

Stable scheduling of single machine with probabilistic parameters

W. BOŻEJKO^{1*}, P. RAJBA², and M. WODECKI²

¹Department of Control Systems and Mechatronics, Faculty of Electronics, Wrocław University of Science and Technology,
Wyb. Wyspiańskiego, 50-370 Wrocław, Poland

²Institute of Computer Science, University of Wrocław, 15 Joliot-Curie St., 50-383 Wrocław, Poland

Abstract. We consider a stochastic variant of the single machine total weighted tardiness problem jobs parameters are independent random variables with normal or Erlang distributions. Since even deterministic problem is NP-hard, it is difficult to find global optimum for large instances in the reasonable run time. Therefore, we propose tabu search metaheuristics in this work. Computational experiments show that solutions obtained by the stochastic version of metaheuristics are more stable (i.e. resistant to data disturbance) than solutions generated by classic, deterministic version of the algorithm.

Key words: scheduling, uncertain parameters, tabu search, stability.

1. Introduction

Single machine scheduling problems with cost goal functions, despite the simplicity of their formulation, belong to the most difficult class of (NP-hard) combinatorial optimization problems. In the literature there are many such problems with different parameters of tasks, functional properties of machines and different goal functions. Starting from the simplest problems – with one constraint (e.g. problem denoted by $1||\sum w_i U_i$, concerning the latest tasks' due dates), to complex problems of set-ups and time windows, where in the optimal solution machines may have downtimes. They are important both from theoretical and practical standpoint because:

1. they enable modeling and analysis of simple production systems and individual positions in more complex systems,
2. they are considered as special cases of more general problems,
3. some of their particular properties can often be generalized,
4. they provide a common basis for the methods for solving many NP-hard problems,
5. they are not only simple in implementation but there is also a number of test examples in the literature.

Single machine problems are important from the theory and practice point of view, because: (1) they allows us to research single production nest of the manufacturing system (bottleneck), (2) their specific properties can be generalized to multi-machine issues, (3) many transportation and logistic problems (e.g. variations of traveling salesman problem) are formulated as single machine problems with setup times and additional constrains. Examples of their practical applications were presented by Lann and Mosheiov [14].

We consider the single machine total weighted tardiness problem (in short – TWT), denoted as $1||\sum w_i T_i$. A set of n jobs

$J = \{1, 2, \dots, n\}$ have to be processed without interruption on a single machine that can handle only one job at a time. All jobs become available for processing at the beginning (time zero). Each job i has an integer processing time p_i , a due date d_i and a positive weight w_i . For a given sequence of jobs (earliest) due date C_i , the tardiness $T_i = \max\{0, C_i - d_i\}$ and the cost $w_i \cdot T_i$ of job $i \in J$. The objective is to find a job sequence which minimizes the sum of the costs $\sum_{i=1}^n w_i \cdot T_i$. This is a classical problem of scheduling theory.

2. Literature review

The total weighted tardiness problem is NP-hard [15]. Enumerative algorithms (which use dynamic programming and branch and bound approaches) for the problem are described in [22, 29]. The algorithms are a significant improvement over exhaustive search but they remain laborious and are applicable only to relatively small problems, with the number of jobs not exceeding 50 (80 in a multi-processor computer [29]). The enumerative algorithms mentioned above may require considerable computer resources both in terms of computation times and core storage. Therefore, many algorithms have been proposed to find near optimal schedules in reasonable time.

Local search methods start from an initial solution and if repeatedly try to improve the current solution by local changes. The interchanges are continued until a solution that cannot be improved is obtained, which is a local minimum. To increase the performance of local search algorithms, metaheuristics like tabu search are used [2, 7], simulated annealing [23], path relinking [4], genetic algorithms [7], ant colony optimization [9]. A very effective iterated local-search method has been proposed by Kirlik and Oguz [13]. The key aspect of the method is its ability to explore an exponential-size neighborhood in polynomial time by a dynamic programming technique.

*e-mail: wojciech.bozejko@pwr.edu.pl

In many applications, serious difficulties occur while indicating parameters or when the data comes from inaccurate measurement equipment. Due to short realization terms, short series and production elasticity there are no comparative data and no possibility to conduct experimental studies that would enable one to determine explicit values of certain parameters. Furthermore, in many economy branches like tourism, agriculture, commerce, building industry, etc., the processes that occur have, by their nature, random character (they depend on weather, market conditions, accidents, etc.). Making decisions in the conditions of uncertainty (lack of exact values of parameters) becomes quotidian. Stochastic scheduling problems, where job durations are random variables with known probability distributions have been studied in the literature for more than 40 years. A comprehensive review of methods and algorithms solving combinatorial optimization problems with random parameters were presented by: Dean [8], Vondrák [28] and Pinedo [21]. In case of single machine problems it is usually assumed that the parameters of the process (eg. release dates, times of tasks performances or due dates, etc.) are independent random variables. Generally, there are two types of objective function considered in the literature:

1. regular – non-negative, non-decreasing function (of completion times). Some typical examples are given below:
 - expected number of late/tardy jobs [10, 27],
 - expected weighted number of late/tardy jobs [17],
 - expected total weighted tardiness [5, 6],
 - total weight of batches of jobs [11],
2. non-regular (in the context in just-in-time scheduling):
 - expected number of early and tardy jobs [25],
 - expected total weighted number of early and tardy jobs [26],
 - expected weighted sum of earliness and tardiness [1, 18].

In this paper we present a problem of scheduling on a single machine with the due dates and the total weighted tardiness cost minimization. We assume that processing times and due dates are deterministic or random variables with standard or Erlang distribution. Stochastic TWT problem will be briefly denoted by STWT. Literature describes certain variants of STWT problem with different distributions of random variables: Jang et al. [12] (normal distribution), Cai and Zhou [5] (exponential distribution), Li et al. [16] (Erlang distribution). Here we study the resistance to random parameter changes on solutions constructed according to the tabu search metaheuristics. We also present a certain measure (called stability) that allows one to evaluate the resistance of solutions to random data perturbations.

3. Problem description and enumeration scheme

To formulate the problem, we will use the following notation:

- J – set of jobs,
- p_i, w_i, d_i – processing time, weight and due date of a job $i \in J$,

- T_i – tardiness of a job i ,
- C_i – completion time of a job i ,
- π – permutation,
- $f(\pi)$ – cost function,
- Φ – set of all permutations of elements from J ,
- π^E, π^T – blocks,
- i_j^k, s_j^k – insert and swap moves,
- $M(\pi)$ – set of moves,
- $N(\pi)$ – neighborhood,
- f – density function,
- F – cumulative distribution.

Each schedule of jobs can be represented by permutation $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ of elements of the jobs set J . Let Φ denotes the set of all n -elementary permutations. For any permutation $\pi \in \Phi$, by $C_{\pi(i)} = \sum_{j=1}^i p_{\pi(j)}$ we denote the completion time of execution of the job $\pi(i)$. The total cost of the solution π is

$$f(\pi) = \sum_{i=1}^n w_{\pi(i)} T_{\pi(i)}, \quad (1)$$

where tardiness

$$T_{\pi(i)} = \max\{0, C_{\pi(i)} - d_{\pi(i)}\}. \quad (2)$$

The job $\pi(i)$ is considered *early*, if it is completed before its due date ($C_{\pi(i)} \leq d_{\pi(i)}$) and *tardy*, if the job is completed after its due date (i.e. $C_{\pi(i)} > d_{\pi(i)}$).

The total weighted tardiness problem consists in finding such a permutation $\pi^* \in \Phi$ which minimizes function f on the set Φ , i.e.

$$f(\pi^*) = \min\{f(\pi) : \pi \in \Phi\}.$$

3.1. Blocks of jobs in permutation. In every permutation $\pi \in \Phi$ there are such subsequences of jobs, that:

1. executing of every job from subsequence ends before its deadline (all of the jobs are not tardy), or
2. executing of every job from subsequence ends after its deadline (all of the jobs are tardy).

Such subsequences are called blocks.

Block of early jobs

We call a subsequence of jobs π^E from permutation $\pi \in \Phi$ an E-block if:

- a) every job $j \in \pi^E$ is early and $d_j \geq C_{last}$, where C_{last} is the time of finishing of executing of the last job from π^E ,
- b) π^E is maximal subsequence which fulfilling limitation (a).

It is easy to prove, that if π^E is E-block, that

$$\min\{j \in \pi^E : d_j\} \geq C_{last}.$$

So, in any permutation of jobs from π^E , every job is executed early in permutation π . Algorithm (AE-block) designating E-block is included in the work [2].

Block of tardy jobs

We call subsequence of jobs (occurring directly one after another) π^T from permutation $\pi \in \Phi$ a T-block, if:

- a') every job $j \in \pi^T$ is early and $d_j < C_{first} + p_j$, where C_{first} is the time of finishing executing of the first job in π^T ,
- b') π^T is maximal subsequence which fulfills limitation (a').

It is easy to proof, that in any permutation of jobs from π^T , every job is tardy in permutation π .

