ARCHIVES OF CIVIL AND MECHANICAL ENGINEERING 19 (2019) 194–204

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: http://www.elsevier.com/locate/acme

Original Research Article

Flowshop scheduling of construction processes with uncertain parameters

Wojciech Bozejko [a,*], *Zdzislaw Hejducki* [b], *Mieczyslaw Wodecki* [c]

[a] *Wrocław University of Science and Technology, Faculty of Electronics, Department of Control Systems and Mechatronics, Janiszewskiego 11-17, 50-372 Wrocław, Poland*

[b] *Wrocław University of Science and Technology, Faculty of Civil Engineering, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland*

[c] *Telecommunications and Teleinformatics Department, Faculty of Electronics, Wrocław University of Science and Technology, Janiszewskiego 11-17, 50-372 Wrocław, Poland*

ARTICLE INFO

Article history:
Received 12 March 2018
Accepted 30 September 2018
Available online 19 October 2018

Keywords:
Discrete optimization
Stochastic parameters
Applications in construction

ABSTRACT

In the paper a construction scheduling problem – namely flowshop – with minimizing the sum of penalties for exceeding the deadline of building structures completion is considered. The problem is illustrated by the investment task concerning the implementation of twelve apartment buildings forming a part of a new housing estate. Uncertain parameters of the system are represented either by fuzzy numbers or random variables, whereas random variables have normal or the Erlang distribution. Since even the deterministic version of the problem is strongly NP-hard, the approximate algorithm based on the tabu search method was used to its solution. The performed computational experiments showed large solution resistance against any potential interference of parameters of the problem.

1. Introduction

In the process of negotiation of construction contracts there appears the necessity to estimate a clear deadline for work completion. This task is very difficult due to the high degree of uncertainty resulting from the existence of many parameters changing during the execution of works. Exceeding the established deadlines causes considerable losses (contractual penalties or unused resources). Hence, the need for such modeling of building projects that most closely match the course of the actual construction processes is very important (see, e.g. [16,9,14,15]). The above mentioned need leads to complex discrete-continuous optimization problems with uncertain parameters and irregular goal functions. While transferring the problems of construction projects scheduling in the field of classical theory of tasks scheduling one encounters many difficulties associated with choosing not only the right model but also the appropriate algorithm. The above mentioned tasks are usually brand new, *strongly NP – hard combinatorial optimization problems* (see [1,6]), for which

* Corresponding author.
E-mail addresses: wojciech.bozejko@pwr.edu.pl (W. Bozejko), zdzislaw.hejducki@pwr.edu.pl (Z. Hejducki), mwd@ii.uni.wroc.pl (M. Wodecki).

https://doi.org/10.1016/j.acme.2018.09.010

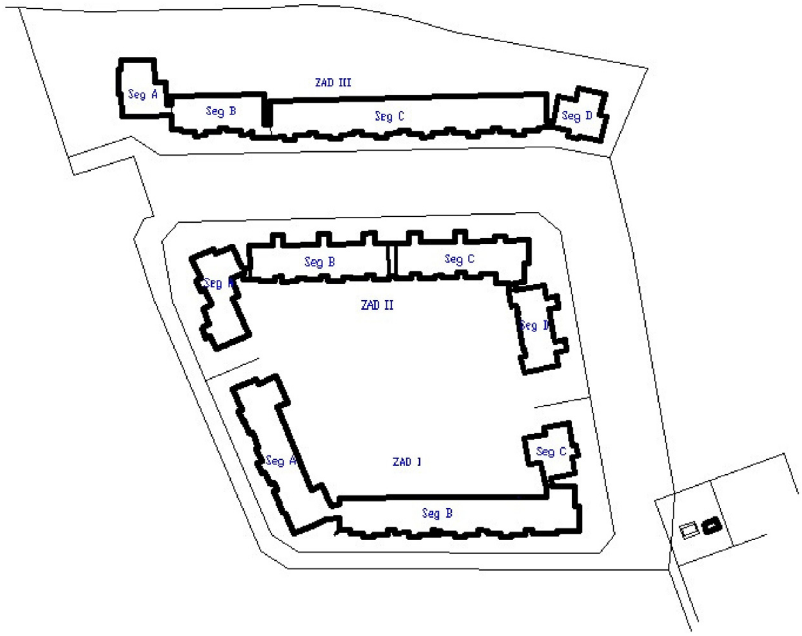

Fig. 1 – Object oriented layout plan.

today there are not known optimized algorithms of polynomial computational complexity. Therefore, in practice, there are fast approximate algorithms with elements of artificial intelligence often used.

An integral part of many management systems is planning of construction projects in flow systems [16,13,4,20]. They concern the implementation of complex building structures consisting of many identical works (for example: earthworks, foundation works, construction works, roofing, installation of windows or facades, finishing works, etc.) arising in the field of construction documentation, performed by working brigades.

The investment task considered in the case study concerns the implementation of twelve apartment buildings forming part of a new housing estate. They are similar in terms of technology. The basis for estimating the time of work performance are workpieces and the Catalog of Material Outlays, on the basis of which work labor intensity can be determined. After establishing and agreeing the necessary human resources (work groups), the possible duration of works is calculated, transforming a number of man-hours onto working days. The data determined in this way form the Structure of the Division of Works, defining the technological order of works.

Fig. 1 shows the design of the complex of build objects. Each object denoted by ZAD I, ZAD II and ZAD III consists of several similar (identical) segments denoted as Seg A, Seg B, Seg C and Seg D (at the bottom of the ground, service facilities, and above them – residential). The order of construction works, on each of the segments is the same. However, due to the standards of finish, size of rooms, etc. times of similar works are different. In the general scope of works, operations which will be carried out by specialized staff brigades were separated, which makes possible to order the tasks as a flow work system (*flowshop*). This made it possible to plan the delivery of materials in the Just-in-Time system, i.e. without storing them. Fig. 2 shows a fragment of one of the objects during the execution of facade works, as a one of final steps of pipeline scheme of works.

Fig. 2 – Facade works.

Flow systems in the field of construction are equivalent of the *flow production* in the industry (flowshop). Build object are recognized as tasks (jobs), working brigades – machines, and work made by brigades is recognized as operations.

Due to such factors as the use of both new techniques and technologies, the unique, atmospheric and geological conditions, etc. there is often no possibility to determine, in an unambiguous way, the value of certain parameters of schedules of works, specifically to determine the completion time for all works. In such cases we deal with the decision-making process under uncertainty.

It is commonly believed that the data collected in management practices are usually uncertain and imprecise. Their values change frequently already at the stage of adopted solutions, destroying its optimality, sometimes admissibility. In addition, the instability leads to completely useless in practice solutions. Thus, the uncertainty of the data translates directly to the amount of risk. The choice of approach for modeling and analysis of unreliable data depends on the characteristics of the system, possibilities of data measurements execution, credibility of the data, effectiveness of theoretical tools, power of available software packages, etc. Knowledge of all the above mentioned elements is essential in the process of efficient solving of practical problems.

Flow work systems in construction are of interest in many scientific centers. It results from the need of optimal management of construction of large engineering structures, among others: sections of roads, bridges, building complexes, industrial facilities, etc. There is a great need to improve traditional methods of scheduling (taking into account e.g. new technologies or data uncertainty) and effective use of modern computational methods.

