J Sched (2007) 10:87–95
DOI 10.1007/s10951-006-0005-5

A note on the two machine job shop with the weighted late work criterion

Jacek Blazewicz · Erwin Pesch · Małgorzata Sterna · Frank Werner

Published online: 1 February 2007

Abstract The paper presents a dynamic programming approach for the two-machine nonpreemptive job-shop scheduling problem with the total weighted late work criterion and a common due date ($J2 \mid n_i \leq 2, d_i = d \mid Y_w$), which is known to be NP-hard. The late work performance measure estimates the quality of an obtained solution with regard to the duration of late parts of tasks not taking into account the quantity of this delay. Providing a pseudopolynomial time method for the problem mentioned we can classify it as binary NP-hard.

Keywords Dynamic programming · Job-shop scheduling problem · Late work criterion · Scheduling

M. Sterna has been supported by a KBN grant.

F. Werner has been supported by INTAS (project 03-51-5501).

J. Blazewicz (✉) · M. Sterna
Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland
e-mail: Jacek.Blazewicz@cs.put.poznan.pl

M. Sterna
e-mail: Malgorzata.Sterna@cs.put.poznan.pl

E. Pesch
Institute of Information Systems, FB 5-Faculty of Economics, University of Siegen, Hoelderlinstr. 3, 57068 Siegen, Germany
e-mail: Erwin.Pesch@fb5.uni-siegen.de

F. Werner
Faculty of Mathematics, Otto-von-Guericke-University, PSF4120, 39016 Magdeburg, Germany
e-mail: Frank.Werner@mathematik.uni-magdeburg.de

1 Introduction

Due date involving criteria are performance measures often used in practical applications. 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 the maximum lateness or the total tardiness (cf. Blazewicz et al., 2001; Brucker, 1998; Pinedo and Chao, 1999) 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 (for a survey of results, see Leung, 2004 and Sterna, 2006), although it finds many practical applications, e.g., in data collecting in control systems (Blazewicz, 1984; Blazewicz and Finke, 1987), supporting agriculture technologies (Blazewicz et al., 2004; Potts and van Wassenhove, 1991; Sterna, 2000), or designing production plans within predefined time periods in manufacturing systems (Sterna, 2000).

In this note, we consider the nonpreemptive scheduling problem with the total weighted late work criterion and a common due date in the two-machine job-shop environment (cf. Blazewicz et al., 2001; Brucker, 1998; Pinedo and Chao, 1999), i.e. $J2 \mid n_i \leq 2, d_i = d \mid Y_w$. 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}, 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 sum up late work for all jobs (where $n = |J|$) taking into account their given weights w_i, i.e.:

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

Within our earlier research, we have shown that analogous problems in open-shop (Blazewicz et al., 2004) and flow-shop systems (Blazewicz et al., 2005) are binary NP-hard. With regard to the hardness of the flow-shop problem, the job-shop one (being its generalization) is also computationally hard (Garey and Johnson, 1979). Here, we propose a pseudopolynomial time dynamic programming method solving the problem considered. (This approach is much more sophisticated than the simpler approach for the flow-shop problem and the methods designed for a similar objective function—the weighted number of late jobs (Jozefowska et al., 1994)). That allows us to classify this case as binary NP-hard and to finish the research on two-machine weighted shop scheduling problems with a common due date.

2 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 (1956), which is optimal from the schedule length point of view. 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 (1954), so within sets J^1 and J^2 all jobs J_i with $p_{i1} \le p_{i2}$ are sequenced in nondecreasing order of p_{i1}, while the rest, with $p_{i1} > p_{i2}$, are scheduled in nonincreasing order of p_{i2}. Moreover, we use the fact (Blazewicz et al., 2005) 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 observations, 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. Consequently, there are only three possible schemes of an optimal solution, which have to be compared. 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). Then, to find an optimal sequence of the 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 an element of J^P.

Summing up, to construct 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 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 last jobs in Jackson's order in both sets, i.e., $\hat{J}_{\tilde{n}}$ and J_u, are treated in a special way. 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 the remaining jobs are performed between those early ones and J^P in an arbitrary order.

2.1 Initial conditions

The weighted early work corresponding to the set of jobs with partially late tasks, J^P, 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

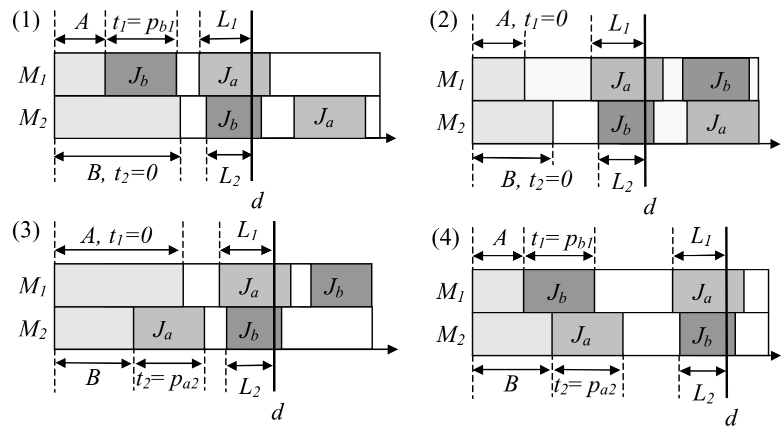


Fig. 1 Initial conditions for different sets $J^P = \{J_a, J_b\}$

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. As already mentioned, there are three possible cases, when set J^P contains two, one, or none job. 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.