Considering sequentially π (starting from $\pi(1)$ and applying appropriate algorithm AE-block or AT-block (see [2]), we can partition π into E and T blocks. Complexity of partition procedure is $O(n)$.

Example 1. Let us consider the 10 jobs' instance that is specified in Table 1.

Table 1.
Data for the instance

i	1	2	3	4	5	6	7	8	9	10
p_i	2	3	1	2	3	2	3	3	2	4
d_i	12	19	12	9	5	1	17	24	19	3
w_i	3	1	2	5	3	3	4	2	4	5

Let $\pi = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$. Permutation π has three T-blocks: $T_1 = (1, 2, 3, 4)$, $T_2 = (7)$, $T_3 = (8)$ and two D-blocks: $D_1 = (5, 6)$, $D_2 = (9, 10)$. We can see these blocks on Fig. 1. The total cost $f(\pi) = 168$.

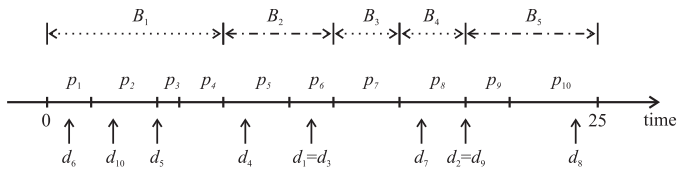


Fig. 1. Blocks in permutation $\pi = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$

With respect to T-blocks π^T in π , it should be noticed that by condition a' from Section 3.1, for any permutation of jobs within π^T , all the jobs are tardy. Therefore, an optimal sequence of the jobs within π^T of π can be obtained, using well-known weighted shortest processing time (WSPT) rule, proposed by Smith [24]. The WSPT rule creates an optimal sequence of the jobs in the non-increasing order of the ratios w_j/p_j .

Example 2. There are two D-blocks in permutation $\pi = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ from Example 1. Jobs does not fulfil WSPT rule in the block $D_1 = (5, 6)$, because $w_5/p_5 = 2/2 < w_6/p_6 = 3/2$. We are swapping jobs 5 and 6, therefore $D_1 = (6, 5)$. Relation is fulfilled in the block $D_2 = (9, 10)$: ($w_9/p_9 = 4/2 > w_{10}/p_{10} = 5/4$).

We are obtaining permutation $\beta = (1, 2, 3, 4, 6, 5, 7, 8, 9, 10)$. The total cost $f(\beta) = 158$. Optimal permutation $\pi^* = (6, 10, 4, 5, 3, 1, 7, 9, 2, 8)$, and $f(\pi^*) = 47$.

Fundamental block properties of the TWT problem are derived from the following Theorem.

Theorem 1. [2] Let $\pi \in \Phi$ be a permutation with blocks B_1, B_2, \dots, B_m , and let the jobs of each T-block of π be ordered according to the WSPT rule. If the permutation β has been obtained from π by an interchange of jobs that $f(\beta) < f(\pi)$, then in β , at least one job of some block of π was moved before the first or after the last job of this block.

Note that Theorem 1 provides the necessary condition to obtain a permutation β from π such that $f(\beta) < f(\pi)$.

3.2. Tabu search algorithm. Generally, in tabu search algorithm, for the given initial permutation, we identify blocks (if there is more than one partition of the permutation into blocks, any of them can be used), and re-order the jobs of each T-block according to the WSPT rule. Then, for the resulting (basic) permutation π , we calculate $f(\pi)$, create the set of moves, compound move v , and the permutation π_v . Next, the search process is repeated for the new basic permutation π_v until a given number of iterations is reached. According to the philosophy of tabu search, the compound move cannot contain single moves with a tabu status; these moves are not allowed.

Moves and neighborhoods

One of the main components of a local search algorithm is the definition of the move set that creates a neighbourhood. A move changes the location of some jobs in a given permutation. Among many types of moves considered in the literature, two of them appear prominently:

1. Insert move (i-move) i_l^k , consists in remove the job $\pi(k)$ from the position k and next insert it in a position l . Thus the move i_l^k generates a new permutation $i_l^k(\pi) = \pi_l^k$.
2. Swap move (s-move) s_l^k , in which the jobs if $\pi(k)$ and $\pi(l)$ are swapped among some positions k and l . The move s_l^k generates permutation $s_l^k(\pi)$.

Computational complexity of executing i-move is $O(n)$, and $O(1)$ of executing s-move.

Any move v (i-move or s-move) is improving, if it gives improvement of cost function, that means generates permutation π_v such, that $f(\pi_v) < f(\pi)$. Theorem 1 follows, that moves consist in changing the order of jobs in any block are not improving moves.

Let job $\pi(j)$ belong to some block in permutation π . Moves, which can improve value of cost function consist in moving job $\pi(j)$ before the first, or after the last job of this block. M_j^{bf} and M_j^{af} be sets of such moves (that means all of i-moves and s-moves), and

$$M(\pi) = \bigcup_{j=1}^n M_j^{bf} \cup \bigcup_{j=1}^n M_j^{af}. \tag{3}$$

The neighborhood of the π is a set of permutations

$$N(\pi) = \{ \tau(\pi) : \tau \in M(\pi) \}. \tag{4}$$

Much more efficient moves and neighbourhoods used in single machine algorithms are described in [3, 30].

To prevent from arising cycle too quickly (returning to the same permutation after some small number of iterations of the algorithm), some attributes of each move are saved on so-called tabu list (list of the forbidden moves).

The tabu moves list

To prevent from arising cycle some attributes of each movement are put in on the list of tabu moves. It is realized by means of FIFO queue. Performing a movement $m_j^r \in M(\pi)$ (i-move or s-move) on the tabu list L attributes of this move, i.e. the triple $(\pi(r), j, f(\pi_j^r))$ are put down.

Let us assume that we examine a move $m_l^k \in M(\beta)$ generating from $\beta \in \Phi$ a permutation β_l^k . If there is a triple (r, j, Ψ) on the list L such that $\beta(k) = r, l = j$ and $f(\beta_l^k) \leq \Psi$ then such a move is removed from the set $M(\beta)$.

4. Randomization

In practice, there are considerable difficulties in defining probability distributions of random variables that are parameters of the model. Especially when there is a lack of statistical data. Typically, in this case, we use expert knowledge that determines both distributions and their parameters. In this section we consider the stochastic single machine total weighted tardiness problem (STWT) where, processing times or tasks due dates are random variables with normal or Erlang distribution. The first distribution is most commonly used in modeling of “natural” randomness (eg. weather, demand, traffic flow, etc.), second – in turn – concerns machine breakdowns, absenteeism, employee errors, etc.

Let $\pi \in \Phi$ be a sequence of the tasks processing for STWT problem. If the execution times \tilde{p}_i or the expected jobs’ due dates $\tilde{d}_i (i \in J)$ are random variables, then tardiness $\tilde{T}_{\pi(i)}$ (equivalent 2) and the objective function (equivalent to 1)

$$\tilde{F}(\pi) = \sum_{i=1}^n w_i \tilde{T}_i. \tag{5}$$

are also random variables.

In the algorithms for solving optimization problems objective function values for different solution are compared. In case when the function is random variable (5) we will use some of its moments or its combinations. Initially performed computational experiments have shown that the best results were obtained when the expected solution and standard deviation were applied to compare the solutions. Therefore, as the comparative criteria for solutions there will be two functions applied:

$$W_1(\pi) = E(\tilde{f}(\pi)) = \sum_{i=1}^n w_{\pi(i)} E(\tilde{T}_{\pi(i)}), \tag{6}$$

$$\begin{aligned} W_2(\pi) &= E(\tilde{f}(\pi)) + \delta \cdot D(\tilde{f}(\pi)) = \\ &= \sum_{i=1}^n \left(w_{\pi(i)} E(\tilde{T}_{\pi(i)}) + \delta \cdot D(\tilde{T}_{\pi(i)}) \right). \end{aligned} \tag{7}$$

Parameter $\delta (0 < \delta \leq 1)$ is determined experimentally.

If X is a random variable, then by F_X and f_X we denote respectively its cumulative distribution and density function.

for considered in the work distributions of random variables (normal or Erlang’s), the density function is equal to the derivative of the cumulative distribution function, i.e. $f_X(x) = F_X'(x)$. Since the variance

$$D^2(X) = E(X^2) - (E(X))^2, \tag{8}$$

in order to calculate the standard deviation one should determine only expected values of random variables X and X^2 .

When calculating the value of the function W_1 and W_2 we will use the following property.