In this paper we consider the problem of minimizing the sum of penalties for exceeding of the agreed deadlines for the completion of building structures. It belongs to a class of strongly NP-hard problems. We assume, moreover, that the times of works execution in building structures are uncertain. We model each with a triangular fuzzy numbers and random variables with normal or the Erlang distribution. Many authors consider normal distribution as a behaviour of construction works duration times. We can cite here Czarnigowska and Sobotka [7] in which construction duration normality check was based on the analysis of residual histograms, scatter diagrams and normality tests of Kolmogorov–Smirnov/Lilliefors and Shapiro–Wilk. A mathematical model, solution algorithms and computational experiments are presented in our work. The main purpose of the carried out experiments was to examine the effect of disturbing the execution times of construction works on changing of the penalty functions for failure to meet the deadlines for completion of individual building structures.

2. Flowshop systems in construction

We consider the construction project (abbreviated to **CP**) consisting in execution of n structures from the set

$$O = \{O_1, O_2, \ldots, O_n\},$$

by m brigades from the set

$$B = \{B_1, B_2, \ldots, B_m\}.$$

Each structure $O_i \in O$ is a sequence of m works

$$O_i = [P_{i,a}, P_{i,b}, \ldots, P_{i,m}],$$

wherein the work $P_{i,j}$ ($i = 1, b, \ldots, n$, $j = 1, 2, \ldots, m$) in structure O_i is executed by brigade B_j in time $p_{i,j}$. Works in the structure $O_i \in O$ must be done in a given *technological order*, i.e. any job $P_{i,j}$ is to be carried out after the completion of $P_{i,j-1}$, but before the commencement of $P_{i,j+1}$ ($2 \le j \le m-1$). At the same time the following constraints must be met:

(i) every work (in the building structure) can be executed only by one, determined by the technological order (sequence), brigade
(ii) any brigade cannot perform at the same time more than one work,
(iii) technological order must be maintained on every building structure,
(iv) execution of any work cannot be interrupted before its completion.

The problem is to determine the order of execution, by each of the brigades, work in the structures (i.e. permutation of objects), for which certain fixed criterion reaches its optimum and restrictions (i) – (iv) are met.

Any solution to the problem in question can be represented by permutation of structures. Let π be some permutation of structures (elements of the set O). This permutation determines the order of execution of subsequent works in order, i.e. brigade $B_j \in B$ carries out the work $P_{\pi(i),j}$ in structure $\pi(i) \in O$, after execution of works $P_{\pi(1),j}, P_{\pi(2),j}, \ldots, P_{\pi(i-1),j}$ successively in order $\pi(1), \pi(2), \ldots, \pi(i-1)$, but before performing works $P_{\pi(i+1),j}$, $P_{\pi(i+2),j}, \ldots, P_{\pi(n),j}$ in structures $\pi(i+1), \ldots, \pi(n)$. Let us denote by Φ the set of all the possible permutations of structures. The cardinality of this set is $n!$.

If work in the structures are performed in the order of $\pi \in \Phi$ whereas $p_{\pi(i),j}$ is the time of execution of work $P_{\pi(i),j}$ ($i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$) then a moment of the completion of work $C_{\pi(i),j}$ can be determined from the following recursive relationship:

$$C_{\pi(i),1} = \sum_{k=1}^{i} p_{\pi(k),1}, \quad i = 2, c, \ldots, n, \tag{2.1}$$

$$C_{\pi(1),j} = C_{\pi(1),j-1} + p_{\pi(1),j}, \quad j = 2, 3, \ldots, m, \tag{2.2}$$

$$C_{\pi(i),j} = \max\{C_{\pi(i),j-1},\ C_{\pi(i-1),j}\} + p_{\pi(i),j}\}, \tag{2.3}$$

for $i = 2, 3, \ldots, n$, $j = 2, 3, \ldots, m$.

In this case, the moments of the beginning of works are

$$S_{\pi(i),j} = C_{\pi(i)j} - p_{\pi(i),j}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m. \tag{2.4}$$

It can be easily checked that determined by (2.1)–(2.4) moments of commencement and completion of works on the building structures meet the limitation (i)–(iv), hence they are the acceptable solutions to **CP** problem.

Model of the above described construction project is known in the theory of scheduling as a *flowshop problem*. If we consider the criterion of minimizing the completion time of all structures (C_{max}), then it belongs to a class of *strongly NP-hard*. Due to the exponentially growing computation time of exact algorithms, these problems are usually solved with heuristic methods. A comprehensive overview of methods and algorithms, including solutions based on artificial intelligence, were presented in the works [10,18].

In construction practice, a very important criterion is execution of each building structures, which meets the deadline, or a possible minimization of penalties for exceeding of agreed, in the contract, terms. These are very difficult problems with irregular functions. In the further part of the work we will present one of such examples.

For ith structure $O_i \in O$, let d_i be its *required completion time*, and w_i *weight* of tardiness penalty function. If $\pi \in \Phi$ is a certain order of execution of works in structures and $C_{\pi(i)} = C_{\pi(i),m}$ is *date of completion* of works in structure, and $\pi(i)$ is an ith job of an order π, then $T_{\pi(i)} = \max\{0, C_{\pi(i)} - d_{\pi(i)}\}$ is *tardiness*, whereas $w_{\pi(i)} \cdot T_{\pi(i)}$ is *tardiness* penalty for structure execution. Then

$$F(\pi) = \sum_{i=1}^{n} w_{\pi(i)} \cdot T_{\pi(i)} \tag{2.5}$$

is *tardiness penalty* for structure execution in the order π (in short *permutation penalty* π).

In the next part we consider the problem of **CP** with minimizing function (2.5). This amounts to designating a permutation $\pi^* \in \Phi$ such that

$$F(\pi^*) = \min\{F(\pi): \ \pi \in \Phi\}. \tag{2.6}$$

The problem **CP** with minimizing the sum of the penalties for tardiness of construction structures execution (2.5) will be denoted by **CPF**. It belongs to a class of *stronglyNP-hard* problems. Indeed, if we assume that the set of brigades (machines) $M = \{1\}$, then we get a strongly *NP*-hard single machine problem of tasks scheduling with minimization of the sum of the cost of tardiness (in literature it is denoted by $1||\sum w_i T_i$). Multi-machine problems with sum-cost goal functions were considered in the works: [19,23,11].

3. Flowshop system with uncertain work times

Discrete optimization problems with uncertain parameters are solved with the use of probabilistic methods or theory of fuzzy sets. In the first case (e.g. [8,19]) distribution of random variables is important. Some processes are inherently random and depend upon weather, traffic, number of accidents, geological conditions, equipment failure, etc. If they also possess certain history, on the basis of existing statistical data, it is possible to specify their distributions.

However, in many matters, uncertainty of the data is non-random, but results from the uniqueness of the process (e.g. construction, technology, materials, etc.), error in measurement, etc. In such a case, the natural way of representing uncertainty are fuzzy numbers. The problem there is the proper selection of membership function and defuzzification method (i.e. conversion of fuzzy numbers into real numbers). In the following part of the work we use both of the approaches to scheduling of work performed in the flow system. For simplicity, wherever the ambiguity can be avoided the order of building structures execution(permutation) $\pi = (1, 2, \ldots, n)$ is assumed.