Case I.1. $J^P = \{J_a, J_b\}$ (cf. Fig. 1)
Case I.1.1. $J_a \in J^1$, $J_b \in J^1$ (cf. Fig. 1(1))
if $(0 \le A \le \min\{d - L_1, d - L_2\} - p_{b1}) \wedge (t_1 = p_{b1}) \wedge (0 < L_1 < p_{a1}) \wedge (0 \le B \le d - L_2) \wedge (t_2 = 0) \wedge (0 < L_2 < p_{b2})$,

$$\text{then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = w_a L_1 + w_b(p_{b1} + L_2) \quad (1)$$

$$\text{else } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = -\infty \quad (2)$$

Case I.1.2. $J_a \in J^1$, $J_b \in J^2$ (cf. Fig. 1(2))
if $(0 \le A \le d - L_1) \wedge (t_1 = 0) \wedge (0 < L_1 < p_{a1}) \wedge (0 \le B \le d - L_2) \wedge (t_2 = 0) \wedge (0 < L_2 < p_{b2})$,

$$\text{then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = w_a L_1 + w_b L_2 \quad (3)$$

$$\text{else } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = -\infty \quad (4)$$

Case I.1.3. $J_a \in J^2$, $J_b \in J^2$ (cf. Fig. 1(3))
if $(0 \le A \le d - L_1) \wedge (t_1 = 0) \wedge (0 < L_1 < p_{a1}) \wedge (0 \le B \le \min\{d - L_1, d - L_2\} - p_{a2}) \wedge (t_2 = p_{a2}) \wedge (0 < L_2 < p_{b2})$,

$$\text{then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = w_a(p_{a2} + L_1) + w_b L_2 \quad (5)$$

$$\text{else } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = -\infty \quad (6)$$

Case I.1.4. $J_a \in J^2$, $J_b \in J^1$ (cf. Fig. 1(4))
if $(0 \le A \le \min\{d - L_1, d - L_2\} - p_{b1}) \wedge (t_1 = p_{b1}) \wedge (0 < L_1 < p_{a1}) \wedge (0 \le B \le \min\{d - L_1, d - L_2\} - p_{a2}) \wedge (t_2 = p_{a2}) \wedge (0 < L_2 < p_{b2})$,

$$\text{then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = w_a(p_{a2} + L_1) + w_b(p_{b1} + L_2) \quad (7)$$

$$\text{else } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = -\infty \quad (8)$$

Determining the initial value of the weighted early work, we count early parts of jobs J_a, J_b for feasible values of parameters A, B, t_1, t_2, L_1, and L_2 (Terms 1, 3, 5, and 7). Early tasks (if any) on M_1, M_2 must fit exactly intervals t_1, t_2, respectively. Similarly, partially late tasks on M_1, M_2 have to fit exactly intervals L_1, L_2. Finally, A and B have to be properly chosen to ensure that jobs J_a, J_b are the jobs with partially late tasks. Infeasible parameter values (Terms 2, 4, 6, and 8) lead to the initial criterion value equal to minus infinity. That means that such solutions are rejected.

Case I.2. $J^P = \{J_x\}$ (cf. Fig. 2)
Case I.2.1. $J_x \in J^1$ (cf. Figs. 2(1) and (2))
if $(0 \le A \le d - L_2 - p_{x1}) \wedge (t_1 = p_{x1}) \wedge (L_1 = 0) \wedge (0 \le B \le d - L_2) \wedge (t_2 = 0) \wedge (0 < L_2 < p_{x2})$,

$$\text{then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = w_x(p_{x1} + L_2) \quad (9)$$

if $(0 \le A \le d - L_1) \wedge (t_1 = 0) \wedge (0 < L_1 < p_{x1}) \wedge (0 \le B \le d) \wedge (t_2 = 0) \wedge (L_2 = 0)$,

$$\text{then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = w_x L_1 \quad (10)$$

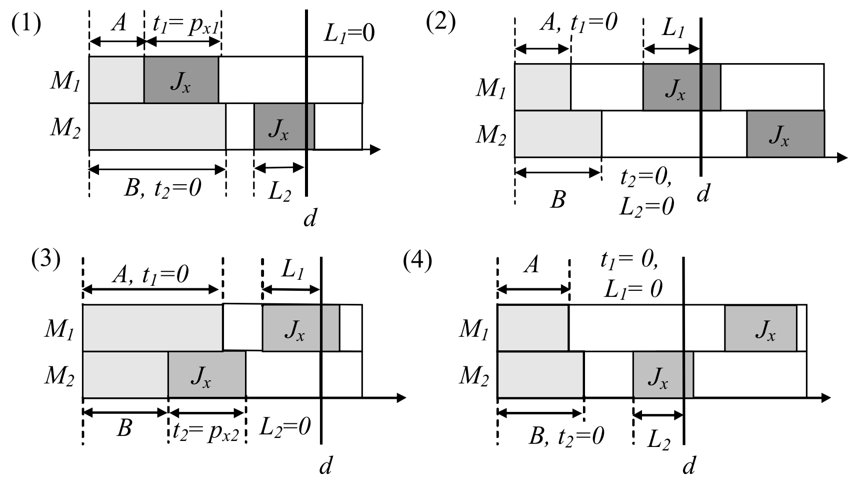
$$\text{if otherwise, then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = -\infty \quad (11)$$