Corollary 1. If the tasks’ processing times $\tilde{p}_i \in J$ are independent random variables, then the deadlines of tasks execution $\tilde{C}_i = \tilde{p}_1 + \tilde{p}_2 + \dots + \tilde{p}_i$ are also random variables but if the variables $\tilde{p}_k (k = 1, 2, \dots, i)$ have the distribution:

(a) normal $\tilde{p}_k \sim N(p_k, \sigma_k)$, then

$$\tilde{C}_i \sim N \left(\sum_{j=1}^i p_j, \sqrt{\sum_{j=1}^i \sigma_j^2} \right). \tag{9}$$

(b) Erlang $\tilde{p}_k \sim E(\alpha_k, \lambda)$, then

$$\tilde{C}_i \sim E \left(\sum_{j=1}^i \alpha_j, \lambda \right). \tag{10}$$

Let $(p_i, d_i, w_i)_{i=1,2,\dots,n}$ be an example of deterministic data for TWT problem. In order to simplify the notation we assume that the current solution to the problem is a natural permutation, i.e. $\pi = (1, 2, \dots, n)$. In this section we consider the stochastic variants of the problem in which some tasks’ parameters (processing times or due dates) are random variables:

- A. processing times $\tilde{p}_i \sim N(p_i, a \cdot p_i)$, deterministic: w_i and d_i ,
- B. due dates $\tilde{d}_i \sim N(d_i, c \cdot d_i)$, deterministic: p_i and w_i ,
- C. processing times $\tilde{p}_i \sim E(\alpha_i, \lambda)$, deterministic: w_i and d_i ,
- D. due dates $\tilde{d}_i \sim E(\beta_i, \mu)$, deterministic: p_i and w_i ,

where $N(\mu_i, c \cdot \mu_i)$ is a normal distribution of average μ_i and standard deviation $c \cdot \mu_i$ (parameter c is determined experimentally). For Erlang distributions E parameter $\alpha_i = p_i \lambda, \beta_i = d_i \mu$, where $\lambda = \max\{2(\min\{p_i: 1 \leq i \leq n\})^{-1}, 1\}$ and $\mu = \max\{2(\min\{d_i: 1 \leq i \leq n\})^{-1}, 1\}$. It is easy to verify that for both distributions the expected value $E\tilde{p}_i = p_i$. Individual cases will be briefly put down using a modification of the notation of Graham. For each of them we will give formulas enabling the calculation of the function W_1 and W_2 . From the equality (8) we can see that the process comes down to the calculation of values of expected variables $E(\tilde{T}_i)$ or $E(\tilde{T}_i^2)$.

Case A. (problem: $1 | \tilde{p}_i \sim N(p_i, a \cdot p_i) | \sum w_i T_i$)

The times of jobs execution $\tilde{p}_i \sim N(p_i, a \cdot p_i) (i \in J)$ are independent random variables with normal distribution. It is easy

to notice that the times of jobs completion are random variables with normal distribution (see. Corollary 1), that is:

$$\tilde{C}_i \sim N \left(p_1 + p_2 \dots + p_i, a \cdot \sqrt{p_1^2 + \dots + p_i^2} \right), \quad (11)$$

and tardiness

$$\tilde{T}_i = \begin{cases} \tilde{C}_i - d_i, & \text{if } \tilde{C}_i > d_i, \\ 0, & \text{if } \tilde{C}_i \leq d_i. \end{cases} \quad (12)$$

Lemma 1. Cumulative distribution of tardiness, i.e. of random variable distribution \tilde{T}_i , $i \in J$

$$F_{\tilde{T}_i}(x) = \begin{cases} F_{\tilde{C}_i}(d_i + x) - F_{\tilde{C}_i}(d_i)F_{\tilde{C}_i}(d_i) + F_{\tilde{C}_i}(d_i), & \text{if } x > 0, \\ 0, & \text{if } x \leq 0, \end{cases} \quad (13)$$

and density function

$$f_{\tilde{T}_i}(x) = \begin{cases} f_{\tilde{C}_i}(d_i + x) - F_{\tilde{C}_i}(d_i)f_{\tilde{C}_i}(d_i + x), & \text{if } x > 0, \\ 0, & \text{if } x \leq 0, \end{cases} \quad (14)$$

where $f_{\tilde{C}_i}(x)$ is density of random variable \tilde{C}_i .

Proof. We consider two cases:

1. $x > 0$, then

$$F_{\tilde{T}_i}(x) = P(\tilde{T}_i < x) = P(\tilde{T}_i < x | \tilde{C}_i - d_i > 0)P(\tilde{C}_i - d_i > 0) + P(\tilde{T}_i < x | \tilde{C}_i - d_i \leq 0)P(\tilde{C}_i - d_i \leq 0).$$

We note that

$$\text{and } \begin{aligned} P(\tilde{T}_i < x | \tilde{C}_i - d_i \leq 0) &= 1 \\ P(\tilde{T}_i < x | \tilde{C}_i - d_i > 0) &= P(\tilde{C}_i - d_i < x). \end{aligned}$$

Thus

$$\begin{aligned} F_{\tilde{T}_i}(x) &= P(\tilde{C}_i - d_i < x)P(\tilde{C}_i - d_i > 0) + P(\tilde{C}_i - d_i \leq 0) \\ &= F_{\tilde{C}_i}(d_i + x)(1 - F_{\tilde{C}_i}(d_i)) + F_{\tilde{C}_i}(d_i) \\ &= F_{\tilde{C}_i}(d_i + x) - F_{\tilde{C}_i}(d_i + x)F_{\tilde{C}_i}(d_i) + F_{\tilde{C}_i}(d_i). \end{aligned}$$

2. $x \leq 0$. From definition of the cumulative distribution function results directly that $f_{\tilde{T}_i}(x) = 0$ for $x \leq 0$.

To prove the formula (14) on the density function $f_{\tilde{T}_i}(x)$ we should only calculate the derivative from cumulative distribution (13). \square

To simplify the notation we assume that

$$\mu = p_1 + \dots + p_i, \text{ and } \sigma = a\sqrt{p_1^2 + \dots + p_i^2}. \quad (15)$$

In this case, the density function of the deadline for tasks completion $i \in J$

$$f_{\tilde{C}_i}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (16)$$

Here we will prove an auxiliary lemma, which we use in calculating the expected value of random variables \tilde{T}_i and \tilde{T}_i^2 .

Lemma 2. If \tilde{C}_i , $i = 1, 2, \dots, n$ is a random variable specifying the date of completion of the task (11), then

$$\begin{aligned} \int_{d_i}^{\infty} y f_{\tilde{C}_i}(y) dy &= \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}} + \\ &+ \mu \left(1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right) \right) \end{aligned} \quad (17)$$

and

$$\int_{d_i}^{\infty} f_{\tilde{C}_i}(y) dy = 1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right). \quad (18)$$

Proof. From the definition of the random variable \tilde{C}_i and (15) and (16) follows that

$$\int_{d_i}^{\infty} y f_{\tilde{C}_i}(y) dy = \int_{d_i}^{\infty} y \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} dy.$$

We introduce substitutions $z = \frac{y-\mu}{\sigma}$, thus $dy = \sigma dz$. By assuming $y = d_i$ we obtain $z = \frac{d_i-\mu}{\sigma}$ and $y = \sigma z + \mu$. Then

$$\begin{aligned} \int_{d_i}^{\infty} y f_{\tilde{C}_i}(y) dy &= \int_{\frac{d_i-\mu}{\sigma}}^{\infty} (\sigma z + \mu) \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz = \\ &= \sigma \int_{\frac{d_i-\mu}{\sigma}}^{\infty} z \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz + \mu \int_{\frac{d_i-\mu}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz. \end{aligned}$$

We notice that

$$(e^{-\frac{z^2}{2}})' = -ze^{-\frac{z^2}{2}}, \text{ there } \int ze^{-\frac{z^2}{2}} dz = -e^{-\frac{z^2}{2}}$$

Therefore

$$\begin{aligned} \int_{d_i}^{\infty} y f_{\tilde{C}_i}(y) dy &= \frac{\sigma}{\sqrt{2\pi}} \left[-e^{-\frac{z^2}{2}} \right]_{\frac{d_i-\mu}{\sigma}}^{\infty} + \\ &+ \mu \left(1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right) \right) = \\ &= \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}} + \mu \left(1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right) \right). \end{aligned}$$

We have proved the first integral of the thesis. Now we can move to proof of equality (18). Using the designation (15), the

density function of the random variable \tilde{C}_i and substitution that we used in proof (17) we obtain

$$\begin{aligned} \int_{d_i}^{\infty} f_{\tilde{C}_i}(y)dy &= \int_{d_i}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} dy = \\ &= \int_{\frac{d_i-\mu}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz = \left(1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right)\right). \quad \square \end{aligned}$$

Theorem 2. If the task completion times are independent random variables normally distributed $\tilde{p}_k \sim N(p_k, a \cdot p_k) (k = 1, 2, \dots, n)$, then the expected value of tardiness (12) of task $i \in J$ is

$$\begin{aligned} E(\tilde{T}_i) &= (1 - F_{\tilde{C}_i}(d_i)) \left(\frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}} + \right. \\ &\left. + (\mu - d_i) \left(1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right)\right) \right). \end{aligned}$$

Proof. By definition of the expected value (11) and (12) it follows that

$$\begin{aligned} E(\tilde{T}_i) &= \int_{-\infty}^{\infty} x f_{\tilde{T}_i}(x) dx = \int_0^{\infty} x \left(f_{\tilde{C}_i}(d_i+x) - \right. \\ &\left. - F_{\tilde{C}_i}(d_i) f_{\tilde{C}_i}(d_i+x) \right) dx = \\ &= (1 - F_{\tilde{C}_i}(d_i)) \int_0^{\infty} x f_{\tilde{C}_i}(d_i+x) dx. \end{aligned}$$

We introduce a substitution $y = d_i + x$.

$$\begin{aligned} (1 - F_{\tilde{C}_i}(d_i)) \int_0^{\infty} x f_{\tilde{C}_i}(d_i+x) dx &= \\ &= (1 - F_{\tilde{C}_i}(d_i)) \int_{d_i}^{\infty} (y - d_i) f_{\tilde{C}_i}(y) dy = \\ &= (1 - F_{\tilde{C}_i}(d_i)) \left(\int_{d_i}^{\infty} y f_{\tilde{C}_i}(y) dy - d_i \int_{d_i}^{\infty} f_{\tilde{C}_i}(y) dy \right). \end{aligned}$$

Using Lemma 2 and performing simple transformations we finally obtain thesis of the theorem. \square

Below we prove the lemma, which will be used to calculate the expected value of a random variable \tilde{T}_i^2 .