4. Uncertain tasks times

It is assumed that the times of works execution are not deterministic. They will be represented either by fuzzy numbers or random variables.

4.1. *Fuzzy tasks times*

In this paper the fuzzy tasks times are represented by a triangular membership function μ (i.e. 3-tuple $\widehat{p}_{i,j} = (p^a{}_{i,j}, p^b{}_{i,j}, p^c{}_{i,j})$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$ with the following properties:

(*a*) $p^a{}_{i,j} \le p^b{}_{i,j} \le p^c{}_{i,j}$,
(*b*) $\mu(x) = 0$ for $x \le p^a_{i,j}$ or $x \ge p^c_{i,j}$,
(*c*) $\mu(p^b_{i,j}) = 1$,
(d) μ is increasing on $[p^a_{i,j}, p^b_{i,j}]$ and decreasing on $[p^b_{i,j}, p^c_{i,j}]$.

Let $\theta = \langle \mathbf{p}, \mathbf{d}, \mathbf{w} \rangle$ (where $\mathbf{p}=[p_{i,j}]_{n\times m}$ is the matrix of the works execution times, whereas $\mathbf{d}$ and $\mathbf{w}$ are n element vectors of respectively deadlines for the completion of structures and weights of penalty function) be an example of deterministic data to **CPF** problem. We assume that the time for works execution, i.e. elements of matrix $[p_{i,j}]_{n\times m}$, are triangular fuzzy numbers of the form:

$$\widehat{p}_{i,j} = (p_{i,j}{}^a, \ p_{i,j}{}^b, \ p_{i,j}{}^c).$$

Then, $\widehat{\theta} = \langle \widehat{\mathbf{p}}, \mathbf{d}, \mathbf{w} \rangle$, where $\widehat{\mathbf{p}} = [\widehat{p}_{i,j}]_{n\times m}$ is the matrix of fuzzy numbers, called *fuzzy data*, and the problem is fuzzy, in short denoted by $\widehat{\text{CPF}}$.

The execution is determined by a fuzzy number

$$\widehat{p}_{i,j} = (p_{i,j}{}^a, \ p_{i,j}{}^b, \ p_{i,j}{}^c),$$

then its finishing time is a fuzzy number in the form of:

$$\widehat{C}_{i,j} = \left(C^a_{i,j}, C^b_{i,j}, C^c_{i,j}\right),$$

where $C^a_{i,j}$, $C^b_{i,j}$ and $C^c_{i,j}$ can be determined from the following recurrent formulas:

$$\begin{aligned} C^a_{i,j} &= \max\{C^a_{i-1,j}, \ C^a_{i,j-1}\} + p^a_{i,j}, \\ C^b_{i,j} &= \max\{C^b_{i-1,j}, \ C^b_{i,j-1}\} + p^b_{i,j}, \\ C^c_{i,j} &= \max\{C^c_{i-1,j}, \ C^c_{i,j-1}\} + p^c_{i,j}, \end{aligned}$$

with the initial conditions

$$C^a_{0,j} = C^b_{0,j} = C^c_{0,j}, \quad j = 1, 2, \ldots, m,$$

$$C^a_{i,0} = C^b_{i,0} = C^c_{i,0}, \quad i = 1, 2, \ldots, n.$$

By $\widehat{C}_i = \widehat{C}_{i,m}$ fuzzy number $\widehat{C}_i = (C^a_i, C^b_i, C^c_i)$ is denoted representing the moment of completion of the structure $i \in O$. Then deffuzification is carried out, i.e. conversion of fuzzy numbers $\widehat{C}_i = (C^a_i, C^b_i, C^c_i)$, $i = 1, 2, \ldots, n$, into real number as follows:

$$C^f_i = \frac{1}{4}\left(C^a_i + C^b_i + C^b_i + C^c_i\right). \tag{4.1}$$

Then, tardiness(equivalent of T_i)

$$T^f_i = \begin{cases} d_i - C^f_i, & \text{if } C^f_i > d_i, \\ 0, & \text{if } C^f_i \le d_i, \end{cases}$$

and the comparison criterion of solutions (equivalent of function (2.5) for a permutation $\pi \in \Phi$) is

$$F^f(\pi) = \sum_{i=1}^{n} w_{\pi(i)} T^f_{\pi(i)}. \tag{4.2}$$

The algorithm solving the $\widehat{CPF}$ problem, in which as the comparison criterion there will be solutions function (4.2) applied, will be called *fuzzy algorithm*.

4.2. Probabilistic tasks times

Let $\theta = \langle \mathbf{p}, \mathbf{d}, \mathbf{w} \rangle$ be an instance of the problem with deterministic data for **CPF** problem. It is assumed that the times of execution of works $p_{i,j}$, $i \in O$, $j \in B$ are independent random variables $\tilde{p}_{i,j}$. Similarly as in the case of fuzzy data, the three $\tilde{\theta} = \langle \tilde{\mathbf{p}}, \mathbf{d}, \mathbf{w} \rangle$, where $\tilde{\mathbf{p}} = [\tilde{p}_{i,j}]_{n \times m}$ is matrix of random variables, are called the *probabilistic data*, and the problem – *probabilistic*, in short denoted by $\widetilde{CPF}$.

As a comparison criterion of solutions (the equivalent of function (2.5), for a permutation $\pi \in \Phi$), in the algorithm solving the problem of $\widetilde{CPF}$, we will use two functions:

$$F^p_E(\pi) = \sum_{i=1}^{n} w_{\pi(i)} E(\tilde{T}_{\pi(i)}), \tag{4.3}$$

$$F^p_{ED}(\pi) = \sum_{i=1}^{n} w_{\pi(i)} (E(\tilde{T}_{\pi(i)}) + \tau \cdot D(\tilde{T}_{\pi(i)})), \tag{4.4}$$

where $E(\tilde{T}_i)$ is the expected value, and $D(\tilde{T}_i)$ – the standard deviation of random variable $\tilde{T}_i$ – tardiness of ith object. Parameter τ is determined by the expert or experimentally.

By F_X and f_X we will denote respectively the cumulative distribution and density function of the random variable X. For considered in the work distributions of random variables (normal or Erlang), density function is equal to the derivative of the cumulative distribution function, i.e. $f_X(x) = F'_X(x)$. What is more, a well-known equality will be used

$$D^2(X) = E(X^2) - (E(X))^2. \tag{4.5}$$

Hence, in order to calculate the variance of random variable X, it is necessary only to calculate the expected values of the variables X and X^2. When calculating the values of the function F^p_E (4.3) and F^p_{ED} (4.4) we will be using the properties that were proven in the paper Bożejko et al. [5].

