TECHNICAL TRANSACTIONS 8/2017
CZASOPISMO TECHNICZNE 8/2017

CIVIL ENGINEERING

DOI: 10.4467/2353737XCT.17.133.6884

Michał Krzemiński (mkrz@il.pw.edu.pl)
Institute of Building Engineering, Faculty of Civil Engineering, Warsaw University of Technology

Optimization of construction schedules with the assumed multi-tasking of working brigades

Optymalizacja harmonogramów budowlanych przy założonej wielozadaniowości brygad roboczych

Abstract

The paper presents a newly developed model aimed at minimizing the downtime in working brigades work. The developed method is dedicated to the construction planned using the stream method of work organization. The model focuses on the assumption of multi-tasking of working brigades. It has been possible to model the performance depending on the type of brigade and the type of work being performed. The article also shows an example illustrating the results of the model's operation.

Keywords: flowshop models, slack minimization, construction schedules

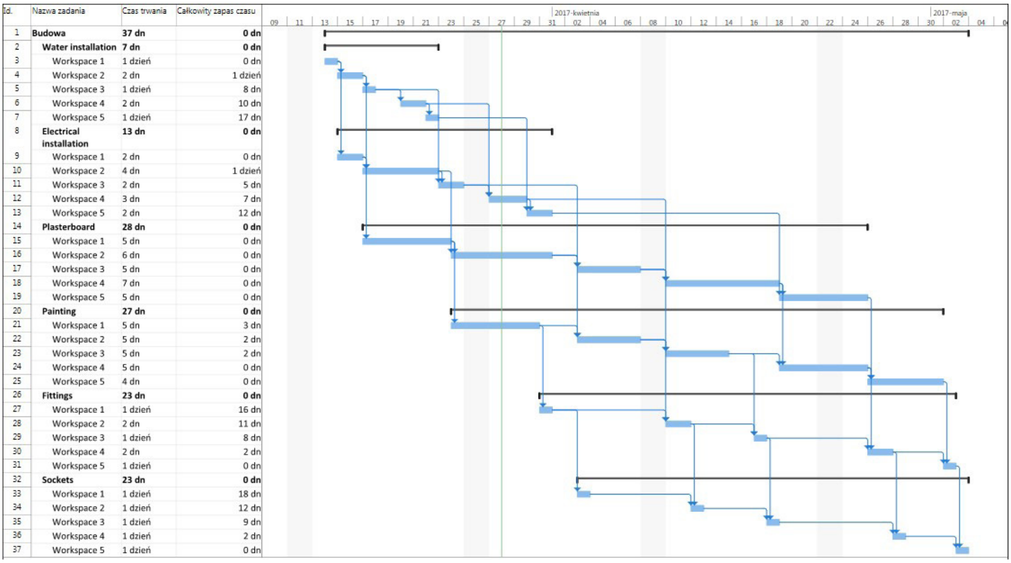
Streszczenie

W artykule zaprezentowano nowo opracowany model mający na celu minimalizację przestojów w pracach brygad roboczych. Opracowana metoda dedykowana jest dla budów planowanych z zastosowaniem potokowej metody organizacji robót. W modelu skupiono się na założeniu wielozadaniowości brygad roboczych. Przyjęto możliwość modelowania wydajności w zależności od typu brygady i rodzaju wykonywanego zadania. W artykule pokazano ponadto przykład ilustrujący wyniki działania modelu.

Słowa kluczowe: modele potokowe, minimalizacja przestoi, harmonogramy budowlane

1. Introduction

The issue of scheduling is the subject of many scientific papers. In the construction industry, there is also a lot of attention paid to this issue. However, this is quite a large area, even after limitation to the conditions of construction output. This article discusses the part of the issue related to scheduling the work organized using the stream method of work organization. The basic assumption is that a given field of operation can be divided into workspaces on which the schedule of operations should be performed. The following figure shows the schedule on the basis of which the new method of optimization proposed by the author will be discussed. For this schedule, the result of the optimization will also be shown.