Lemma 3. If \tilde{C}_i is the time (11) of task completion $i \in J$, then

$$\begin{aligned} \int_{d_i}^{\infty} y^2 f_{\tilde{C}_i}(y) dy &= (\sigma^2 + \mu^2) \left(1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right)\right) + \\ &+ \frac{2\sigma^2\mu + d_i - \mu}{\sqrt{2\pi}\sigma} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}}. \end{aligned}$$

Proof. From the definition of the distribution function of the random variable \tilde{C}_i , (15) and (16) it follows that

$$\int_{d_i}^{\infty} y^2 f_{\tilde{C}_i}(y) dy = \int_{d_i}^{\infty} y^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} dy.$$

Introducing, the same as in the proof of Lemma 2 substitutions, we get

$$\begin{aligned} \int_{d_i}^{\infty} y^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} dy &= \int_{\frac{d_i-\mu}{\sigma}}^{\infty} (\sigma z + \mu)^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz = \\ &= \sigma^2 \int_{\frac{d_i-\mu}{\sigma}}^{\infty} z^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz + \\ &+ 2\sigma\mu \int_{\frac{d_i-\mu}{\sigma}}^{\infty} z \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz + \mu^2 \int_{\frac{d_i-\mu}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz. \end{aligned} \quad (19)$$

Now we can individually calculate each of the above integrals.

The first integral. Since

$$(-ze^{-\frac{z^2}{2}})' = -1 \cdot e^{-\frac{z^2}{2}} + (-z)e^{-\frac{z^2}{2}} \cdot (-z) = z^2 e^{-\frac{z^2}{2}} - e^{-\frac{z^2}{2}},$$

then

$$\int z^2 e^{-\frac{z^2}{2}} - e^{-\frac{z^2}{2}} dz = \int z^2 e^{-\frac{z^2}{2}} dz - \int e^{-\frac{z^2}{2}} dz = -ze^{-\frac{z^2}{2}}.$$

Hence the definite integral

$$\int_{\frac{d_i-\mu}{\sigma}}^{\infty} z^2 e^{-\frac{z^2}{2}} dz - \int_{\frac{d_i-\mu}{\sigma}}^{\infty} e^{-z^2} 2dz = \left[-ze^{-\frac{z^2}{2}} \right]_{\frac{d_i-\mu}{\sigma}}^{\infty}.$$

Therefore

$$\begin{aligned} \int_{\frac{d_i-\mu}{\sigma}}^{\infty} z^2 e^{-\frac{z^2}{2}} dz - \sqrt{2\pi} \int_{\frac{d_i-\mu}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz &= \\ &= \lim_{z \rightarrow \infty} (-ze^{-\frac{z^2}{2}}) + \frac{d_i - \mu}{\sigma} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}}. \end{aligned}$$

After simple transformations we obtain the equality

$$\begin{aligned} \int_{\frac{d_i-\mu}{\sigma}}^{\infty} z^2 e^{-\frac{z^2}{2}} dz &= \sqrt{2\pi} \left(1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right)\right) + \\ &+ \frac{d_i - \mu}{\sigma} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}}. \end{aligned} \quad (20)$$

The second integral.

Using the equality $\int ze^{-\frac{z^2}{2}} dz = -e^{-\frac{z^2}{2}}$, we get

$$\begin{aligned} \int_{\frac{d_i-\mu}{\sigma}}^{\infty} z \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz &= \left[\frac{1}{\sqrt{2\pi}} (-e^{-\frac{z^2}{2}}) \right]_{\frac{d_i-\mu}{\sigma}}^{\infty} = \\ &= \frac{1}{\sqrt{2\pi}} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}}. \end{aligned} \quad (21)$$

Finally, the third integral is equal to

$$\int_{\frac{d_i-\mu}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz = 1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right). \quad (22)$$

We return to the formulation (19). Using (20, 21) and (22), after simple transformations, we ultimately obtain the thesis of the lemma

$$\begin{aligned} \int_{d_i}^{\infty} y^2 f_{\tilde{C}_i}(y) dy &= \frac{\sigma^2}{\sqrt{2\pi}} \int_{\frac{d_i-\mu}{\sigma}}^{\infty} z^2 e^{-\frac{z^2}{2}} dz \\ + 2\sigma\mu \int_{\frac{d_i-\mu}{\sigma}}^{\infty} z \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz + \mu^2 \int_{\frac{d_i-\mu}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz &= \\ = (\sigma^2 + \mu^2) \left(1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right)\right) + & \\ + \frac{2\sigma^2\mu + d_i - \mu}{\sqrt{2\pi\sigma}} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}}. & \quad \square \end{aligned}$$

Theorem 3. If the task completion times are independent random variables normally distributed $\tilde{p}_k \sim N(p_k, a \cdot p_k)$ ($k = 1, 2, \dots, n$), then the expected value of a random variable \tilde{T}_i^2

$$\begin{aligned} E(\tilde{T}_i^2) &= (\sigma^2 + \mu^2) \left(1 - F_{N(0,1)}\left(\frac{d_i-\mu}{\sigma}\right)\right) + \\ &+ \frac{2\sigma^2\mu + d_i - \mu}{\sqrt{2\pi\sigma}} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}}. \end{aligned}$$

Proof. From the definition of the expected value

$$\begin{aligned} E(\tilde{T}_i^2) &= \int_{-\infty}^{\infty} x^2 f_{\tilde{T}_i}(x) dx = \\ &= \int_0^{\infty} x^2 f_{\tilde{C}_i}(d_i+x) dx - F_{\tilde{C}_i}(d_i) \int_0^{\infty} x^2 f_{\tilde{C}_i}(d_i+x) dx = \\ &= (1 - F_{\tilde{C}_i}(d_i)) \int_0^{\infty} x^2 f_{\tilde{C}_i}(d_i+x) dx. \end{aligned}$$

It remains to calculate $\int_0^{\infty} x^2 f_{\tilde{C}_i}(d_i+x) dx$. By introducing a substitution $y = d_i + x$ we get

$$\begin{aligned} \int_0^{\infty} x^2 f_{\tilde{C}_i}(d_i+x) dx &= \int_{d_i}^{\infty} (y-d_i)^2 f_{\tilde{C}_i}(y) dy = \\ &= \int_{d_i}^{\infty} y^2 f_{\tilde{C}_i}(y) dy - 2d_i \int_{d_i}^{\infty} y f_{\tilde{C}_i}(y) dy + d_i^2 \int_{d_i}^{\infty} f_{\tilde{C}_i}(y) dy. \end{aligned}$$

Therefore

$$\begin{aligned} (1 - F_{\tilde{C}_i}(d_i)) \int_0^{\infty} x^2 f_{\tilde{C}_i}(d_i+x) dx &= (1 - F_{\tilde{C}_i}(d_i)) \\ \left(\int_{d_i}^{\infty} y^2 f_{\tilde{C}_i}(y) dy \right) - 2d_i \int_{d_i}^{\infty} y f_{\tilde{C}_i}(y) dy &+ d_i^2 \int_{d_i}^{\infty} f_{\tilde{C}_i}(y) dy. \end{aligned}$$

Using Lemma 3 we finally get the conclusion of Theorem. \square

If the times of the tasks execution are independent random variables with a normal distribution ($\tilde{p}_i \sim N(p_i, a \cdot p_i)$, $i \in J$), then when calculating the value of the function W_1 and W_2 we will use Theorem 4 and 5.

Case B. (problem: $1|\tilde{d}_i \sim N(d_i, c \cdot d_i)|\sum w_i T_i$)

We assume that deadlines of tasks completion are independent random variables normally distributed ($\tilde{d}_i \sim N(d_i, c \cdot d_i)$, $i \in J$). Tardiness of task $i \in J$

$$\tilde{T}_i = \begin{cases} C_i - \tilde{d}_i, & \text{if } C_i > \tilde{d}_i, \\ 0, & \text{if } C_i \leq \tilde{d}_i, \end{cases} \quad (23)$$

is also a random variable.