Similarly as for a two-job set J^P, we detect infeasible parameter values (Term 11). For feasible parameter values, we check two possible ways of scheduling job J_x: with a partially late task on M_2 (Term 9, Fig. 2(1)) and with a partially late task on M_1 (Term 10, Fig. 2(2)).

Case I.2.2. $J_x \in J^2$ (cf. Figs. 2(3) and (4))
if $(0 \le A \le d - L_1) \wedge (t_1 = 0) \wedge (0 < L_1 < p_{x1}) \wedge (0 \le B \le d - L_1 - p_{x2}) \wedge (t_2 = p_{x2}) \wedge (L_2 = 0)$,

$$\text{then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = w_x(p_{x2} + L_1) \quad (12)$$

Fig. 2 Initial conditions for $J_x \in J^1$, when J_x is partially late on M_2 (1) and on M_1 (2) and for $J_x \in J^2$, when J_x is partially late on M_1 (3) and on M_2 (4)



$$\text{if}\,(0 \le A \le d) \wedge (t_1 = 0) \wedge (L_1 = 0) \wedge (0 \le B \le d - L_2) \wedge (t_2 = 0) \wedge (0 < L_2 < p_{x2}),$$

$$\text{then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = w_x L_2 \tag{13}$$

$$\text{if otherwise, then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = -\infty \tag{14}$$

As in the previous case, we detect infeasible parameter values (Term 14) and for feasible parameter values, we check two possible ways of scheduling job J_x: with a partially late task on M_1 (Term 12, Fig. 2(3)) and on M_2 (Term 13, Fig. 2(4)).

Case I.3. $J^P = \emptyset$

Finally, we have to analyze the case when no partially late task exists in the system, for which the initial conditions are formulated as follows. 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.

$$\text{if}\,(0 \le A \le d) \wedge (t_1 = 0) \wedge (L_1 = 0) \wedge (0 \le B \le d) \wedge (t_2 = 0) \wedge (L_2 = 0),$$

$$\text{then } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = 0 \tag{15}$$

$$\text{else } f_{\tilde{n}+1}(A, t_1, L_1, B, t_2, L_2) = -\infty \tag{16}$$

2.2 Recurrence relations

After determining the initial conditions for a particular set J^P, we calculate the recurrence relations for the remaining jobs $\hat{J}_k \in J \backslash J^P$, numbered according to Jackson's rule as $\hat{J}_1, \ldots, \hat{J}_u, \hat{J}_{u+1}, \ldots, \hat{J}_{\tilde{n}}$. As already 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 $\hat{J}_k \in J^2 \backslash J^P$ or $\hat{J}_k \in J^1 \backslash J^P$. Parameter F always denotes the completion time of the last early job from $\{\hat{J}_k, \ldots, \hat{J}_u\} \backslash J^P$ on M_1. For jobs $\hat{J}_k \in J^2 \backslash J^P$ (analyzed first), this value is not known yet and it has to be considered as a variable. For jobs from $\hat{J}_k \in J^1 \backslash J^P$, 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).

Case R.1. $k = \tilde{n}, \ldots, u+1$ (i.e. $\hat{J}_k \in J^2 \backslash J^P$)

For job $\hat{J}_k \in J^2 \backslash J^P$, processed first on M_2 then on M_1, $f_k(A, t_1, T_1, r_1, T_1, F, B, t_2, T_2, r_2, L_2)$ 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 $J^1 \backslash J^P$ will be scheduled within time A in the following DP stages),
- there are at least r_2 time units in the interval $[B, d]$ not used for processing jobs from $J^2 \backslash J^P$ 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 the interval $[A, d]$ reserved for processing jobs from $J^2 \backslash J^P$ on M_1 (all tasks of early jobs from $J^2 \backslash J^P$ 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 $\hat{J}_i \in J^2 \backslash J^P$ $(i < k)$,
- there are exactly L_1 (L_2) units of partially late tasks on M_1, M_2 (they belong to jobs from J^P).

Parameters t_1, T_1 are not important at this stage of the analysis (those intervals are embedded within A from $\hat{J}_k$'s 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 $\{\hat{J}_1, \ldots, \hat{J}_u\} \backslash J^P$ on M_1. Jobs $\hat{J}_k \in J^2 \backslash J^P$ are analyzed from $k = \tilde{n}$ to $u + 1$. Determining the recurrence relation f_k for $\hat{J}_k$, we use the result obtained for $\hat{J}_{k+1}(f_{k+1})$. For this reason, the formulation of the recurrence relations for the last job $\hat{J}_{\tilde{n}}$, requiring the result of the initial condition calculation $f_{\tilde{n}+1}$, is slightly different. It is calculated as the first one. However, for sake of clarity, we will present it later.