Normal distribution

The problem of $\widetilde{CPF}$ is considered, in which times of works execution are independent random variables with normal distribution $\tilde{p}_{i,j} \sim N(p_{i,j}, \alpha_{i,j} \cdot p_{i,j})$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. Parameters $\alpha_{i,j}$ are determined by an expert or experimentally. In this particular case, the times of completion of structures execution are also random variables. We can approximate random tasks completion times by random variables which have a normal distribution

$$\tilde{C}_i \sim N(\mu_i, \sigma_i), \tag{4.6}$$

where

$$\mu_i = \alpha_i \cdot (p_{i,1} + p_{i,2} +, \ldots, p_{i,m}) \text{ and } \sigma_i = \beta_i \cdot \sqrt{p^2_{i,1} + p^2_{i,2} +, \ldots, p^2_{i,m}}, \tag{4.7}$$

are respectively the expected value and standard deviation. The values of the parameters α_i, β_i ($i = 1, 2, \ldots, n$) will be determined experimentally. In this case, the tardiness of the structures execution is also a random variable

$$\tilde{T}_i = \begin{cases} \tilde{C}_i - d_i, & \text{if } \tilde{C}_i > d_i, \\ 0, & \text{if } \tilde{C}_i \le d_i. \end{cases} \tag{4.8}$$

It is easy to show, that the cumulative distribution of the $\tilde{T}_i$ variable is

$$F_{\tilde{T}_i}(x) = \begin{cases} F_{\tilde{C}_i}(d_i + x) - F_{\tilde{C}_i}(d_i + x) F_{\tilde{C}_i}(d_i) + F_{\tilde{C}_i}(d_i), & \text{if } x > 0, \\ 0, & \text{if } x \le 0. \end{cases} \tag{4.9}$$

Equations which makes possible $E(\tilde{T}_{\pi(i)})$ and $E(\tilde{T}^2_{\pi(i)})$ determination were proven in the paper Bożejko et al. [5]. Therefore, if the times of tasks execution are independent random variables normally distributed, then – when calculating the value of function (4.3) or (4.4).

Erlang distribution

We assume that the moments of execution of works $p_{i,j}$ are independent random variables having the Erlang distribution, $p_{\widetilde{i,j}} \sim E(p_{i,j}, 1)$. Similarly, the moments of completion of building structures are random variables having the Erlangs distribution

$$\tilde{C}_i \sim E(\mu_i, \lambda_i), \tag{4.10}$$

where

$$\mu_i = \alpha_i \cdot (p_{i,1} + p_{i,2} +, \ldots, p_{i,m}) \text{ and } \lambda_i = \omega_i \cdot i. \tag{4.11}$$

Parameters α_i and ω_i are determined experimentally.

In this case, the tardiness in the execution of ith structure is

$$\tilde{T}_i = \begin{cases} \tilde{C}_i - d_i, & \text{if } \tilde{C}_i > d_i, \\ 0, & \text{if } \tilde{C}_i \le d_i, \end{cases} \tag{4.12}$$

and distribution function

$$\tilde{T}_i(x) = \begin{cases} (1 - F_{\tilde{C}_i}(d_i)) f_{\tilde{C}_i}(x + d_i), & \text{if } x > 0, \\ 0, & \text{if } x \le 0. \end{cases} \tag{4.13}$$

Similar as for the normal distribution, values of $E(\tilde{T}_{\pi(i)})$ and $E(\tilde{T}^2_{\pi(i)})$ in F^p_E (4.3) and F^p_{ED} (4.4) can be determinate with using equations proven in the paper Bożejko et al. [5].

5. The tabu search algorithm

In solving NP-hard problems of discrete optimization we almost always use approximate algorithms. The solutions given by these algorithms are, in their appliance, fully satisfying (they often differ from the best known solutions by less than 1%). Most of them belong to the local search methods group. Their acting consists in viewing in sequence a subset of a set of acceptable solutions, and in pointing out the best one according to a determined criterion. One of this method realizations is the tabu search, whose basic criterions are:

(a) *neighborhood* – a subset of a set of acceptable solutions, whose elements are rigorously analyzed,
(b) *move* – a function that converts one solution into another one,
(c) *tabu list* – a list containing the attributes of a certain number of solutions analyzed recently,
(d) *ending condition* – most of the time fixed by the number of algorithm iterations.

Let $\pi \in \Phi$ be a (starting) permutation, L a tabu list, F costs function, and π^* the best solution found at this moment (the starting solution and initial value of π^* can be any permutation because it will be changed immediately after beginning of the algorithm work).

To solve the problems considered in this work (deterministic and with uncertain times of work execution) there was an algorithm adopted which had been included in the work Bożejko et al. [3]. In order to speed up the calculations, in addition, there were used not only multimoves but also elements of neighbourhood described in the works [2,3,22]. Below we present in details the basics elements of the algorithm.

5.1. *The move and the neighborhood*

Let $\pi = (\pi(1), \ldots, \pi(n))$ be a permutation from the Φ, and

$Z(\pi) = \{\pi(i): \ C_{\pi(i)} > d_{\pi(i)}\},$

a *set of late structures* in π.

By π_l^k ($l = 1, 2, \ldots, k-1, k+1, \ldots, n$) we mark a permutation received from π by changing in π the element $\pi(k)$ and $\pi(l)$. We can say at that point that the permutation π_l^k was generated from π by a *swap move* (*s-move*) s_l^k (it means that the permutation $\pi_l^k = s_l^k(\pi)$). Then, let $M(\pi(k))$ be a set of all the s-moves of the $\pi(k)$ element. By

$M(\pi) = \bigcup_{\pi(k) \in Z(\pi)} M(\pi(k)),$

we mean an *s-moves* set of the late elements π in the permutation. The power of the set $M(\pi)$ is top-bounded by $n(n-1)/2$.

The neighborhood $\pi \in \Phi$ is the permutation set

$N(\pi) = \left\{ s_l^k(\pi) : s_l^k \in M(\pi) \right\}.$

While implementing the algorithm, we remove from the neighborhood the permutations whose attributes are on the forbidden attributes list L.

5.2. *The tabu list*

In order to avoid generating a cycle (by returning to the same permutation after a small number of algorithm iterations), some attributes of every move are saved on a tabu list. It is operated according to the FIFO queue. By making the $s_j^r \in M(\pi)$ (generating from $\pi \in \Phi$ the permutation π_j^r) we write on the tabu list L of this move's attributes, the tuple $(\pi(r), j, F(\pi_j^r))$.

Suppose, that we analyze the move $s_l^k \in M(\beta)$ generating from $\beta \in \Phi$ the β_l^k permutation. If the tuple (r, j, Ψ), such that $\beta(k) = r$, $l = j$ and $F(\beta_l^k) \geq \Psi$ is on the L list, such a move is forbidden and removed from the $M(\beta)$ set. The only parameter of this list is its length, the number of the elements it contains. There are many realizations of the tabu list in the bibliography.

The basic version of the algorithm has been modified accordingly which made it possible to perform calculations for uncertain (fuzzy or random) times of works execution.

6. Algorithms stability

Let $\mathbf{p} = [p_{i,j}]_{n \times m}$ be (deterministic) jobs execution times for an instance of the **CPF** problem. By $D(\mathbf{p})$ we describe a set of examples of data generated from $\mathbf{p}$ by the disturbance of jobs execution times (i.e. elements from $\mathbf{p}$). The disturbance consists in random changes of $p_{i,j}$ values, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$.