Lemma 4. Cumulative distribution of tardiness

$$\begin{cases} 1 - F_{\tilde{d}_i}(C_i - x) F_{\tilde{d}_i}(C_i), & \text{if } x > 0, \\ 0, & \text{if } x \leq 0, \end{cases} \quad (24)$$

and density function

$$f_{\tilde{T}_i}(x) = \begin{cases} F_{\tilde{d}_i}(C_i) f_{\tilde{d}_i}(C_i - x), & \text{if } x > 0, \\ 0, & \text{if } x \leq 0, \end{cases} \quad (25)$$

$i = 1, 2, \dots, n$.

Proof. The proof is similar as Lemma 1. \square

Similarly as in Case A we will now prove two theorems allowing us to calculate the expected values of random variables \tilde{T}_i and \tilde{T}_i^2 .

Theorem 4. If the expected deadlines for the completion of tasks are independent random variables normally distributed $\tilde{d}_i \sim N(d_i, c \cdot d_i)$, then the expected value of tardiness (23)

$$\begin{aligned} E(\tilde{T}_i) &= F_{N(0,1)}\left(\frac{C_i-\mu}{\sigma}\right) \left(C_i F_{N(0,1)}\left(\frac{C_i-\mu}{\sigma}\right) + \right. \\ &\left. + \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(C_i-\mu)^2}{2\sigma^2}} \right) - \mu F_{N(0,1)}\left(\frac{C_i-\mu}{\sigma}\right) \end{aligned}$$

Proof. Similarly, as in the case of a proof of theorem 2 should be this time the use of lemma 2. \square

Thus, it remains to compute the expected value of the variable $E(\tilde{T}_i^2)$.

Theorem 5. If the deadlines of tasks are independent random variables normally distributed $\tilde{d}_i \sim N(d_i, c \cdot d_i)$, then the expected value of a random variable \tilde{T}_i^2

$$F_{N(0,1)}(\vartheta_i) \left(\frac{\sigma^2}{\sqrt{2\pi}} \vartheta_i e^{-\frac{\vartheta_i^2}{2}} - \left(\frac{\sigma^2}{\sqrt{2\pi}} + \frac{2\sigma\mu}{\sqrt{2\pi}} \right) e^{-\frac{\vartheta_i^2}{2}} + \mu^2 F_{N(0,1)}(\vartheta_i) + 2C_i \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{\vartheta_i^2}{2}} - 2C_i \mu F_{N(0,1)}(\vartheta_i) + C_i^2 F_{N(0,1)}(\vartheta_i) \right),$$

where $\vartheta_i = \frac{C_i - \mu}{\sigma}$.

Proof. Similarly, as in the theorem 3 (use of lemma 3). \square

In this case, to compute the value of the function W_1 and W_2 we will use Theorem 2 and 3.

Case C. (problem: $\tilde{p}_i \sim E(\alpha_i, \lambda)$)

We assume that the times of the tasks executions $p_i, i \in J$ are independent random variables with Erlang distribution, $\tilde{p}_i \sim E(\alpha_i, \lambda)$. Then, the completion date of task $i \in J$ is a random variable with Erlang distribution (Corollary 1)

$$\tilde{C}_i \sim \sum_{j=1}^i \tilde{p}_j \sim E(\alpha, \lambda),$$

where $\alpha = \alpha_1 + \alpha_2 \dots + \alpha_i$.

From the definition of the Erlang distribution, the density function of the random variable $\tilde{C}_i \sim E(\alpha, \lambda)$

$$f_{\tilde{C}_i}(x) = \begin{cases} \frac{1}{(\alpha-1)!} \lambda^\alpha x^{\alpha-1} e^{-\lambda x}, & \text{if } x > 0, \\ 0, & \text{if } x \leq 0. \end{cases} \quad (26)$$

In such a case tardiness

$$\tilde{T}_i = \begin{cases} \tilde{C}_i - d_i, & \text{if } \tilde{C}_i > d_i, \\ 0, & \text{if } \tilde{C}_i \leq d_i. \end{cases} \quad (27)$$

Lemma 5. The cumulative distribution function of the random variable \tilde{T}_i (27)

$$F_{\tilde{T}_i}(x) = \begin{cases} F_{\tilde{C}_i}(d_i) + (1 - F_{\tilde{C}_i}(d_i)) F_{\tilde{C}_i}(x + d_i), & \text{if } x > 0, \\ 0, & \text{if } x \leq 0, \end{cases} \quad (28)$$

and density function

$$f_{\tilde{T}_i}(x) = \begin{cases} (1 - F_{\tilde{C}_i}(d_i)) f_{\tilde{C}_i}(x + d_i), & \text{if } x > 0, \\ 0, & \text{if } x \leq 0. \end{cases} \quad (29)$$

Proof. The proof is omitted because it is similar to the proof of Lemma 1. \square

Now we prove an auxiliary lemma, which we use at calculating the expected value of variables \tilde{T}_i and \tilde{T}_i^2 .

Lemma 6. If the deadlines for the completion of tasks (27) have the Erlang distribution $\tilde{C}_i \sim E(\alpha, \lambda)$, then

$$\int_x^\infty x f_{\tilde{C}_i}(x) dx = \frac{\alpha}{\lambda} (1 - F_{E(\alpha+1, \lambda)}(a)), \quad (30)$$

and

$$\int_x^\infty x^2 f_{\tilde{C}_i}(x) dx = \frac{\alpha(\alpha+1)}{\lambda^2} (1 - F_{E(\alpha+2, \lambda)}(a)). \quad (31)$$

Proof. Using the density function (26) of random variable \tilde{C}_i

$$\begin{aligned} \int_a^\infty x f_{\tilde{C}_i}(x) dx &= \int_a^\infty x \frac{1}{(\alpha-1)!} \lambda^\alpha x^{\alpha-1} e^{-\lambda x} dx = \\ &= \frac{\alpha}{\lambda} \int_a^\infty \frac{1}{((\alpha+1)-1)!} \lambda^{\alpha+1} x^{(\alpha+1)} e^{-\lambda x} dx. \end{aligned}$$

Since

$$\frac{1}{((\alpha+1)-1)!} \lambda^{\alpha+1} x^{(\alpha+1)} e^{-\lambda x} = f_{E(\alpha+1, \lambda)}(x), \quad (32)$$

thus

$$\begin{aligned} \frac{\alpha}{\lambda} \int_a^\infty \frac{1}{((\alpha+1)-1)!} \lambda^{\alpha+1} x^{(\alpha+1)} e^{-\lambda x} dx &= \\ &= \frac{\alpha}{\lambda} (1 - F_{E(\alpha+1, \lambda)}(a)). \end{aligned}$$

Now we prove the second equality thesis of the lemma.

$$\begin{aligned} \int_{d_i}^\infty x^2 f_{\tilde{C}_i}(x) dx &= \int_{d_i}^\infty x^2 \frac{1}{(\alpha-1)!} \lambda^\alpha x^{\alpha-1} e^{-\lambda x} dx = \\ &= \frac{\alpha(\alpha+1)}{\lambda^2} (1 - F_{E(\alpha+2, \lambda)}(d_i)). \end{aligned}$$

The last equality used (32). \square

Theorem 6. If the times of the tasks execution are independent random variables of the Erlang distribution $E(\alpha_i, \lambda), i \in J$, then the expected value of tasks tardiness is

$$\begin{aligned} E(\tilde{T}_i) &= (1 - F_{E(\alpha, \lambda)}(d_i)) \left(\frac{\alpha}{\lambda} (1 - F_{E(\alpha+1, \lambda)}(d_i)) - \right. \\ &\quad \left. - d_i (1 - F_{E(\alpha, \lambda)}(d_i)) \right). \end{aligned}$$

Proof. The expected value

$$E(\tilde{T}_i) = \int_0^\infty x f_{\tilde{T}_i}(x) dx = (1 - F_{\tilde{C}_i}(d_i)) \int_0^\infty f_{\tilde{C}_i}(x + d_i) dx. \quad (33)$$

We calculate the integral $\int_0^\infty f_{\tilde{C}_i}(x + d_i) dx$. Using substitution $u = x + d_i$ and complying with (30) we obtain

$$\begin{aligned} \int_0^\infty f_{\tilde{C}_i}(x+d_i)dx &= \int_0^\infty f_{\tilde{C}_i}(x+d_i)dx = \\ &= \int_{d_i}^\infty (u-d_i)f_{\tilde{C}_i}(u)du - d_i \int_{d_i}^\infty f_{\tilde{C}_i}(u)du = \\ &= \frac{\alpha}{\lambda} (1 - F_{E(\alpha+1,\lambda)}(d_i)) - d_i (1 - F_{E(\alpha,\lambda)}(d_i)). \end{aligned}$$

Substituting (33) into the above equality we obtain the assertion theorem. \square

We calculate the expected value of the variable \tilde{T}_i^2 .