Id.	Nazwa zadania	Czas trwania	Całkowity zapas czasu
1	**Budowa**	**37 dn**	**0 dn**
2	**Water installation**	**7 dn**	**0 dn**
3	Workspace 1	1 dzień	0 dn
4	Workspace 2	2 dn	1 dzień
5	Workspace 3	1 dzień	8 dn
6	Workspace 4	2 dn	10 dn
7	Workspace 5	1 dzień	17 dn
8	**Electrical installation**	**13 dn**	**0 dn**
9	Workspace 1	2 dn	0 dn
10	Workspace 2	4 dn	1 dzień
11	Workspace 3	2 dn	5 dn
12	Workspace 4	3 dn	7 dn
13	Workspace 5	2 dn	12 dn
14	**Plasterboard**	**28 dn**	**0 dn**
15	Workspace 1	5 dn	0 dn
16	Workspace 2	6 dn	0 dn
17	Workspace 3	5 dn	0 dn
18	Workspace 4	7 dn	0 dn
19	Workspace 5	5 dn	0 dn
20	**Painting**	**27 dn**	**0 dn**
21	Workspace 1	5 dn	3 dn
22	Workspace 2	5 dn	2 dn
23	Workspace 3	5 dn	2 dn
24	Workspace 4	5 dn	0 dn
25	Workspace 5	4 dn	0 dn
26	**Fittings**	**23 dn**	**0 dn**
27	Workspace 1	1 dzień	16 dn
28	Workspace 2	2 dn	11 dn
29	Workspace 3	1 dzień	8 dn
30	Workspace 4	2 dn	2 dn
31	Workspace 5	1 dzień	0 dn
32	**Sockets**	**23 dn**	**0 dn**
33	Workspace 1	1 dzień	18 dn
34	Workspace 2	1 dzień	12 dn
35	Workspace 3	1 dzień	9 dn
36	Workspace 4	1 dzień	2 dn
37	Workspace 5	1 dzień	0 dn

Fig. 1. Base schedule of the undertaking

Numerous papers talk about how these workspaces can be sequenced. For example, well-known task sequencing rules such as SPT (shortest processing time) and LPT (longest processing time) [5]. The models from the collection of artificial intelligence methods can also be used here [2, 6]. The initial schedule discussed has been developed using the specialized KASS software allowing for analysis of all possible workspace sequencing [3]. As the sequencing criteria, the shortest lead times and minimal downtime were assumed in the work of the brigades.

The following table shows the execution times for all activities on each workspace.

Table 1. Performance times

W1		Workspace				
		W2	W3	W4	W5	
Working brigade	Water installation	1	2	1	2	1
	Electrical installation	2	4	2	3	2
	Plasterboard	5	6	5	7	5
	Painting	5	5	5	5	4
	Fittings	1	2	1	2	1
	Sockets	1	1	1	1	1

It should be noted that with this sequence of workspaces, there was a downtime in the work, mainly the fifth and sixth brigade. It should be noted that further optimization work on the schedule is possible.

The simplest way to do this would be to increase the employment in the third and fourth brigade. Such an operation would increase their productivity and, consequently, shorten the performance of tasks [4].

Another way would be to use a multi-variant work execution. Such an approach would involve proposing alternatives for the work, for third and fourth workspace the shorter variants should be given and longer variants for fifth and sixth workspace. This solution would probably result in obtaining a schedule that is characterized by continuity of work [1].

This article focuses on finding a solution to the problem, assuming that neither the number of means of production can be employed nor multi-variant work performance is possible. It is also not possible to relocate the workers to another construction site in periods when they are not employed. Such restrictions may, in practice, result from the characteristics of the contractor.

This paper focuses on the assumptions of the possibility of schedule optimization using the possible multitasking of work brigades. The brigades with downtime can assist the preceding brigades in order to minimize these downtimes. It was assumed that the preceding brigades cannot assist the subsequent brigades because it seems reasonable to assume that when they finish working on their workspace they may move on to the next field of operation or construction site. The performance of the assisting brigades is shown in a special matrix in which 1 means that the brigade has the same performance and 0 that it cannot perform the work, the intermediate values determine the performance coefficient. Later in this article there is a detailed description of the newly developed algorithm and a numerical example illustrating its operation.