For jobs $\hat{J}_k \in J^2 \backslash J^P$, where $k = \tilde{n} - 1, \ldots, u + 1$, recurrence relations are as follows:

if $(B + t_2 + T_2 + r_2 + L_2 \le d) \wedge (A + r_1 + L_1 \le d) \wedge (t_1 + T_1 \le A) \wedge (F \le A - (t_1 + T_1))$, then
if $(B + p_{k2} + t_2 + T_2 + r_2 + L_2 \le d) \wedge (\max\{A, B + p_{k2}\} + p_{k1} + L_1 \le d) \wedge (p_{k1} \le r_1)$, then

$$
\begin{aligned}
&f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)\\
&\quad = \max\{w_k(p_{k1} + p_{k2}) + f_{k+1}(\max\{A, B + p_{k2}\}\\
&\qquad + p_{k1}, t_1, T_1, r_1 - p_{k1}, L_1, F, B + p_{k2},\\
&\qquad t_2, T_2, r_2, L_2), \qquad (17)\\
&\quad w_k p_{k2} + f_{k+1}(A, t_1, T_1, r_1, L_1, F, B, t_2 - p_{k2},\\
&\qquad T_2 + p_{k2}, r_2, L_2) \quad \text{if } p_{k2} \le t_2, \qquad (18)\\
&\quad f_{k+1}(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)\} \qquad (19)
\end{aligned}
$$

$$
\begin{aligned}
&\text{else } f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)\\
&\quad = \max\{w_k p_{k2} + f_{k+1}(A, t_1, T_1, r_1, L_1, F, B, t_2 - p_{k2},\\
&\qquad T_2 + p_{k2}, r_2, L_2) \text{ if } p_{k2} \le t_2, \qquad (20)\\
&\quad f_{k+1}(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)\} \qquad (21)\\
&\text{else } f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2) = -\infty \qquad (22)
\end{aligned}
$$

If parameter values are infeasible (i.e., gaps between A, B and d are not long enough to contain intervals r_1, L_1 and t_2, T_2, r_2, L_2, respectively, or A is too small to contain t_1, T_1, or the assumed completion time F of last early job from $J^1 \backslash J^P$ on M_1 is too big), then the function takes the value minus infinity (Term 22). Otherwise, we have to check all possible ways of scheduling job $\hat{J}_k$ and select the best one (ensuring the maximum weighted early work). If job $\hat{J}_k$ can be scheduled early (Terms 17–19, Fig. 3), then we compare three possible subschedules, when this job is early (Term 17, Fig. 3(1)), only its first task is early (Term 18, Fig. 3(2)), and the job is totally late (Term 19, Fig. 3(3)). The case when $\hat{J}_k$ is early only on M_2 is under consideration only, if interval t_2 is long enough to contain the whole task of $\hat{J}_k$. If job $\hat{J}_k$ cannot be scheduled early (Terms 20, 21), then only two cases are possible when only its first task is early (assuming that t_2 is long enough, Term 20) or it is totally late (Term 21).

As mentioned, recurrence relations for the last job from set $J^2 \backslash J^P$ ($\hat{J}_{\tilde{n}}$) are formulated differently:

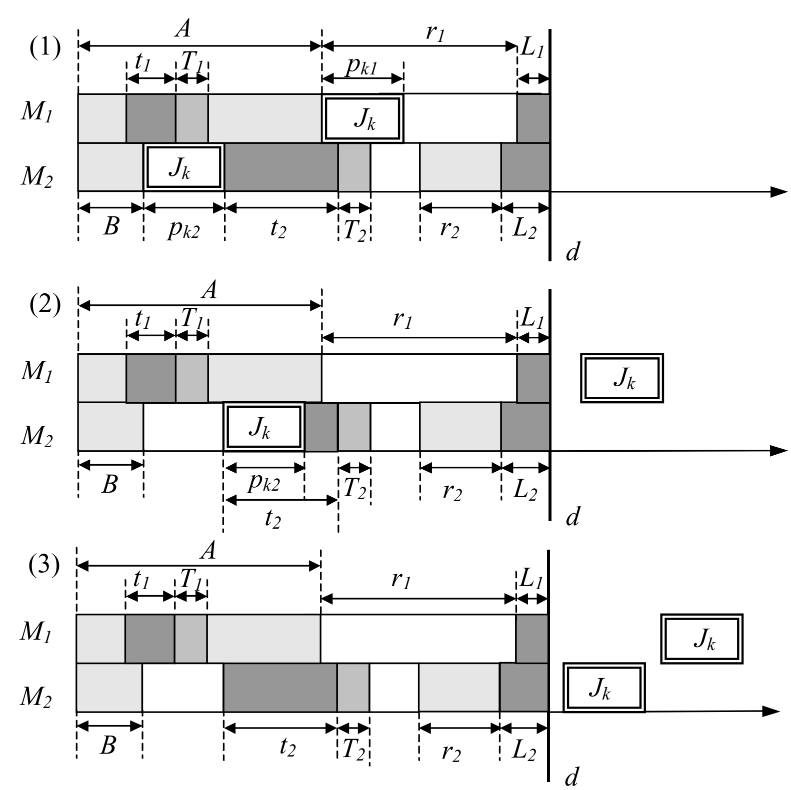


Fig. 3 Recurrence relations for $\hat{J}_k \in J^2 \backslash J^P$ executed early (1), early only on M_2 (2), or totally late (3)