Theorem 7. If the task completion times are independent random variables with erlang distribution $E(\alpha_i, \lambda)$, $i \in J$, then the expected value of tardiness (27) of task $i \in J$ is

$$\begin{aligned} E(\tilde{T}_i^2) &= (1 - F_{E(\alpha,\lambda)}(d_i)) \left(\frac{\alpha(\alpha+1)}{\lambda^2} \right. \\ & (1 - F_{E(\alpha+2,\lambda)}(d_i)) - 2d_i \frac{\alpha}{\lambda} (1 - F_{E(\alpha+1,\lambda)}(d_i)) + \\ & \left. + d_i^2 (1 - F_{E(\alpha,\lambda)}(d_i)) \right). \end{aligned} \quad (34)$$

Proof. By the definition of the expected value

$$\begin{aligned} E(\tilde{T}_i^2) &= \int_{-\infty}^\infty x^2 f_{\tilde{T}_i}(x)dx = \int_0^\infty x^2 (P(\tilde{C}_i > d_i) f_{\tilde{C}_i}(x+d_i))dx \\ & P(\tilde{C}_i > d_i) \int_0^\infty x^2 f_{\tilde{C}_i}(x+d_i)dx. \end{aligned}$$

Similarly, as in the proof of Theorem 6 by performing substitution $u = x + d_i$ we obtain

$$\begin{aligned} E(\tilde{T}_i^2) &= P(\tilde{C}_i > d_i) \left(\int_{d_i}^\infty u^2 f_{\tilde{C}_i}(u)du - \right. \\ & \left. - 2d_i \int_{d_i}^\infty u f_{\tilde{C}_i}(u)du + d_i^2 \int_{d_i}^\infty f_{\tilde{C}_i}(u)du \right). \end{aligned}$$

The first two integrals are calculated using the equality (30) and (31). Then, after simple transformations, we obtain the assertion (34). \square

When tasks completion times are random variables with Erlang distribution, values of the function W_1 and W_2 are calculated using the Theorem 6 and 7.

Case D. (problem: $1|\tilde{d}_i \sim E(\beta_i, \chi)|\sum w_i T_i$)

We assume that the required deadlines for the completion of tasks are independent random variables with Erlang distribution $\tilde{d}_i \sim E(\beta_i, \mu)$, where $\beta_i = d_i \mu$, $\mu = \max\{2(\min\{d_i: 1 \leq i \leq n\})^{-1}, 1\}$. Similarly, as for case B, we determine the cumulative distribution function of the density of tasks tardiness $i \in J$

$$\tilde{T}_i = \begin{cases} C_i - \tilde{d}_i, & \text{if } C_i > \tilde{d}_i, \\ 0, & \text{if } C_i \leq \tilde{d}_i, \end{cases} \quad (35)$$

where the date of tasks completion is $C_i = \sum_{j=1}^i p_j$.

Lemma 7. Cumulative distribution of tasks tardiness (35)

$$F_{\tilde{T}_i}(x) = \begin{cases} 1 - F_{\tilde{d}_i}(C_i - x)F_{\tilde{d}_i}(C_i) & \text{if } x > 0, \\ 0, & \text{if } x \leq 0, \end{cases} \quad (36)$$

and density function

$$f_{\tilde{T}_i}(x) = \begin{cases} F_{\tilde{d}_i}(C_i)f_{\tilde{d}_i}(C_i - x), & \text{if } x > 0, \\ 0, & \text{if } x > 0. \end{cases} \quad (37)$$

Proof. We omit the proof because it is similar to the proof of Lemma 5. \square

Then, we can proceed to calculate the expected values of random variables \tilde{T}_i and \tilde{T}_i^2 .

Theorem 8. If the requested deadlines for the completion of tasks are independent random variables with Erlang distribution $\tilde{d}_i \sim E(\beta_i, \mu)$, $i = 1, 2, \dots, n$, then the expected value of tardiness (35) is

$$\begin{aligned} E(\tilde{T}_i) &= F_{E(\beta,\mu)}(C_i)(C_i F_{E(\beta,\mu)}(C_i) - \\ & - \frac{\beta}{\mu} F_{E(\beta+1,\mu)}(C_i)), \end{aligned} \quad (38)$$

where $\beta = \beta_1 + \beta_2 + \dots + \beta_n$.

Proof. Using the density function of the tardiness of random variable \tilde{T}_i (Lemat 7) the expected value

$$\begin{aligned} E(\tilde{T}_i) &= \int_0^\infty x F_{\tilde{d}_i}(C_i) f_{\tilde{d}_i}(C_i - x)dx = \\ & = F_{\tilde{d}_i}(C_i) \int_0^\infty x f_{\tilde{d}_i}(C_i - x)dx. \end{aligned}$$

The next part is similar to the proof of 4 \square

Theorem 9. If the expected deadlines for the completion of tasks are independent random variables with Erlang distribution $\tilde{d}_i \sim E(\beta_i, \mu)$, the expected value of the square tardiness is

$$\begin{aligned} E(\tilde{T}_i^2) &= F_{E(\alpha,\lambda)}(C_i) \left(\frac{\alpha(\alpha+1)}{\lambda^2} F_{E(\alpha+2,\lambda)}(C_i) \right. \\ & \left. - 2C_i \frac{\alpha}{\lambda} F_{E(\alpha+1,\lambda)}(C_i) + C_i^2 F_{E(\alpha,\lambda)}(C_i) \right). \end{aligned} \quad (39)$$

Proof. The expected value

$$\begin{aligned} E(\tilde{T}_i^2) &= \int_0^\infty x^2 F_{\tilde{d}_i}(C_i) f_{\tilde{d}_i}(C_i - x)dx = \\ & = F_{\tilde{d}_i}(C_i) \int_0^\infty x^2 f_{\tilde{d}_i}(C_i - x)dx. \end{aligned} \quad (40)$$

In carrying out the substitution, as in the proof of Theorem 4 and using Lemma 6 (equality (30) and (31)) we finally obtain

$$\begin{aligned}
 E(\tilde{T}_i^2) &= F_{\tilde{d}_i}(C_i) \int_{-\infty}^{C_i} (y^2 - 2C_i y + C_i^2) f_{\tilde{d}_i}(y) dy = \\
 &= F_{\tilde{d}_i}(C_i) \left(\int_{-\infty}^{C_i} y^2 f_{\tilde{d}_i}(y) dy - \right. \\
 &\quad \left. - 2C_i \int_{-\infty}^{C_i} y f_{\tilde{d}_i}(y) dy + C_i^2 \int_{-\infty}^{C_i} f_{\tilde{d}_i}(y) dy \right). \tag{41}
 \end{aligned}$$

In this way we have proved the thesis statement. \square
 Proved equalities (Theorem 8 and 9) will be used when calculating the value of function W_1 oraz W_2 .

5. The algorithms' stability

In this section we shall introduce a certain measure which let us examine the influence of the change of jobs' parameters on the goal function value (1) i.e. the solution stability.

Let $\delta = ((p_1, u_1, w_1, e_1, d_1), \dots, (p_n, u_n, w_n, e_n, d_n))$ be an example of deterministic data for the TWT problem. By $D(\delta)$ we denote a set of data generated from δ by a disturbance of jobs parameters. A disturbance consists in changing these times on random determined values.

Let $A = \{AD, AP\}$ where AD and AP is the deterministic and the probabilistic algorithm, respectively (i.e. solving examples with deterministic or random times of jobs' performance) for the TWT problem. By π_δ we denote a solution (a permutation) determined by the algorithm A for a data δ . Then, let $f(A, \pi_\delta, \varphi)$ be the cost of jobs' execution (1) for the example φ in a sequence determined by a solution (a permutation) π_δ determined by the algorithm A for data δ . Then,

$$\Delta(A, \delta, D(\delta)) = \frac{1}{|D(\delta)|} \sum_{\varphi \in D(\delta)} \frac{f(A, \pi_\delta, \varphi) - f(AD, \pi_\varphi, \varphi)}{f(AD, \pi_\varphi, \varphi)},$$

is called the solution stability π_δ (of an example δ) determined by the algorithm A on the set of disturbed data $D(\delta)$.

Let Ω be a set of deterministic examples for the problem of jobs' arrangement. The stability rate of the algorithm A on the set Ω is defined in the following way:

$$S(A, \Omega) = \frac{1}{|\Omega|} \sum_{\delta \in \Omega} \Delta(A, \delta, D(\delta)) \cdot 100\%. \tag{42}$$

In the following section we will present numerical experiments that allow comparisons of the deterministic stability coefficient $S(AD, \Omega)$ with the probabilistic stability coefficient $S(AP, \Omega)$.

6. Computational experiments

Presented in Chapter 3.2 tabu search algorithm was adopted to solve the deterministic TWT problem (i.e. $1 \parallel \sum w_i T_i$) and its variants with random parameters. In short, the deterministic algorithm will be denoted by AD , and the stochastic one by AP ,

whereas by AP_1 we denote stochastic algorithm with selection criterion W_1 (6), and by AP_2 – with criterion W_2 (7). The algorithms have been tested on two classes of instances of various size and level of difficulty:

- (i) 375 benchmark instances of three different sizes with 40, 50 and 100 jobs from the OR-Library [20].
- (ii) test problems were generated as follows: for each job i , an integer processing time p_i was generated from the uniform distribution $[1, 100]$ and integer weights w_i were generated from the uniform distribution $[1, 10]$. Let $P = \sum_{i=1}^n p_i$. Distributions of deadline d_i depend on P and two additional parameters L and R which take on values from 0.2 to 1.0 in increments of 0.2. An integer deadline d_i was generated from the uniform distribution $[P(L - R/2), P(L + R/2)]$. Five problems were generated for each of the 25 pairs of values of R and L , yielding 125 problems for each value of $n = 200, 300, 400, 500$.