We use the following notion:

A	-	an algorithm of solving **CPF** problem,
$\mathbf{p} = [p_{i,j}]_{n \times m}$	-	an instance of data (execution times) for the **CPF** problem,
$\pi_{\mathbf{p}}^A$	-	a solution (jobs permutation) determined by the algorithm A for the data $\mathbf{p}$,
$F(\pi_{\mathbf{p}}^A, \mathbf{d})$	-	a value of the cost function for the data $\mathbf{d}$ and a sequence of jobs execution (permutation) $\pi_{\mathbf{p}}^A$.

Let $\mathbf{p}$ be an instance of deterministic data and $D(\mathbf{p})$ a set of disturbed data. For an algorithm A and an instance of disturbed data $\mathbf{d}$

$$\delta(A, \mathbf{p}, \mathbf{d}) = \frac{F(\pi_{\mathbf{p}}^A, \mathbf{d}) - F(\pi_{\mathbf{d}}^A, \mathbf{d})}{F(\pi_{\mathbf{d}}^A, \mathbf{d})} \cdot 100\%. \tag{6.1}$$

This formula defines a percentage relative deviation of the cost function value for the $\mathbf{d}$ if jobs are executed in the sequence $\pi_{\mathbf{p}}^A$ and $\pi_{\mathbf{d}}^A$. By

$$\Delta(A, \mathbf{p}, D(\mathbf{p})) = \frac{1}{|D(\mathbf{p})|} \sum_{\mathbf{d} \in D(\mathbf{p})} \delta(A, \mathbf{p}, \mathbf{d}) \tag{6.2}$$

we define the ***stability of the best solution*** of an instance $\mathbf{p}$ determined by an algorithm A on the set of disturbed data $D(\mathbf{p})$.

Let Ω be a set of some (deterministic) data instances for the **CPF** problem. The ***algorithm stability*** A on the data set Ω

$$S(A, \Omega) = \frac{1}{|\Omega|} \sum_{\mathbf{p} \in \Omega} \Delta(A, \mathbf{p}, D(\mathbf{p}))). \tag{6.3}$$

6.1. *Computational experiments*

Presented in Section 5 tabu search algorithm was programmed in C++ and run on a personal computer with a 2.4 GHz processor. The computations were made on the basis of six groups of Taillard examples [21] for the problem of $F||C_{\max}$). Each group $n \times m$: 20×5, 20×10, 20×20, 50×5, 50×10, 50×20 contains 10 examples (a total of 60 examples). Weights of penalty function w_i were generated with the use of uniform distribution from the set {1,2,...,10}, whereas the required deadlines for the completion of structures were determined using the following procedure

Step 1: compute

$$\rho_i = \sum_{k=1}^{m} p_{i,k}, \quad i = 1, 2, \ldots, n.$$

Step 2: For each structure O_i designate the required completion moment

$$d_i = \lceil \rho_i(1 + 3\epsilon) \rceil.$$

Parameter ϵ is a realization of a random variable with uniform distribution over the interval [0,1]. A similar method of data generating is described in the works by Hasij and Rajendra [11]. A set of all 60 of deterministic examples is denoted by Ω . For each example of deterministic data there was a corresponding instance of data with fuzzy and probabilistic (normal distribution and the Erlang distribution) times of works execution established:

1 *Fuzzy times of works execution*

If $p_{i,j}$ ($i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$) is an instance of deterministic data for the **BFP** problem, then fuzzy jobs execution times $\widehat{p}_{i,j}$ are represented by a triple $(p^a_{i,j}, p^b_{i,j}, p^c_{i,j})$, where

$$p^a_{i,j} = \lceil p_{i,j} - p_{i,j}/3 \rceil \}, \quad p^b_{i,j} = p_{i,j}, \quad \text{and} \quad p^c_{i,j} = \lceil p_{i,j} + p_{i,j}/2 \rceil.$$

This set of data is denoted by $\widehat{\Omega}$.

2 *Random execution times of works*

(a) normal distribution:

- execution times of works $\tilde{p}_{i,j} \sim N(p_{i,j}, \alpha_{i,j} \cdot p_{i,j})$, where $\alpha_{i,j} = 0.2$,
- moments of completion of building structures $\tilde{C}_i \sim N(\mu_i, \sigma_i)$, where $\mu_i = \alpha_i \cdot (p_{i,1} + p_{i,2} +, \ldots, p_{i,m})$, $\alpha_i = 0.75$ and $\sigma_i = \beta_i \cdot \sqrt{p^2_{i,1} + p^2_{i,2} +, \ldots, p^2_{i,m}}$, $\beta_i = 0.15$,

(b) Erlangs distribution:

- times of works execution $\tilde{p}_{i,j} \sim E(p_{i,j}, 1)$,
- moments of completion of structures $\tilde{C}_i \sim E(\mu_i, \lambda_i)$, where $\mu_i = \alpha_i \cdot (p_{i,1} + p_{i,2} +, \ldots, p_{i,m})$, $\alpha_i = 0.75$ and $\lambda_i = \omega_i \cdot i$, $\omega_i = 0.8$.

A set of these data is denoted by $\tilde{\Omega}^N$ and $\tilde{\Omega}^E$.

Algorithms

Computational experiments were carried out on several versions of an algorithm described in Section 5:

1 *AD* – deterministic algorithm with the function of choice of element from the neighborhood (2.5), deterministic data Ω.

1. *AF* - fuzzy algorithm with fuzzy times of works execution and the function selecting an element from the neighborhood (4.2), fuzzy data $\widehat{\Omega}$.
2. AN_E, AN_{ED} – probabilistic algorithms with times of works execution normally distributed and function selecting elements from the neighborhood respectively with normal distribution, and the function selecting the elements from the neighborhood respectively (4.3) or (4.4), probabilistic data $\tilde{\Omega}^N$.
3. AE_E, AE_{ED} - probabilistic algorithms with times of works execution with the Erlang distribution and function selecting elements from the neighborhood respectively (4.3) or (4.4), probabilistic data $\tilde{\Omega}^E$.

Starting permutation.

Starting solution of algorithms, whose construction was based on the tabu search method, was determined with the use of the *NEH* algorithm [17]. The algorithm requires designating of a sequence of structures execution. Hence, they were sorted in a non-increasing order in reference to $\sum_{j=1}^{m} p_{i,j} - d_i$, $i = 1, 2, \ldots, n$.

Algorithms' parameters. Taking two examples of each group of deterministic the following parameters were determined:

- the length of tabu list, $\sqrt{n}$,
- the number of algorithms' iterations, n^2,
- parameter in the formula (4.4), $\theta = 0.1$.

6.2. *Computational results*

Since for the considered problem there are no comparative data in the literature, we must limit ourselves to comparison of only the results of the algorithms presented in the work. Firstly, the quality of solutions designated by individual algorithms was examined. The results of a deterministic algorithm *AD* and both probabilistic – AN_E and AE_E algorithms – were compared with the solutions designated by the *NEH* algorithm. For any example, let F^A be the value of the objective function of the solution designated by the algorithm $A \in \{AD, AF, AN_E, AE_E\}$. Then, $(F^{NEH} - F^A)/F^{NEH} \cdot 100\%$ is *percent relative error* (improvement) of the solution F^A in reference to the value of the solution F^{NEH} determined by the *NEH* algorithm.