if $(B + t_2 + T_2 + r_2 + L_2 \le d) \wedge (A + r_1 + L_1 \le d)$
$\wedge\, (t_1 + T_1 \le A) \wedge (F \le A - (t_1 + T_1))$, then
if $(B+p_{\tilde n2}+t_2+T_2+r_2+L_2 \le d) \wedge (\max\{A, B + p_{\tilde n2}\} + p_{\tilde n1} + L_1 \le d) \wedge (p_{\tilde n1} \le r_1)$, then

$$f_{\tilde n}(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$$
$$= \max\{w_{\tilde n}(p_{\tilde n1} + p_{\tilde n2}) + f_{\tilde n+1}(F, t_1, L_1, B + p_{\tilde n2}, t_2, L_2), \tag{23}$$
$$w_{\tilde n} p_{\tilde n2} + f_{\tilde n+1}(F, t_1, L_1, B, t_2 - p_{\tilde n2}, L_2) \quad \text{if } p_{\tilde n2} \le t_2, \tag{24}$$
$$f_{\tilde n+1}(F, t_1, L_1, B, t_2, L_2)\} \tag{25}$$

else $f_{\tilde n}(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$
$$= \max\{w_{\tilde n} p_{\tilde n2} + f_{\tilde n+1}(F, t_1, L_1, B, t_2 - p_{\tilde n2}, L_2) \quad \text{if } p_{\tilde n2} \le t_2 \tag{26}$$
$$f_{\tilde n+1}(F, t_1, L_1, B, t_2, L_2)\} \tag{27}$$

else $f_{\tilde n}(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2) = -\infty$ (28)

The case study for $\hat J_{\tilde n}$ is identical to that for other jobs $\hat J_k \in J^2 \setminus J^P$. The only difference is that while calculating $f_{\tilde n}$, we determine the weighted early work for jobs $\{\hat J_{\tilde n}\} \cup J^P$ and we have to use the criterion value $f_{\tilde n+1}$ calculated for J^P (not f_{k+1} calculated for another job $\hat J_{k+1} \in J^2 \setminus J^P$). Function $f_{\tilde n+1}$ is defined for a different parameter set than the recurrence relation f_k. Parameters representing reserved intervals T_1, r_1, T_2, r_2 are not important for J^P. Parameter F is used for determining the possible starting time on M_1 for J^P, i.e., for determining the value of parameter A for J^P. The possible starting time on M_2 for J^P results from a current partial schedule, i.e., the value of parameter B for J^P results from the value of B for $\hat J_{\tilde n}$ increased by $p_{\tilde n2}$, if $\hat J_{\tilde n}$ is early on M_2. Similarly as for $\hat J_{\tilde n}$, we have to change the formulation of the recurrence relations for jobs $\hat J_k \in J^2 \setminus J^P$ that contain only one task, requiring machine M_2 ($p_{k1} = 0$). In those cases, precedence constraints do not exist and we remove Term 17 from the definition of f_k (and Term 23, if $\hat J_{\tilde n}$ contains only one task).

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. The analysis of $J^2 \setminus J^P$ is followed by an analysis of jobs with the first (only) task on machine M_1. As mentioned earlier, recurrence relations have to be adjusted to a different type of precedence constraints between tasks.

Case R.2. $k = u, \dots, 1$ (i.e. $\hat J_k \in J^1 \setminus J^P$)
For job $\hat J_k \in J^1 \setminus J^P$, processed first on M_1 then on M_2, $f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$ denotes the maximum amount of the weighted early work of jobs $\{\hat J_k, \dots, \hat J_{\tilde n}\} \cup J^P$ 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 earlier),
– 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 $\hat J_i \in J^1 \setminus J^P$ on M_2 ($i < k$),
– the first tasks of tardy jobs from $\{\hat J_k, \dots, \hat J_{\tilde n}\} \cup J^P$ 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 $\hat J_i \in J^1 \setminus J^P (i < k)$,
– there are exactly L_1 (L_2) units of partially late tasks on M_1, M_2 (they belong to jobs from J^P).

Similarly as in Case R.1, 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 time of the last early job from $\{\hat J_k, \dots, \hat J_u\} \cup J^P$ on M_1. Jobs $\hat J_k \in J^1 \setminus J^P$ are analyzed from $k = u$ to 1. Again, determining the recurrence relation f_k for $\hat J_k$, we use the result obtained for $\hat J_{k+1}$ (f_{k+1}). For this reason, the recurrence relation formulation for the last job $\hat J_u$, requiring value f_{u+1}, is slightly different, because $\hat J_{u+1}$ belongs to $J^2 \setminus J^P$ not to $J^1 \setminus J^P$. The value of f_u is calculated as the first one. However, for the sake of clarity, we will present it later, as in Case R.1.