2. The description of the brigades' downtime elimination model

The presented algorithm is iterative. It was developed on the assumption that only the subsequent brigades can assist the preceding brigades. The following figure shows the block diagram of the algorithm:

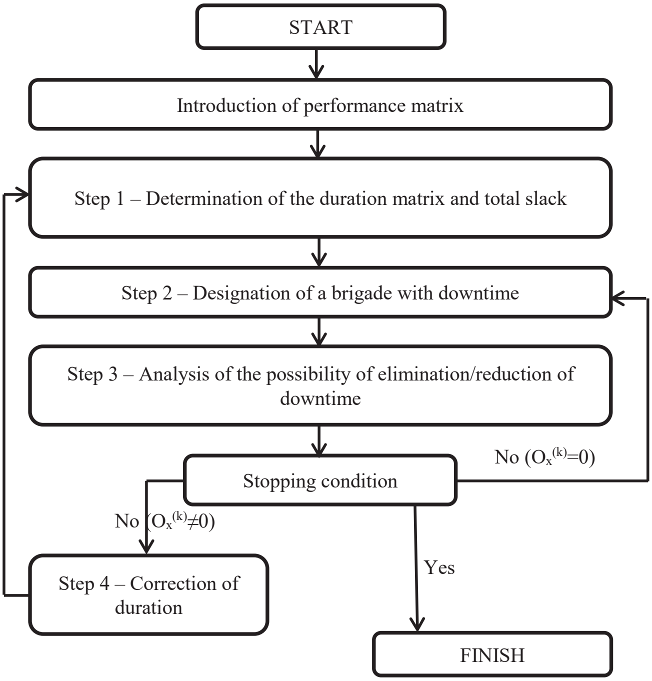

START
Introduction of performance matrix
Step 1 – Determination of the duration matrix and total slack
Step 2 – Designation of a brigade with downtime
Step 3 – Analysis of the possibility of elimination/reduction of downtime
Stopping condition
No (O$_x^{(k)}$=0)
No (O$_x^{(k)}$≠0)
Yes
Step 4 – Correction of duration
FINISH

Fig. 2. The algorithm of brigades' downtime elimination model

2.1. Introduction of performance matrix

The assumption of the method is that when a brigade supports another brigade, its performance on the executed process may be the same as that of the basic brigade or less. The performance is characterized by performance matrix.

$$EC_{p,i}\begin{bmatrix} ec_{1,1} & \cdots & ec_{1,i} \\ \vdots & \ddots & \vdots \\ ec_{p,1} & \cdots & ec_{p,i} \end{bmatrix}; p=1,\ ...,\ m;\ i=1,\ ...,\ m \tag{1}$$

where:
$ec_{p,i}$ – means the brigade performance when performing work on the process i,
p – number of assisted process,
i – (assisting) brigade number,
m – total number of working brigades.

The next steps in the algorithm will be described in the next section.

2.2. Step 1

Determination of the duration matrix and matrix of total slack. The matrices are determined by the analysis of the network model chosen to optimize the work schedule.

$$T_{i,j}^{(k)} = \begin{bmatrix} t_{1,1} & \cdots & t_{1,j} \\ \vdots & \ddots & \vdots \\ t_{i,1} & \cdots & t_{i,j} \end{bmatrix}; i=1, ..., m; j=1, ..., n \tag{2}$$

$$TF_{i,j}^{(k)} = \begin{bmatrix} tf_{1,1} & \cdots & tf_{1,j} \\ \vdots & \ddots & \vdots \\ tf_{i,1} & \cdots & tf_{i,j} \end{bmatrix}; i=1, ..., m; j=1, ..., n \tag{3}$$

where:
k – iteration number,
j – workspace number,
n – number of workspaces.

2.3. Step 2

The next step in the algorithm is to select the first brigade that has downtime in its work. To find such a brigade, use the following short algorithm:

1) $i = 1$,
2) Determine the value $tf_{i,1}^{(k)}$,
3) If the value $tf_{i,1}^{(k)} = 0$ then $i{:=}i{+}1$ and return to step 2 of the algorithm. If, however, $tf_{i,1}^{(k)} \neq 0$ proceed to 4 point of the algorithm,
4) The value i for which $tf_{i,1}^{(k)} \neq 0$ determines the number of the brigade which has downtime in its work.

This condition allows the brigades that have downtime to be determined. In the case in which the condition $tf_{i,1}^{(k)} = 0; i = 1, ..., m$ is met, we would have a schedule with no downtime.

Having selected the first brigade with downtime we can proceed to step 3 of the algorithm.