The set of all 975 examples of deterministic data is denoted by Ω .

For each example of deterministic data there was an example of the probabilistic data designated with the following distributions of random variables (other values are deterministic):

- (a) processing times: $\tilde{p}_i \sim N(p_i, \alpha \cdot p_i)$, where $\alpha = 0.2$,
- (b) latest due dates: $\tilde{d}_i \sim N(d_i, c \cdot d_i)$, $c = 0.1$,
- (c) processing times: $\tilde{p}_i \sim E(\alpha_i, \lambda_\alpha)$, $\alpha_i = p_i \cdot \lambda_\alpha$, $\lambda_\alpha = 1$,
- (d) latest due dates: $\tilde{d}_i \sim E(\beta_i, \mu_\beta)$, $\beta_i = d_i \cdot \mu_\beta$, $\mu_\beta = 1$,

Parameters' values $\alpha = 0.2$ (case (a)), $c = 0.1$ (case (b)) and $\lambda_\alpha = \mu_\beta = 1$ (case (c) i (d)) were established on the basis of statistical data relating to road construction and on carried out computational experiments.

For simplicity, each of the these data sets (respectively for the case(a), (b), (c) and (d)) is denoted by $\tilde{\Omega}$.

Initial permutation. The quality of solutions calculated by tabu search algorithm strongly depend on the starting point. Below we present the constructive heuristic algorithm which computes these solutions. It's based on the idea of NEH algorithm [19] and creates n elements' permutation $\pi \in \Phi$.

For the job $i \in J$ and the number $x \geq 0$, let $f_i(x) = \max\{0, x - d_i\}$.

Algorithm. CA {Constructive Algorithm}

Enumerate jobs such as $p_1/w_1 \geq p_2/w_2 \geq \dots, \geq p_n/w_n$;
 $l := 1$;

for $i := 2$ **to** n **do**

begin

Insert a job i on one of positions $1, 2, \dots, l$

so that the sum

$\sum_{j=1}^l f_{\pi(j)}(C_{\pi(j)})$ was minimal,

where $C_{\pi(j)} = \sum_{s=1}^j p_{\pi(s)}$;

$l := l + 1$

end {for}

The CA algorithm requires $O(n^2)$ time.

Deterministic algorithms CA and AD on the examples from the set Ω , and probabilistic AP_1, AP_2 on the examples from the set $\tilde{\Omega}$.

Parameter selection. For parameters' values determination preliminary experiments have been conducted for randomly generated instances (5 per each number of jobs $n = 40, 50, 100$). After analysis of the received results we have assumed: the length of tabu list $ITS(ite) = \lceil \sqrt{n} \rceil$, and the number of iterations algorithm $Maxiter = n^2$.

6.1. Comparative study. Firstly, the quality of the examples of solutions designated by separate algorithms from (i) was examined. For probabilistic algorithms it was assumed that the execution time of tasks are random variables with normal distribution (see case (A)). We compared the results of the deterministic (CA, AD) and probabilistic algorithms (AP_1, AP_2) with the results taken from the OR-Library page [20] For each test instance we have computed values: F^A – the makespan found by the algorithm $A \in \{CA, AD, AP_1, AP_2\}$, and $(F^A - UB)/UB \cdot 100\%$ – the relative percentage deviation of the makespan F^A to the best known upper bound UB (taken from [20]).

For each size n (group of test instances) we have calculated the following values:

- δ_{aprd} – the average value (for 125 instances) of the relative percentage deviation of the cost function F^A (found by algorithm A) to the best known upper bound,
- δ_{mrpd} – the maximal relative percentage deviation from the upper bound value.

We present the results obtained for the test problems of class (i) in Table 2. When comparing the average relative error it turns out that, regardless of the number of tasks, deterministic algorithm AD sets significantly better solutions than both probabilistic algorithms. The average relative error of the deterministic algorithm is 0.51% and is considerably smaller (by about about 30%) than errors for probabilistic algorithms. It is especially clear for examples with a greater number of tasks. Both probabilistic algorithms AP_1 and AP_2 set similar solutions because their errors vary little and are respectively 0.70% and 0.75%. According to predictions the worst algorithm appeared to be a construction algorithm CA . The average relative error was 6.06% and almost 12 times greater than the error of the algorithm AD . Average maximum errors (δ_{mrpd}) have similar proportions. By far the best appeared to be algorithm AD . The worst set by this algorithm solutions differ on average by 0.67% from the best currently known values given in [20]. Computation time of one algorithm for all the 375 examples did not exceed 5 seconds. Since for the examples of the number of

Table 2

Comparison of results of the algorithms with results taken from the OR- Library [20] (class (i))

n	Algorithm CA		Algorithm AD		Algorithm AP_1		Algorithm AP_2	
	δ_{aprd}	δ_{mrpd}	δ_{aprd}	δ_{mrpd}	δ_{aprd}	δ_{mrpd}	δ_{aprd}	δ_{mrpd}
40	4.21	5.23	0.11	0.16	0.19	0.26	0.17	0.21
50	5.33	7.57	0.24	0.32	0.28	0.39	0.31	0.44
100	8.64	10.14	1.17	1.54	1.63	2.42	1.76	2.16
all	6.06	7.65	0.51	0.67	0.70	1.02	0.75	0.94

Table 3

The comparison of results of AD, AP_1, AP_2 algorithms with the results of the constructive algorithm CA (for data of the class (ii))

n	Algorithm AD		Algorithm AP_1		Algorithm AP_2	
	δ_{aprd}	δ_{mrpd}	δ_{aprd}	δ_{mrpd}	δ_{aprd}	δ_{mrpd}
200	-8.74	-11.27	-5.14	-8.44	-5.93	-7.93
300	-10.87	-14.55	-9.38	-12.37	-8.17	-12.11
400	-13.41	-17.84	-9.98	-13.53	-9.81	-14.72
500	-16.58	-19.26	-11.37	-14.81	-10.47	-13.52
all	-12.40	-15.73	-8.96	-12.28	-8.59	-12.07

tasks bigger than 100, there is no reference data in the literature, we compare the results of algorithms AD, AP_1 and AP_2 with the results of the design algorithm CA . Calculations performed on test instances of class (ii). The obtained results were presented in Table 3. They are similar to those listed in Table 2, the highest average (relative) improvements of solutions -12.40% set by AC algorithm was obtained by AD algorithm. Both stochastic algorithms set similar solutions. Average improvements differ slightly between each other and are respectively -8.96% and -8.59%. Similar proportions were observed for maximum improvement. Computation time of one algorithm in all 600 examples, did not exceed 3 minutes. Taken into consideration the simplicity of algorithms and a small number of iterations it must be emphasized that the set solutions are fully satisfactory.

6.2. Stability of algorithms. In order to investigate the stability of algorithms there were disturbed data sets generated. The basis was constituted on 975 examples of deterministic data from the set Ω described in (i) and (ii). For the example of deterministic data $\delta = (p_i, w_i, d_i)_{i=1,2,\dots,n}$, $\delta \in \Omega$ there were 100 examples of disturbed data $D(\delta)$ generated according to the following distributions:

- a) problem 1 | $\tilde{p}_i \sim N(p_i, a \cdot p_i) | \sum w_i T_i, \tilde{p}_i \sim N(p_i, 0, 2 \cdot p_i)$,
- b) problem 1 | $\tilde{d}_i \sim N(d_i, c \cdot d_i) | \sum w_i T_i, \tilde{d}_i \sim N(d_i, 0, 1 \cdot d_i)$,
- c) problem 1 | $\tilde{p}_i \sim E(\alpha_i, \lambda) | \sum w_i T_i, \tilde{p}_i \sim E(p_i, 1)$,
- d) problem 1 | $\tilde{d}_i \sim E(\beta_i, \chi) | \sum w_i T_i, \tilde{d}_i \sim E(p_i, 1)$.