Table 1 presents average δ_{aprd} and maximum δ_{mrpd} percentage improvement (relative errors) for the different groups of data.

Comparing the average relative error δ_{aprd}, it appears that regardless of the number of works and machines, the deterministic algorithm *AD* determines much better solutions than the algorithms with uncertain times of works execution. In this case, the mean improvements of the solutions of the algorithm *AD* is 8.53% and is definitely greater than the other algorithms. It can be seen specifically for the examples with a larger number of objects. Average maximum errors δ_{mrpd} have similar proportions. The results of two other probabilistic algorithms AN_{ED} and AE_{ED}, were not presented since they are similar to the results of algorithms AN_E and AE_E. The computation time of one algorithm, of all 60 examples did not exceed 5 seconds.

Table 1 – Improvement of the solutions designated by the *NEH* algorithm.

$n \times m$	AD		AF		AN_E		AE_E	
	δ_{aprd}	δ_{mrpd}	δ_{aprd}	δ_{mrpd}	δ_{aprd}	δ_{mrpd}	δ_{aprd}	δ_{mrpd}
20 × 5	5.19	9.54	3.27	8.67	3.57	7.15	4.17	9.17
20 × 10	6.37	9.26	4.86	6.25	4.01	10.27	4.16	8.65
20 × 20	10.02	14.08	8.74	11.18	6.13	11.82	4.54	10.44
50 × 5	7.64	11.63	6.51	9.27	6.31	11.34	5.32	8.59
50 × 10	11.83	16.37	9.83	9.13	6.09	9.76	4.99	8.39
50 × 20	9.07	12.08	8.11	14.92	7.25	12.81	5.81	9.87
Average	**8.53**	**12.16**	**6.88**	**9.90**	**5.56**	**10.52**	**4.83**	**9.18**

6.3. *The stability of algorithms*

In order to investigate the stability of algorithms there were sets of proper perturbed data generated. The basis was 60 examples of deterministic data from the set Ω.

Generating of perturbed data. Let φ be an example of deterministic data with times of works execution $\mathbf{p} = \{p_{i,j}\}_{m \times n}$. For the example φ there wer 100 examples of perturbed data generated - elements of the set $D(\varphi)$. If the example of the perturbed data is $\varphi' \in D(\varphi)$, then the perturbed times of works execution $p'_{i,j}$ were drawn in accordance with the uniform distribution, from the interval

$$[\max\{1, \lceil p_{i,j} - p_{i,j}/3 \rceil\}, \lceil p_{i,j} + p_{i,j}/2 \rceil].$$

Other elements of examples of φ and φ' (i.e. moments of works completion and penalty function coefficients) are the same.

In total there was 6000 examples of the disturbed data generated. They were then solved by the algorithm *AD*, whose solutions served as the basis for designation of the stability coefficient of the studied algorithms. The results of the conducted numerical experiments are given in Table 2.

Both fuzzy and probabilistic algorithms AN_E and AN_{ED} have a stability coefficient far smaller than the deterministic algorithm. For the fuzzy times of the works execution the fuzzy algorithm *AF* has the smallest stability coefficient 3.63% and it is almost twice as small as the size of 6.72% – the coefficient of a deterministic algorithm *AD*. The coefficient 3.63% of the algorithm *AF* results in the fact that the random disturbance of the execution times of works causes the deterioration of the value of the goal function (in reference to the solutions of the algorithm *AD*) by 3.63% by mean. Probabilistic algorithms with times of works execution normally distributed have also, although slightly, smaller stability coefficient than the coefficient of the algorithm *AD*.

Table 2 – Stability coefficient of $S(A, \Omega)$ algorithms.

$n \times m$	AD	AF	AN_E	AE_E	AN_{ED}	AE_{ED}
20 × 5	4.23	1.89	2.13	4.77	4.17	3.9
20 × 10	4.17	2.57	3.02	5.63	4.09	6.72
20 × 20	7.69	2.85	3.17	8.42	5.98	7.27
50 × 5	6.84	4.07	5.37	10.17	7.53	9.46
50 × 10	7.99	4.62	4.99	12.59	8.97	12.88
50 × 20	9.44	5.83	6.11	13.26	9.16	11.76
Average	**6.72**	**3.63**	**4.13**	**9.14**	**6.65**	**8.66**

The other two probabilistic algorithms AE_E and AE_{ED} have the greater stability coefficient. In summary, based on the obtained results, it can be concluded that the fuzzy algorithm is much more stable than the others. The solutions determined by the fuzzy algorithm are much less sensitive to possible random changes of the parameters of the problem. For the random times of the works execution the algorithm AN_E. Is much more stable. All computations were performed in less than one hour.

7. Case study

The investment project concerns the realization of a complex of twelve residential buildings ($n = 12$). The projects of the buildings are characterized by a similar set of construction works forming an orderly nine-element ($m = 9$) sequence of works beginning with earthworks and with finishing works ending. Based on the (National) Contractors Estimator (KNR, [12]) there were times (matrix p) of works execution estimated (in working days), Table 3. The KNR is a normative document containing individual material inputs of labor, materials and equipment necessary to create a cost estimate in the investment documentation.

Graphical representations of sample schedules are presented in Figs. 3 and 4. The numbers next to the points represent the moments (times) of completing the operation in the deterministic version of the problem under consideration. Two classical approaches to the jobs starting times were considered: to have minimal idle times between the same kind of works on objects (Fig. 3) and to have minimal idle times

Table 3 – Time duration of works (processes) on objects.

Work's number	Object's number											
	1	2	3	4	5	6	7	8	9	10	11	12
1	7	8	7	7	7	8	7	7	6	7	5	4
2	8	11	8	9	9	11	8	9	8	9	8	8
3	8	11	10	9	9	11	10	9	11	9	9	9
4	7	8	7	7	8	8	7	7	8	8	8	7
5	6	7	7	7	7	7	7	7	7	7	8	15
6	11	14	11	13	13	14	11	13	14	13	14	8
7	9	14	9	11	10	13	9	11	8	10	11	9
8	4	8	6	7	5	7	7	8	9	9	9	5
9	6	9	5	9	7	5	8	9	8	7	7	7

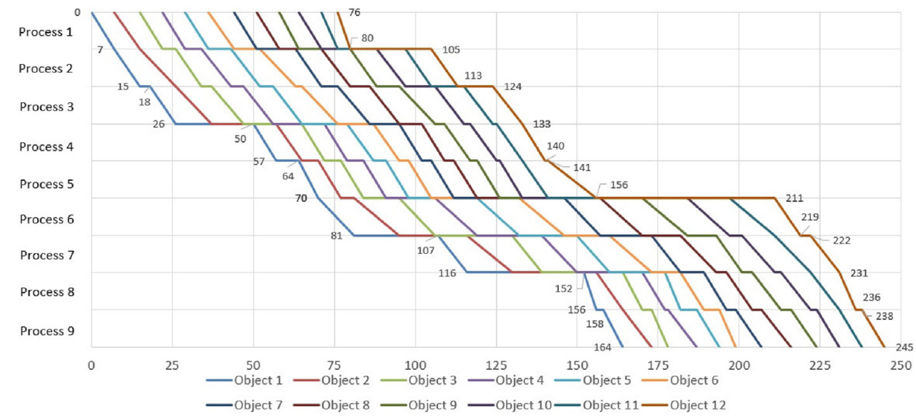

Fig. 3 – Case study schedule with minimal idle time between objects.