2.4. Step 3

From the previous step of the algorithm we know that the selected brigade has downtime at work. We know that these downtimes are caused by the prolonged working time of the preceding brigade. In this step of the algorithm, the first downtime should be identified, and then it should be checked if there is a possibility to eliminate it.

To identify the first downtime, use the following short algorithm:

1) The value is constant and determined on the basis of step 2 of the algorithm, $j = 1$,
2) Determine the value of $tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)}$,

3) If $tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)} = 0$ then $j := j+1$ and return to step 2 of the algorithm. If, however, $tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)} \neq 0$ proceed to step 4 of the algorithm,
4) The value $tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)} \neq 0$ for which determines the number of the workspace after which the brigade with number i has downtime.

The downtime is generated by the action of the preceding brigade. In order to eliminate the downtime, it is necessary to shorten the execution time of the preceding action by a value equal to $tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)}$.

In order to determine all possible variants of elimination of the downtime, a set should be developed containing the days in which the given downtime can be eliminated, i.e. the dates from the beginning of the preceding action to finish. The boundaries of the set can be determined by the following dependency:

$$lower\ boundry - ES_{i-1,j+1}^{(k)}, \tag{4}$$

$$upper\ boundry - EF_{i-1,j+1}^{(k)}, \tag{5}$$

Having a determined the set proceed to determine the sets in which the information is provided concerning the downtimes of successive brigades. Downtime in the work of the subsequent brigades can be used to reduce the working time of the preceding brigade. Index gives the number of the next variant. The following dependencies should be used to define the boundaries of variants:

$$lower\ boundry - EF_{i-1,j+1}^{(k)},\ i = i, \ldots, n;\ j = 1, \ldots, n-1, \tag{6}$$

$$upper\ boundry - ES_{i-1,j+1}^{(k)},\ i = i, \ldots, n;\ j = 1, \ldots, n-1, \tag{7}$$

Having determined all sets $B_x^{(k)}$ it should be determined:

$$D_x^{(k)} = A^{(k)} \cap B_x^{(k)};\ x = 1, \ldots, z, \tag{8}$$

where:
z – number of predefined variants.

Then the value $\left|D_x^{(k)}\right|$ should be determined, which specifies the cardinal number of the set $D_x^{(k)}$. The penultimate element of the third step is to determine possible variants of the shortening of the preceding operation. These variants are determined on the basis of the following dependency:

$$X \wedge Y \wedge Z \tag{9}$$

where:

$$X = \left(\left|D_x^{(k)}\right| \leq \left\lfloor \frac{A^{(k)}}{2} \right\rfloor \rightarrow \begin{cases} O_x^{(k)+} = \left|D_x^{(k)}\right| \\ O_x^{(k)-} = \left\lfloor \left|D_x^{(k)}\right| * ec_{p,i} \right\rfloor \end{cases} \right) \tag{10}$$

where:
$\left|A^{(k)}\right|$ – specifies the cardinal number of the set.

The first component of the formula is responsible for the case where the possible shortening is less than or equal to half the length of the shortened operation. Rounding down in the formula, which is encrypted as “$\lfloor \square \rfloor$” is intended to preserve the basic unit of the schedule as an integer.

The second component of the formula is responsible for the case in which it is possible to exceed the duration of the operation. Such shortening would make no sense because by shortening the working time of the first brigade, we would increase the working time of the second brigade which would not shorten the work on the workspace anyway.

$$Y=\left(\left(\left|D_x^{(k)}\right|\leq\left\lfloor\frac{A^{(k)}}{2}\right\rfloor\rightarrow\begin{cases}O_x^{(k)+}=\left|D_x^{(k)}\right|\\O_x^{(k)-}=\left\lfloor\left|D_x^{(k)}\right|*ec_{p,i}\right\rfloor\end{cases}\right)\wedge\left(O_x^{(k)-}=0\rightarrow O_x^{(k)+}=0\right)\right)\qquad(11)$$

The second component contains two elements. The first is responsible for shortening the working time of the assisted brigade and increasing the working time of the assisting brigade. The second, responsible for elimination of the case of prolonging the working time of the assisted brigade without shortening the time of the assisting brigade.