In total, for each variant of probabilistic problem a), b), c) and d) there 97 500 examples of disturbed data generated. They were then solved by the algorithm AD , whose solutions constituted the basis for determining the stability coefficient of examined algorithms. Table 4 shows the results for problems with random parameters with the normal distribution. Both probabilistic algorithms have stability coefficient significantly smaller than the deterministic algorithm. For random times of the tasks execution probabilistic algorithm AP_1 has a stability coefficient of 3.91% and it is more than twice lower than 8.25% – coefficient of deterministic algorithm AD . Coefficient 3.91% of AP_1 algorithm shows that the random disturbance of tasks execution times decreases the value of the objective function (in relation to solutions of AD) algorithm on average by 3.91%. In case of random deadlines for the completion of tasks the smallest stability coefficient has AP_2 algorithm. It amounts to

Table 4
Stability coefficients (42) for random parameters examples with the normal distribution (case A and B)

n	Case A ($\tilde{p}_i \sim N(p_i, a \cdot p_i)$)			Case B ($\tilde{d}_i \sim N(d_i, c \cdot d_i)$)		
	AD	AP ₁	AP ₂	AD	AP ₁	AP ₂
40	4.75	2.16	1.89	4.88	4.11	3.72
50	4.99	2.08	2.25	6.03	3.87	3.54
100	6.07	3.75	3.61	8.48	6.28	6.21
200	6.83	3.63	4.38	13.27	7.55	7.66
300	8.66	4.11	5.35	15.17	9.13	7.72
400	11.07	4.82	5.89	17.75	8.96	9.74
500	15.36	6.86	7.06	22.46	11.04	11.57
Average	8.25	33.91	4.35	12.58	7.23	7.16

7.16%. AP₁ algorithm has slightly larger coefficient: 7.23%, whereas deterministic AD algorithm – the largest: 12.58%.

Computational results for the examples of processing times or tasks due dates with Erlang distribution were given in Table 5. For cases where the task parameters are random variables with Erlang distribution stability of solutions (presented in Table 5) is similar to the normal distribution (Table 4). Both probabilistic algorithms are much more stable than the deterministic algorithm. In summary, on the basis of the given results it can be concluded that probabilistic algorithms are significantly more stable. The solutions determined by them are much less sensitive to any random change in the parameters of the problem. For random processing times of tasks algorithm AP₁ is more stable, whereas for the tasks due dates – slightly better is algorithm AP₂.

Table 5
Stability coefficients (42) for examples random parameters examples with Erlang distribution (case C and D)

n	Case C ($\tilde{p}_i \sim E(\alpha_i, \lambda)$)			Case D ($\tilde{d}_i \sim E(\beta_i, \mu)$)		
	AD	AP ₁	AP ₂	AD	AP ₁	AP ₂
40	2.56	1.83	2.01	4.89	3.34	2.86
50	3.16	1.97	1.62	5.11	3.82	3.44
100	4.27	2.64	3.07	7.93	5.56	4.73
200	5.08	3.37	3.12	9.36	7.28	8.12
300	7.11	4.27	4.68	13.17	9.88	7.87
400	9.48	5.13	5.26	16.58	9.66	10.08
500	13.88	5.76	6.24	21.43	10.08	11.17
Average	6.51	3.56	3.71	11.21	7.08	6.89

7. Remarks and conclusions

In this work there was considered a problem of uncertain data modeling methods with the use of random variables with normal or Erlang distribution, which well describes the ‘natural’

randomness most often met while dealing with management practices. The paper presents the design of algorithm based on the tabu search method for a single machine jobs scheduling problem. Computational experiments were conducted to investigate the stability of algorithms, that is, the problem of the impact of the disorder parameter on changes in values of the optimized criterion. The obtained results clearly indicate that much more stable are the probabilistic algorithms, that is the algorithms, in which, as the comparative criterion, there was a function of central moments of random goal functions adopted.

Possible implementation of proposed methods includes optimization of single production nest of the manufacturing system (bottleneck) and its generalization to multi-machine problems with uncertain parameters (e.g dependent on the weather) which can be modelled by probabilistic processing times, for instance in construction or industry.

REFERENCES

- [1] B. Alidee, I. Dragon, “A note on minimizing the weighted sum of tardy and early completion penalties in a single machine”, *Journal of Operational Research* 96 (3), 559–563 (1997).
- [2] W. Bożejko, J. Grabowski, M. Wodecki, “Block approach-tabu search algorithm for single machine total weighted tardiness problem”, *Computers & Industrial Engineering* 50, 1–14 (2006).
- [3] W. Bożejko, M. Wodecki, “On the theoretical properties of swap multimoves”, *Operations Research Letters* 35(2), 227–231 (2007).
- [4] W. Bożejko, “Parallel path relinking method for the single machine total weighted tardiness problem with sequence-dependent setups”, *Journal of Intelligent Manufacturing* 21(6), 777–785 (2010).
- [5] X. Cai, X. Zhou, “Single-machine scheduling with exponential processing times and general stochastic cost functions”, *Journal of Global Optimization* 31, 317–332 (2005).
- [6] X. Cai, X. Wu, X. Zhou, “Optimal stochastic scheduling”, *International Series in Operations Research & Management Science* 207, Springer, 2014.
- [7] H.A.J. Crauwels, C.N. Potts, L.N. Van Wassenhove, “Local search heuristics for the single machine total weighted tardiness scheduling problem”, *INFORMS Journal on Computing* 10(3), 341–350 (1998).
- [8] B.C. Dean, “Approximation algorithms for stochastic scheduling problems”, PhD thesis, Massachusetts Institute of Technology (2005).
- [9] M.T. Den Basten, M. Stützle, M. Dorigo, “Design of iterated local search algorithms. An example application to the single machine total weighted tardiness problem”, J.W.Boers et al. (eds.) *Evo Worskshop 2001, Lecture Notes in Computer Science* 2037, 441–451 (2001).
- [10] A. Elyasi, N. Salmasi, “Stochastic scheduling with minimizing the number of tardy jobs using chance constrained programming”, *Mathematical and Computer Modeling* 57, 1154–1164 (2013).
- [11] Hu, K., Zhang, X., Gen, M., Jo J., “A new model for single machine scheduling with uncertain processing time”, *Journal of Intelligent Manufacturing*, DOI 10.1007/s10845–015–1033–9, 1–9 (2015).

- [12] W. Jang, "Dynamic scheduling of stochastic jobs a single machine", *European Journal of Operational Research*, 138, 518–503 (2002).
- [13] G. Kirlik, C. Oguz, "A variable neighbourhood search for minimizing total weighted tardiness with sequence dependent setup times on a single machine", *Computers&Operatins Research* 39, 1506–1520 (2012).
- [14] A. Lann, G. Mosheiov, "Single machine scheduling to minimize the number of early and tardy jobs", *Computers and Operations Research*, 23(8), 769–781 (1996).
- [15] J.K. Lenstra, A.G.H. Rinnoy Kan, A.G.H., P.Brucker, "Complexity of machine scheduling problems", *Annals of Discrete Mathematics* 1, 343–362 (1977).
- [16] W. Li, W.J. Braun, Y.Q. Zhao, "Stochastic scheduling on a repairable machine with Erlang uptime distribution", *Advances in Applied Probability* 30(4), 1073–1088 (1998).
- [17] Y. Li, R. Chen, "Stochastic single machine scheduling to minimize the weighted number of tardy jobs", B.-Y.Cao et al. (eds.): *Fuzzy Information and Engineering*, AISC, 363–368 (1998).
- [18] M.M Mazdeh., H. Hadad, P. Ghanbari, "Solving a single machine stochastic scheduling problem using a branch and bound algorithm and simulated annealing", *International Journal of Management Science and Engineering Management* 7(2), 110–118 (2012).
- [19] M. Nawaz, E.E. Enscore, I. Ham, "A heuristic algorithm for the m -machine, n -job flow-shop sequencing problem", *OMEGA* 11(1), 91–95 (1983).
- [20] OR-Library <https://files.nyu.edu/jeb21/public/jeb/info.html>
- [21] M.L. Pinedo: *Scheduling: Theory, Algorithms, and Systems*, fourth edition, Springer New York Dordrecht Heidelberg London (2010).
- [22] C.N. Potts, L.N. Van Wassenhove, "A branch and bound algorithm for the total weighted tardiness problem", *Operations Research* 33, 177–181 (1985).
- [23] C.N. Potts, L.N. Van Wassenhove, "Single machine tardiness sequencing heuristics", *IIE Transactions* 23, 346–354 (1991).
- [24] W.E. Smith, "Various optimizers for single-stage production", *Naval Research Logist Quartely* 3, 59–66 (1956).
- [25] H.M. Soroush, "Single machine scheduling with stochastic processing times or stochastic due-dates to minimize the number of early and tardy jobs", *International Journal of Operations Research* 3(2), 90–108 (2006).
- [26] H.M. Soroush, "Minimizing the weighted number of early and tardy jobs in a stochastic single machine scheduling problem", *European Journal of Operational Research* 181, 266–287 (2007).
- [27] J.M. Van den Akker, H. Hoogeveen, "Minimizing the number of late jobs in a stochastic setting using a chance constraint", *Journal of Scheduling* 11, 59–69 (2008).
- [28] J. Vondrák, "Probabilistic methods in combinatorial and stochastic optimization", PhD Thesis, MIT (2005).
- [29] M. Wodecki, "A branch-and-bound parallel algorithm for single-machine total weighted tardiness problem", *Advanced Manufacturing Technology* 37, 996–1004 (2008).
- [30] M. Wodecki, "A block approach to earliness-tardiness scheduling problems", *International Journal on Advanced Manufacturing Technology* 40, 797–807 (2009).