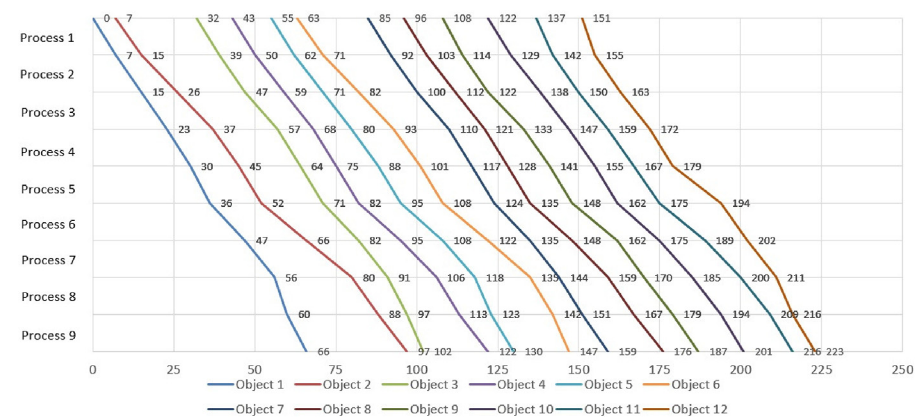

Fig. 4 – Case study schedule with minimal idle time between processes.

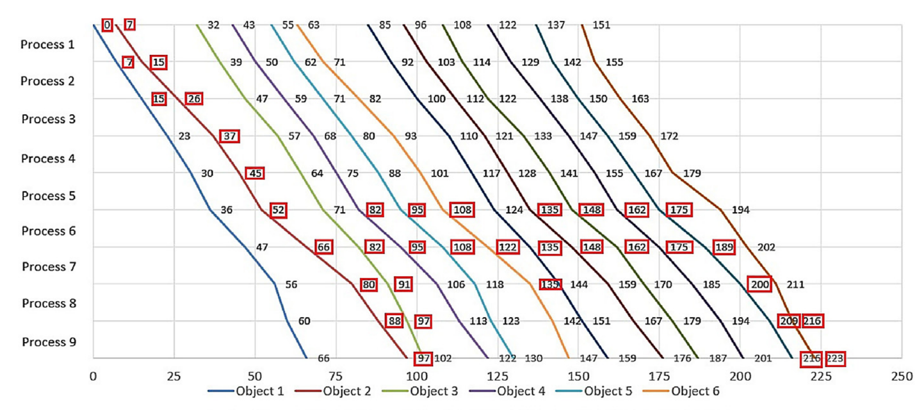

Fig. 5 – Case study schedule with minimal idle time between processes with critical path marked by red boxes.

between processes (Fig. 4), with the same permutation (solution) as objects order.

Fig. 3 presents an approach with a minimum wait time between works on objects. This means that after completing work on the site, Process will wait as soon as possible to start work on the next facility. In practice, this means delaying the commencement of works so as to minimize demigration of the brigades between objects. It is visible in Fig. 3 in the form of horizontal lines representing the waiting for the commencement of works on the site and after completion of works on the previous facility.

Figs. 4 and 5 present the schedule of works with continuous processes (i.e., works of construction brigades). Minimizing downtime de facto means here a glazed time of downtime, because the moments when commencing works on objects are delayed so as to obtain continuity of works for brigades (i.e., processes). The form of the critical path shown in Fig. 5 for this model is interesting. The start and end moments of works belonging to the critical path are placed in red rectangles. It can be noticed otherwise than for the classical pipeline system (where the trick passes ''right'' or ''down''), the critical path may "return" to previous processes, wandering somehow "upwards". This fact complicates the process of determining the critical path significantly and causes that classic algorithms cannot be used directly.

As described at the beginning of Section 6.1 there were moments of structures completion designated

$$\mathbf{d} = (210, 162, 196, 247, 96, 159, 211, 107, 287, 231, 192, 115)$$

and the weight of the penalty function for exceeding the completion time limits

$$\mathbf{w} = (7, 2, 6, 9, 5, 3, 6, 8, 4, 7, 1, 4).$$

Then, on the basis of the presented in Table 3 deterministic times of execution of works $[p_{i,j}]_{9\times 12}$, there were uncertain data (i.e. the uncertain times of execution of works) determined:

1. fuzzy (Table 4), $\hat{p}_{i,j}$ are represented by a triple $(p^a_{i,j}, p^b_{i,j}, p^c_{i,j})$,
2. probabilistic:
   - normal distribution $\tilde{p}_{i,j} \sim N(p_{i,j}, 0, 2{\cdot}p_{i,j})$, $i = 1, 2, \ldots, 9$, $j = 1, 2, \ldots, 12$,
   - Erlang distribution $\tilde{p}_{i,j} \sim E(p_{i,j}, 1)$, $i = 1, 2, \ldots, 9$, $j = 1, 2, \ldots, 12$.

There were computations performed for particular algorithms, for deterministic algorithms and with uncertain data and the following results (penalties for delays in the implementation of structures) $F(AD) = 1626$, $F(AF) = 1708$, $F(AN_E) = 1708$, $F(AE_E) = 1782$, $F(AN_{E+D}) = 1704$, $F(AE_{E+D}) = 1782$. Similarly, as in the case of the described in Section 6.2 computational experiments the best solution was determined by the deterministic algorithm AD. In this case, the penalty for tardiness is 1626. The worst appeared to be the solution determined by the algorithm AE_{E+D}.

Table 4 – Fuzzy times of works execution of builded objects.

1	2	3	4	5	6	7	8	9	10	11	12
Object's number											
(5,7,11)	(6,8,12)	(5,7,11)	(5,7,11)	(5,7,11)	(6,8,12)	(5,7,11)	(5,7,11)	(4,6,9)	(5,7,11)	(4,5,8)	(3,4,6)
(6,8,12)	(8,11,17)	(6,8,12)	(6,9,14)	(6,9,14)	(8,11,17)	(6,8,12)	(6,9,14)	(6,8,12)	(6,9,14)	(6,8,12)	(6,8,12)
(6,8,12)	(8,11,17)	(7,10,15)	(6,9,14)	(6,9,14)	(8,11,17)	(7,10,15)	(6,9,14)	(8,11,17)	(6,9,14)	(6,9,14)	(6,9,14)
(5,7,11)	(6,8,12)	(5,7,11)	(5,7,11)	(6,8,12)	(6,8,12)	(5,7,11)	(5,7,11)	(6,8,12)	(6,8,12)	(6,8,12)	(5,7,11)
(4,6,9)	(5,7,11)	(5,7,11)	(5,7,11)	(5,7,11)	(5,7,11)	(5,7,11)	(5,7,11)	(5,7,11)	(5,7,11)	(6,8,12)	(10,15,23)
(8,11,17)	(10,14,21)	(8,11,17)	(8,13,20)	(8,13,20)	(10,14,21)	(8,11,17)	(8,13,20)	(10,14,21)	(8,13,20)	(10,14,21)	(6,8,12)
(6,9,14)	(10,14,21)	(6,9,14)	(8,11,17)	(7,10,15)	(8,13,20)	(6,9,14)	(8,11,17)	(6,8,12)	(7,10,15)	(8,11,17)	(6,9,14)
(3,4,6)	(6,8,12)	(4,6,9)	(5,7,11)	(4,5,8)	(5,7,11)	(5,7,11)	(6,8,12)	(6,9,14)	(6,9,14)	(6,9,14)	(4,5,8)
(4,6,9)	(6,9,14)	(4,5,8)	(6,9,14)	(5,7,11)	(4,5,8)	(6,8,12)	(6,9,14)	(6,8,12)	(5,7,11)	(5,7,11)	(5,7,11)