The task of the last dependency component is to eliminate the case in which there will be an excessive reduction in working time of the assisted brigade due to the operation of the mathematical model. Such shortening would unnecessarily involve the assisting brigade.

$$Z=\left(O_x^{(k)-}>\left(tf_{i,j}^{(k)}-tf_{i,j+1}^{(k)}\right)\rightarrow\begin{cases}O_x^{(k)+}=\left\lceil\dfrac{\left(tf_{i,j}^{(k)}-tf_{i,j+1}^{(k)}\right)}{ec_{p,i}}\right\rceil\\O_x^{(k)-}=tf_{i,j}^{(k)}-tf_{i,j+1}^{(k)}\end{cases}\right)\qquad(12)$$

Rounding up in the formula ($\lceil \square \rceil$) is intended to preserve the basic unit of the schedule as an integer.

The last element of step three is choosing the optimal variant. In order to choose the best option, the following rules should be observed:

- A variant that allows for shortening the preceding action as quickly as possible should be chosen.
- If several variants of shortening allow for shortening the preceding action by the same value, then the one with the highest performance coefficient should be chosen.
- If, however, there is a case in which several variants are equivalent for shortening and have the same performance coefficient, there should be chosen the variant in which the assisting brigade has a smaller index , the brigade that occurs earlier.

There is also the case where no variant is available to shorten the brigade downtime. In that case, go back to the beginning of step 3 of the algorithm and look for the next downtime at work and repeat the third step of the algorithm.

2.5. Stopping condition

It is also possible to schedule a pattern in which, in the course of the work of the whole brigade no downtime can be shortened, in which case the downtime of the next brigade should be reconsidered, that is, the one with increased by one index and return to the second point of the algorithm.

If and are achieved and i no shortening occurs, it means that no further optimization of the schedule is possible and the model must be discontinued.

If the shortening occurs, then proceed to the fourth step of the algorithm.

2.6. Step 4

In the fourth step, the duration of activities should be adjusted, the adjustment is performed using the following dependency:

$$t_{i-1,j+1}^{(k+1)} = t_{i-1,j+1}^{(k)} - O_x^{(k)-} \tag{13}$$

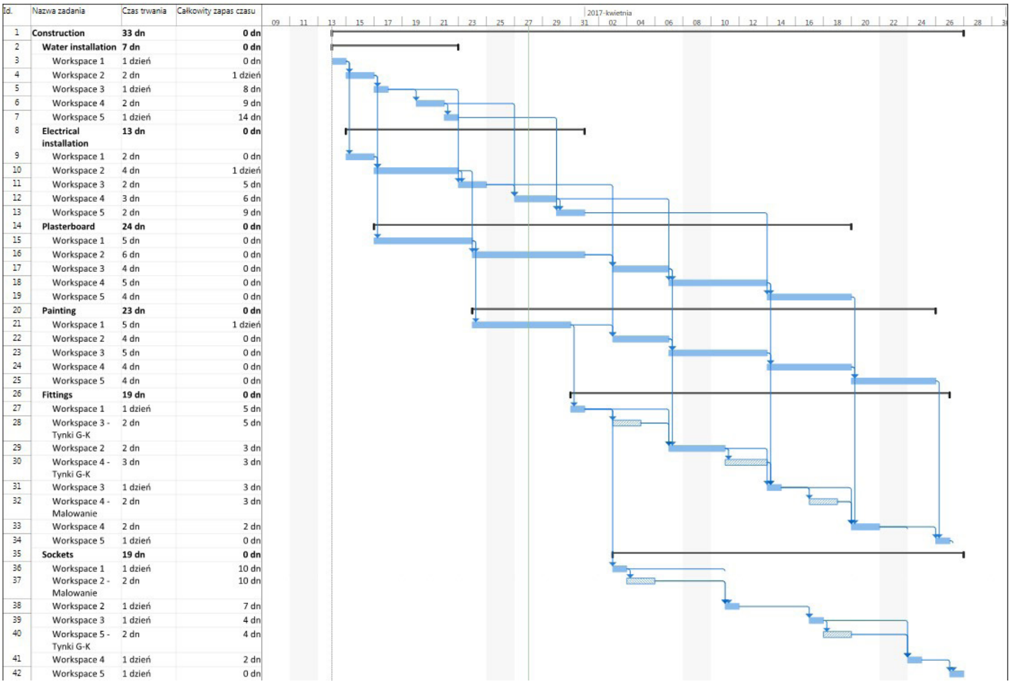
Based on the selected $O_x^{(k)+}$ the working time of the brigade on the basis of which the set $B_x^{(k)}$ was determined should also be added.