For jobs $\hat J_k \in J^1 \setminus J^P$, where $k = u - 1, \dots, 1$, recurrence relations are as follows:

if $(A + t_1 + T_1 + r_1 + L_1 \le d) \wedge (B + r_2 + L_2 \le d)$
$\wedge\, (t_2 \le B) \wedge (T_2 = 0)$, then
if $(A + p_{k1} + t_1 + T_1 + r_1 + L_1 \le d)$
$\wedge\, (\max\{A + p_{k1}, B\} + p_{k2} + r_2 + L_2 \le d)$, then

$$f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$$
$$= \max\{w_k(p_{k1}+p_{k2}) + f_{k+1}(A + p_{k1}, t_1, T_1, r_1, L_1, A+p_{k1}, \max\{A+p_{k1}, B\}+p_{k2}, t_2, T_2, r_2 + p_{k2}, L_2), \tag{29}$$
$$w_k p_{k1} + f_{k+1}(A, t_1 - p_{k1}, T_1 + p_{k1}, r_1, L_1, A, B, t_2, T_2, r_2, L_2) \text{ if } p_{k1} \le t_1, \tag{30}$$
$$f_{k+1}(A, t_1, T_1, r_1, L_1, A, B, t_2, T_2, r_2, L_2)\} \tag{31}$$

else $f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2, F)$
$$= \max\{w_k p_{k1} + f_{k+1}(A, t_1 - p_{k1}, T_1 + p_{k1}, r_1, L_1, A, B, t_2, T_2, r_2, L_2) \quad \text{if } p_{k1} \le t_1, \tag{32}$$
$$f_{k+1}(A, t_1, T_1, r_1, L_1, A, B, t_2, T_2, r_2, L_2)\} \tag{33}$$

else $f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2) = -\infty$ (34)

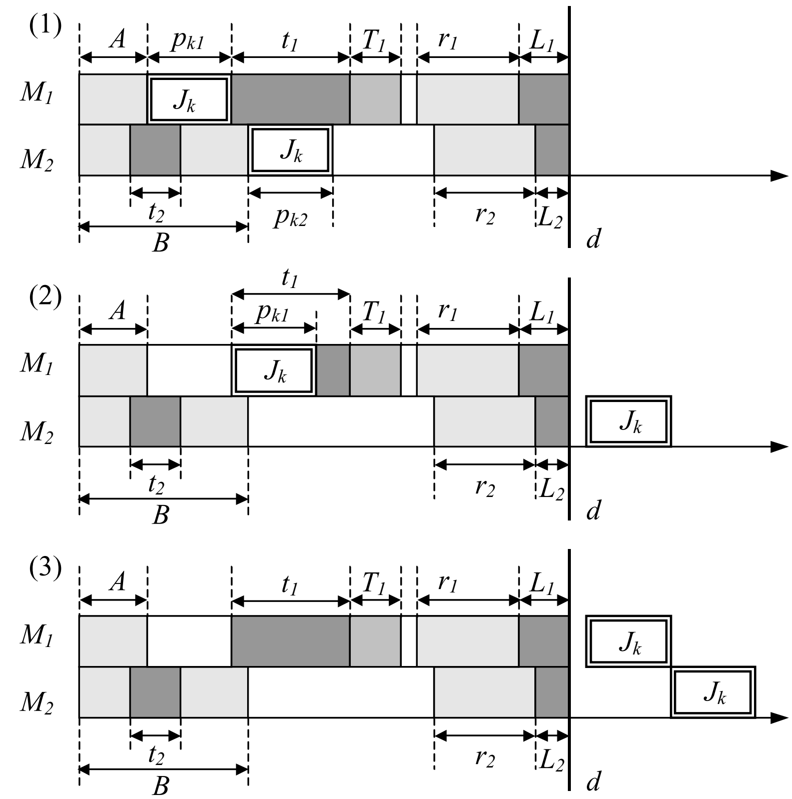


Fig. 4 Recurrence relations for $\hat{J}_k \in J^1 \setminus J^P$ executed early (1), early only on M_1 (2), or totally late (3)

If parameter values are infeasible (i.e., gaps between A, B, and d are not long enough to contain intervals t_1, T_1, r_1, L_1, and r_2, L_2 respectively, or B is too small to contain t_2, or T_2 is different from 0, i.e., there is a reserved interval for jobs from J^2, although all jobs from this set have been already considered), then the function takes the minus infinity value (Term 34). Otherwise, we have to analyze all possible ways of executing job $\hat{J}_k$ and choose the best one subject to the weighted early work. If job $\hat{J}_k$ can be scheduled early (Terms 29–31, Fig. 4) then we compare three possible solutions, namely, when this job is early (Term 29, Fig. 4(1)), only its first task is early (Term 30, Fig. 4(2)), and the job is totally late (Term 31, Fig. 4(3)). The case when $\hat{J}_k$ is early only on M_1 is under consideration only, if interval t_1 is long enough to contain the whole task of $\hat{J}_k$. If job $\hat{J}_k$ cannot be scheduled early (Terms 32, 33), then two cases are possible: when only its first task is early (assuming that t_1 is long enough, Term 32) or it is totally late (Term 33).