Immunity to perturbances of data, designated by the individual algorithms solutions was tested as follows. There were drawn (in accordance with the uniform distribution) 6 out of 12 objects. For each of the object there was a random selection (uniform distribution in the interval [5;25]) the number p and the execution time of each work on this object was increased by p%. For the disturbed data, penalties for tardiness in execution of the structures are respectively: $F(AD) = 1894$, $F(AF) = 1856$, $F(AN_E) = 1810$, $F(AE_E) = 1907$, $F(AN_{E+D}) = 1863$, $F(AE_{E+D}) = 2017$. In this case, the minimum penalty is a solution of an algorithm AN, since $F(AN_E) = 1810$. The biggest increase, from 1626 to 1894, was noted in the cost of the solution determined by the deterministic algorithm AD. This fact confirms the thesis that the deterministic algorithms solutions are the best, but unfortunately very susceptible to disturbance of the data.

An example from the practice was presented above, on which the methodology of constructing the input data of the algorithm was demonstrated, i.e. the duration of the operation was blurred. However, the main result of the work is not a case study that only demonstrates the technology we propose, but a new innovative methodology for the construction of fault tolerant schedules by using algorithms based on a probabilistic model.

8. Comments and conclusions

The work presents the sum-cost construction scheduling problem with uncertain data represented by fuzzy numbers and random variables with normal or the Erlang distribution. These distributions described well the natural randomness with which we deal mostly in management practice. The new finding proposed in the work consists in introduce the new method of constructing schedules resistant to data disturbances assuming normal or Erlang distributions of uncertain operations durations. To solve the problems there was the algorithm used, whose design was based on the tabu search method. There were experiments conducted to investigate the computational stability of algorithms, i.e. the influence of disturbance parameters of the problem on the change of the value of the optimized criterion. The obtained results clearly show that the most stable is a probabilistic algorithm with times of works execution represented by random variables with normal distribution. The use of probability elements in adapting of tabu search methods can effectively solve large examples of uncertain data for many difficult practical optimization problems. Similar conclusions, resulting from studies concerning single-machine task scheduling problem with desired moments of tasks completion, are included in the work Bożejko et al. [4].

Funding

The paper was partially supported by the Wroclaw University of Science and Technology internal grant PWR-RUDN no. 45WB/00001/17 and National Science Centre of Poland, grant OPUS no. DEC 2017/25/B/ST7/02181.

REFERENCES

[1] D. Arditi, O.B. Tokdemir, K. Suh, Scheduling System for Repetitive Unit Construction Using Line-of-Balance Technology, Department of Civil and Architectural Engineering, Illinois Institute of Technology, Department of Civil Engineering, Honan University, Honam, South Korea, 2002.

[2] W. Bożejko, M. Wodecki, On the theoretical properties of swap multimoves, Oper. Res. Lett. 35 (2) (2007) 227–231.

[3] W. Bożejko, M. Uchroński, M. Wodecki, The new golf neighborhood for the flexible job shop problem, Procedia Comput. Sci. 1 (2010) 289–296.

[4] W. Bożejko, Z. Hejducki, M. Wodecki, Applying metaheuristic strategies in construction projects management, J. Civil Eng. Manag. 18 (5) (2012) 621–630.

[5] W. Bożejko, P. Rajba, M. Wodecki, One machine scheduling problems with random due date and processing times, Bull. Polish Acad. Sci. Tech. Sci. 65 (2) (2017) 219–231.

[6] S.-M. Chen, F.H. Griffis, P.-H. Chen, L.-M. Chang, Simulation and analytical techniques for construction resource planning and scheduling, Autom. Constr. 21 (2012) 99–113.

[7] A. Czarnigowska, A. Sobotka, Time–cost relationship for predicting construction duration, Arch. Civil Mech. Eng. 13 (4) (2013) 518–526.

[8] B.C. Dean, Approximation algorithms for stochastic scheduling problems, PhD thesis, MIT, 2005.

[9] K. El-Rayes, O. Moselhi, Optimal resource utilization for repetitive construction projects, J. Constr. Eng. Manag. ASCE 127 (1) (2001) 18–27.

[10] J. Grabowski, M. Wodecki, A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion, Comput. Oper. Res. 31 (2004) 1891–1909.

[11] S. Hasija, C. Rajendran, Scheduling in flowshop to minimize total tardiness of jobs, Int. J. Prod. Res. 42 (11) (2004) 2289–2301.

[12] KNR,;1; Regulation of the polish Minister of Infrastructure of 18 May 2004, Dz.U. No. 130, Item 1391.

[13] A. Jiang, A. Issa, M. Malek, Construction project cash flow planning using the Pareto optimality efficiency network model, J. Civil Eng. Manag. 17 (2011) 510–519.

[14] S.-S. Liu, C.-J. Wang, Optiminizing project selection and scheduling problems with time-dependent resourceconstructions, Autom. Constr. 20 (8) (2011) 1110–1119.

[15] S.-S. Liu, C.-J. Wang, Optimizing linear project scheduling with multi-skilled crews, Autom. Constru. 24 (2012) 16–23.

[16] C.-J. Mattila, A. Park, Comparison of linear scheduling model and repetitive scheduling method, J. Constr. Eng. Manag. 129 (1) (2003) 56–64.

[17] M. Nawaz, E.E. Enscore, I. Ham, A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, OMEGA 11 (1) (1983) 91–95.

[18] G. Onwubolu, D. Davendra, Scheduling flow shop using differential evolution algorithm, Eur. J. Oper. Res. 171 (2006) 674–692.

[19] C. Rajendran, H. Ziegler, Scheduling to minimize the sum of weighted flowtime and weighted tardiness of jobs in a flowshop with sequence-dependent setup times, Eur. J. Oper. Res. 147 (3) (2003) 513–522.

[20] B.R. Sarker, P.J. Egbelu, T.W. Liao, J. Yu, Planning and design models for construction industry: a critical survey, Autom. Constr. 22 (2012) 123–134.

[21] E. Taillard, Benchmarks for basic scheduling problems, Eur. J. Oper. Res. 64 (1993) 278–285.

[22] M. Wodecki, A block approach to earliness-tardiness scheduling problems, Int. J. Adv. Manuf. Technol. 40 (2009) 797–807.

[23] E. Vallada, R. Ruiz, G. Millea, Minimizing total tardiness in the m-machine flowshop problem: a review and evaluation heuristics and metaheuristics, Comput. Oper. Res. 35 (4) (2008) 1350–1373.