicz, E. Pesch, M. Sterna, F. Werner, "The two-machine flow-shop problem with weighted late work criterion and common due date". European Journal of Operational Research, to appear.
[7] P. Brucker, "Scheduling Algorithms". 2nd edn. Springer, Berlin Heidelberg New York, 1998.
[8] M.R. Garey, D.S. Johnson, "Computers and Intractability". W.H. Freeman and Co., San Francisco, 1979.
[9] J.R. Jackson, "An extension of Johnson's results on job shop scheduling". Naval Research Logistics Quarterly, 3, 201-203, 1956.
[10] S.M. Johnson, "Optimal two- and three-stage production schedules". Naval Research Logistics Quarterly, 1, 61-68, 1954.
[11] J. Józefowska, B. Jurisch, W. Kubiak, "Scheduling shops to minimize the weighted number of late jobs". Operation Research Letters, 16/5, 277-283, 1994.
[12] J.I. McGill, G.J.van Ryzin, "Revenue management. Research overview and prospects". Transportation Science, 33, 233-256, 1999.
[13] M. Pinedo, X. Chao, "Operation Scheduling with Applications in Manufacturing and Services". Irwin/McGraw-Hill, Boston, 1999.
[14] C.N. Potts, L.N. Van Wassenhove, "Single machine scheduling to minimize total late work". Operations Research, 40/3, 586-595, 1991.
[15] C.N. Potts, L.N. Van Wassenhove, "Approximation algorithms for scheduling a single machine to minimize total late work". Operations Research Letters, 11, 261-266, 1991.
[16] M. Sterna, "Problems and Algorithms in Non-Classical Shop Scheduling". Scientific Publishers of the Polish Academy of Sciences, Poznań, 2000.

[17] J. Wirtz, S.E. Kimes, J. Ho Pheng Theng, P. Patterson, "Revenue management: resolving potential customer conflicts". Working Paper, Cornell University 2002.