As we have announced, recurrence relations for the last job from set $J^1 \setminus J^P$, i.e. job $\hat{J}_u$, are formulated differently:

if $(A + t_1 + T_1 + r_1 + L_1 \le d) \wedge (B + r_2 + L_2 \le d)$
$\wedge (t_2 \le B) \wedge (T_2 = 0)$, then
if $(A + p_{u1} + t_1 + T_1 + r_1 + L_1 \le d) \wedge (\max\{A + p_{u1}, B\}$
$+ p_{u2} + r_2 + L_2 \le d)$, then

$$f_u(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2, F)$$

$$= \max\{w_u(p_{u1} + p_{u2}) + f_{u+1}(A + p_{u1} + t_1 + T_1, t_1, T_1, d - (A + p_{u1} + t_1 + T_1 + L_1), L_1, A + p_{u1}, 0, t_2, T_2, r_2 + p_{u2}, L_2) \quad (35)$$

$$w_u p_{u1} + f_{u+1}(A + t_1 + T_1, t_1 - p_{u1}, T_1 + p_{u1}, d - (A + p_{u1} + t_1 + T_1 + L_1), L_1, A + p_{u1}, 0, t_2, T_2, r_2, L_2) \quad \text{if } p_{u1} \le t_1 \quad (36)$$

$$f_{u+1}(A + t_1 + T_1, t_1, T_1, d - (A + t_1 + T_1 + L_1), L_1, A, 0, t_2, T_2, r_2, L_2)\} \quad (37)$$

else $f_u(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$

$$= \max\{w_u p_{u1} + f_{u+1}(A + t_1 + T_1, t_1 - p_{u1}, T_1 + p_{u1}, d - (A + p_{u1} + t_1 + T_1 + L_1), L_1, A + p_{u1}, 0, t_2, T_2, r_2, L_2) \quad \text{if } p_{u1} \le t_1 \quad (38)$$

$$f_{u+1}(A + t_1 + T_1, t_1, T_1, d - (A + t_1 + T_1 + L_1), L_1, A, 0, t_2, T_2, r_2, L_2)\} \quad (39)$$

else $f_u(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2) = -\infty \quad (40)$

We consider for $\hat{J}_u$, the same cases as for other jobs $\hat{J}_k \in J^1 \setminus J^P$, but determining the weighted early work for jobs

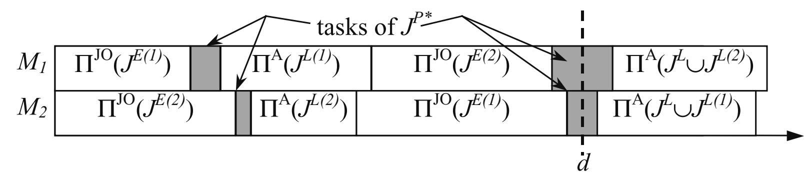


Fig. 5 The structure of an optimal solution

$\{\hat{J}_u, \hat{J}_{u+1}, \ldots, \hat{J}_{\tilde{n}}\} \cup J^P$, we have to use the criterion value f_{u+1} calculated for a job from set $J^2 \setminus J^P$. Switching from set $J^1 \setminus J^P$ to $J^2 \setminus J^P$, i.e., calling function f_{u+1}, we assume that B equals 0, because early jobs from $J^2 \setminus J^P$ start at time 0 on M_2. Then, A is extended with intervals t_1 and T_1 (parameter A takes the value $A + t_1 + T_1$ or $A + t_1 + T_1 + p_{u1}$ for job $\hat{J}_{u+1}$, depending on the way $\hat{J}_u$ is scheduled). From the point of view of job $\hat{J}_{u+1}$, all jobs from $J^1 \setminus J^P$ have to be executed before A, despite the fact whether they are early or partially late. The completion time of the last early job from $\hat{J}_k \in J^1 \setminus J^P$ is stored as value F for J_{u+1}, equal to A or $A + p_{u1}$ depending on the way J_u is scheduled. Finally, we determine interval r_1, not used by jobs from $J^1 \setminus J^P$, as $d - (A + p_{u1} + t_1 + T_1 + L_1)$ or $d - (A + t_1 + T_1 + L_1)$, depending on the way $\hat{J}_u$ is executed. For $\hat{J}_{u+1}$, we have to know exactly the length of the interval not used by jobs from $J^1 \setminus J^P$.

As for $\hat{J}_k \in J^2 \setminus J^P$, in the case of $\hat{J}_k \in J^1 \setminus J^P$ having only one task, requiring machine M_1 ($p_{k2} = 0$), we remove Term 29 from the definition of f_k (and Term 35, if job J_u contains only one task). Similarly, calculating recurrence relations $f_k(A, t_1, T_1, r_1, L_1, F, B, t_2, T_2, r_2, L_2)$ for jobs from set $J^1 \setminus J^P$ 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 $\{\hat{J}_1, \ldots, \hat{J}_{\tilde{n}}\} \cup J^P$ subject to 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.