3. Application examples

The optimized schedule is described in the introduction to the article. Below, the schedule after optimization is presented.

The hatched actions indicate an additional work of assisting brigades. The description contains information on which field they work in and what activities they perform.

The initial total time is 37 days, after optimization 33 days, the shortening was 11%. The total initial downtime is 37 days, after optimization 16 days, the decrease was 57%. The results obtained can therefore be considered as satisfactory.



Id.	Nazwa zadania	Czas trwania	Całkowity zapas czasu
1	**Construction**	**33 dn**	**0 dn**
2	**Water installation**	**7 dn**	**0 dn**
3	Workspace 1	1 dzień	0 dn
4	Workspace 2	2 dn	1 dzień
5	Workspace 3	1 dzień	8 dn
6	Workspace 4	2 dn	9 dn
7	Workspace 5	1 dzień	14 dn
8	**Electrical installation**	**13 dn**	**0 dn**
9	Workspace 1	2 dn	0 dn
10	Workspace 2	4 dn	1 dzień
11	Workspace 3	2 dn	5 dn
12	Workspace 4	3 dn	6 dn
13	Workspace 5	2 dn	9 dn
14	**Plasterboard**	**24 dn**	**0 dn**
15	Workspace 1	5 dn	0 dn
16	Workspace 2	6 dn	0 dn
17	Workspace 3	4 dn	0 dn
18	Workspace 4	5 dn	0 dn
19	Workspace 5	4 dn	0 dn
20	**Painting**	**23 dn**	**0 dn**
21	Workspace 1	5 dn	1 dzień
22	Workspace 2	4 dn	0 dn
23	Workspace 3	5 dn	0 dn
24	Workspace 4	4 dn	0 dn
25	Workspace 5	4 dn	0 dn
26	**Fittings**	**19 dn**	**0 dn**
27	Workspace 1	1 dzień	5 dn
28	Workspace 3 - Tynki G-K	2 dn	5 dn
29	Workspace 2	2 dn	3 dn
30	Workspace 4 - Tynki G-K	3 dn	3 dn
31	Workspace 3	1 dzień	3 dn
32	Workspace 4 - Malowanie	2 dn	3 dn
33	Workspace 4	2 dn	2 dn
34	Workspace 5	1 dzień	0 dn
35	**Sockets**	**19 dn**	**0 dn**
36	Workspace 1	1 dzień	10 dn
37	Workspace 2 - Malowanie	2 dn	10 dn
38	Workspace 2	1 dzień	7 dn
39	Workspace 3	1 dzień	4 dn
40	Workspace 5 - Tynki G-K	2 dn	4 dn
41	Workspace 4	1 dzień	2 dn
42	Workspace 5	1 dzień	0 dn

Fig. 3. Project schedule after optimization

4. Summary

The article presents a newly developed method of optimizing construction schedules, performed assuming the organization by stream method. The work itself did not focus on sequencing the tasks, it was assumed that this issue had already been resolved earlier. The task of the new algorithm is to assign the work to brigades with downtimes that will reduce their total number. The developed model allows for the introduction of performance dependencies. The model, in its premise, first assigns the work to the brigades with the highest efficiency. The developed model can be successfully used in the construction industry. Further work is planned to verify the assumptions that have been adopted in the model.

References

[1] Biruk S., Jaśkowski P., *Scheduling construction of multi-object projects: problem of works continuity in consecutive units,* Zeszyty Naukowe WSOWL, 3 (157), 2010, 340–349.

[2] Hejducki Z., *Scheduling Model of Construction Activity with Time Couplings,* Journal of Civil Engineering and Management, 9:4, 2003, 284–291.

[3] Krzemiński M., *Use of the KASS Program in Scheduling,* Technical Transaction, 2-B(6), 2014, 217–224.

[4] Marcinkowski R., *Metody rozdziału zasobów realizatora w działalności inżynieryjno-budowlanej,* WAT, Warszawa 2002.

[5] Pinedo M.L., *Scheduling: Theory, Algorithms, and Systems,* Springer, 2012.

[6] Rogalska M., Bożejko W., Hejducki Z., *Time/Cost Optimization Using Hybrid Evolutionary Algorithm in Construction Project Scheduling,* Automotion in Construction, 18, 2008, 24–31.