2.3 Solution construction and complexity of the algorithm

To find an optimal solution of problem $J2 \mid n_i \le 2, d_i = d \mid Y_w$, we have to analyze all possible sets J^P of jobs with partially late tasks on machines M_1, M_2. 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. In each case, DP calculations require first the initial conditions determination in $O(d^6)$ time. Fixing recurrence relations for all $O(n)$ jobs from $J \setminus J^P$ takes $O(d^{11})$ time and then determining the maximum criterion value for a certain set J^P can be done in $O(d^6)$ time. The overall complexity of this stage of the dynamic programming method is $O(n^3 d^{11})$.

After determining the set J^{P^*}, which results in a schedule with the maximum (optimal) weighted early work, we have to construct an optimal solution based on the decisions taken during the DP calculations for J^{P^*}. They divided $J \setminus J^{P^*}$ into five subsets (cf. Fig. 5):

- $J^{E(1)}$, $J^{E(2)}$ with early jobs from J^1, J^2, respectively,
- $J^{L(1)}$, $J^{L(2)}$ with jobs from $J^1 \setminus J^{P^*}$, $J^2 \setminus J^{P^*}$ whose first task is early and the second one is totally late,
- J^L with totally late jobs.

To build a schedule on machine M_1, denoted as Π_1, first we execute early jobs from $J^{E(1)}$ in Jackson's order obtaining subschedule $\Pi^{JO}(J^{E(1)})$. It is followed by the early task of a job from J^{P^*} and, then, by tasks from $J^{L(1)}$ executed in arbitrary order (subschedule $\Pi^{A}(J^{L(1)})$). After those tasks of partially late jobs, we perform the second tasks of early jobs from $J^{E(2)}$ in Jackson's order obtaining subschedule $\Pi^{JO}(J^{E(2)})$. Then, the partially late task of a job from J^{P^*} has to be scheduled followed by arbitrarily ordered late tasks of jobs from $J^L \cup J^{L(2)}$ (subschedule $\Pi^{A}(J^L \cup J^{L(2)})$). Schedule Π_2 on machine M_2 is constructed in a similar way. Depending on a problem instance, some subschedules mentioned earlier may be empty. The construction of an optimal schedule does not increase the overall complexity of the dynamic programming approach.

3 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 $J2 \mid n_i \le 2, d_i = d \mid Y_w$. The NP-hardness of the flow-shop problem, $F2 \mid d_i = d \mid Y_w$, being a special case of $J2 \mid n_i \le 2, d_i = d \mid Y_w$, 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 dynamic programming method with pseudopolynomial time complexity, we have proven the binary NP-hardness of the problem considered.

References

Blazewicz, J., "Scheduling preemptible tasks on parallel processors with information loss," *Recherche Technique et Science Informatiques*, **3**(6), 415–420 (1984).

Blazewicz, J., K. Ecker, E. Pesch, G. Schmidt, and J. Weglarz, *Scheduling Computer and Manufacturing Processes*. 2nd edn. Springer, Berlin, Heidelberg, New York, 2001.

Blazewicz. J. and G. Finke, "Minimizing mean weighted execution time loss on identical and uniform processors," *Information Processing Letters*, **24**, 259–263 (1987).

Blazewicz, J., E. Pesch, M. Sterna, and F. Werner, "Open shop scheduling problems with late work criteria," *Discrete Applied Mathematics*, **134**, 1–24 (2004).

Blazewicz, J., E. Pesch, M. Sterna, and F. Werner, "The two-machine flow-shop problem with weighted late work criterion and common due date," *European Journal of Operational Research*, **165**(2), 408–415 (2005).

Brucker, P., *Scheduling Algorithms*. 2nd edn. Springer, Berlin, Heidelberg, New York, 1998.

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

Jackson, J. R., "An extension of Johnson's results on job shop scheduling," *Naval Research Logistics Quarterly*, **3**, 201–203 (1956).

Johnson, S. M., "Optimal two- and three-stage production schedules with setup times included," *Naval Research Logistics Quarterly*, **1**, 61–68 (1954).

Jozefowska, J., B. Jurisch, and W. Kubiak, "Scheduling shops to minimize the weighted number of late jobs," *Operation Research Letters*, **16**(5), 277–283 (1994).

Leung, J. Y. T., "Minimizing total weighted error for imprecise computation tasks and related problems," in: J. Y. T. Leung (ed.), *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. CRC Press, Boca Raton, 2004; Chapter 34:1–16.

Pinedo, M. and X. Chao, *Operation Scheduling with Applications in Manufacturing and Services*. Irwin/McGraw-Hill, Boston (1999).

Potts, C. N. and L. N. van Wassenhove, "Single machine scheduling to minimize total late work," *Operations Research*, **40**(3), 586–595 (1991).

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

Sterna, M., *Late Work Scheduling in Shop Systems.* Publishing House of Poznan University of Technology, Poznan (2006).