DE GRUYTER OPEN

DOI: 10.1515/ace-2017-0037

ARCHIVES OF CIVIL ENGINEERING
Vol. LXIII ISSUE 4 2017



# OPTIMIZATION OF WORK SCHEDULES EXECUTED USING THE FLOW SHOP MODEL, ASSUMING MULTITASKING PERFORMED BY WORK CREWS

**M. KRZEMIŃSKI**[1]

The problem described in this paper deals with construction project scheduling for multi-object projects. The use of the flow shop model allows for quick execution through optimal, simultaneous means of production. The correct order of tasks according to assumed criteria is the subject of a lot of research. In this paper, we have focused on the aspect of multi-project work crews. In practice, we often observe cases where overburdened crews are supported by ones with downtime. This paper presents a newly developed model used to smoothe out schedules in terms of maintaining crew work continuity. A full description of the algorithm and example usage are presented.

*Keywords*: scheduling, flow shop models, brigade multitasking, job continuity

## 1. INTRODUCTION

The scheduling of multi-object projects is the subject of many research papers in the sphere of construction project engineering. The fundamental assumption is the planning of work by using the flow shop model, which means we are dealing with crews performing work sequentially and transitioning between individual work sites. The first step that must be taken, is the creation of the base schedule. In order to perform the optimization, scheduling of tasks must be done to set the right order of crew movement between work sites, thus set the order of work units. This part of the optimization can be performed using time couplings [2,3] by using time-cost ordering criteria.

[1] PhD., Eng., Warsaw University of Technology, Faculty of Civil Engineering, Al. Armii Ludowej 16, 00-637 Warsaw, Poland, e-mail: m.krzeminski@il.pw.edu.pl

For example, it is also possible to order using the time-cost criterion[6], or even the travelling salesman problem [1]. Factory models may be used here [7] or even artificial intelligence methods [8]. Attempts to solve this problem using a full review technique were undertaken by the author in earlier works [5].

The use of earlier mentioned methods does not guarantee simultaneous continuity of work for crews on different work fronts. With sufficiently unfavorable input data, resulting from construction technological dependencies, it is still possible for works to stop. The simplest way to eliminate such stops, is by changing the scale of means of production in order to speed up processes. However, it may occur that such a move is impossible due to factors such as limited infrastructure to support an increase in employment, or technological barriers.

Earlier mentioned methods are based on the assumption of single-tasking by work crews, meaning only one type of task is assigned to a given crew. However, in practice it can often be observed that when a crew is halting the project, a held up crew will join in to help. Thus it is the author's opinion that it is worth developing a model allowing idle crews to have additional tasks assigned when setting up the work schedule.

## 2. THE DESCRIPTION OF THE MODEL THAT ELIMINATES CREW DOWNTIME

### 2.1. ALGORITHM TO ELIMINATE DOWNTIME

The flowchart below presents the algorithm to eliminate downtime for work crews. The assumption of this method is that when a work crew is supporting another, its efficiency on a given task can be the same or less than that of the base crew. The efficiency is described by $EC_{p,i}$

$$EC_{p,i} = \begin{bmatrix} ec_{1,1} & \cdots & ec_{1,i} \\ \vdots & \ddots & \vdots \\ ec_{p,1} & \cdots & ec_{p,i} \end{bmatrix}; p = 1, \dots, m; i = 1, \dots, m \tag{2.1}$$

where:

$ec_{p,i}$ – signifies the efficiency of crew $p$ when performing work on process $i$,

$p$ – number of process being supported,

$i$ – crew number (supporting),

$m$ – total number of work crews.

The following figure number 1 presents the algorithm of the proposed model.

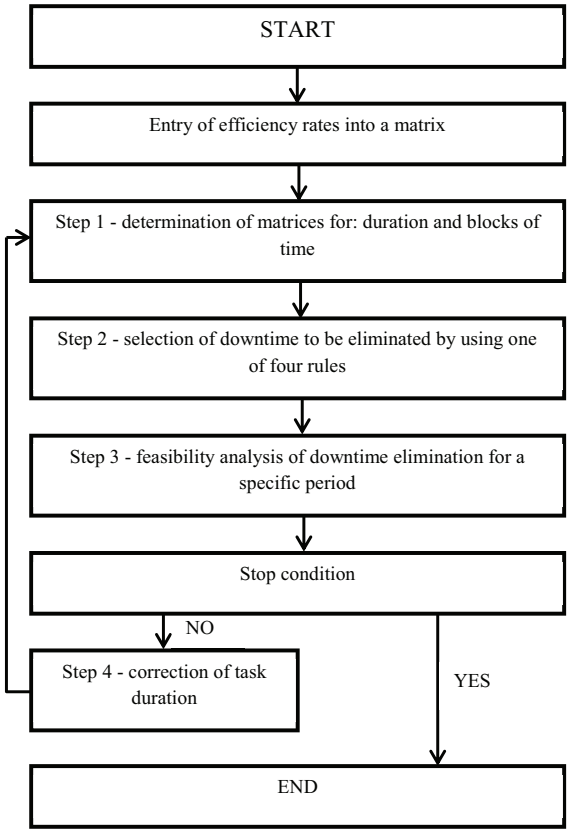


Fig. 1. Algorithm to eliminate downtime

The presented algorithm has an iterative quality. Further steps of this algorithm will be described later on in this chapter

## 2.2. STEP 1

A matrix of task duration and a matrix of time stores must be determined. Matrices are determined based on analysis of a network model selected to optimize the job schedule. The model is analyzed by the CPM method. It is assumed that the basic unit in this schedule is one day.

$$T_{i,j}^{(k)} = \begin{bmatrix} t_{1,1} & \cdots & t_{1,j} \\ \vdots & \ddots & \vdots \\ t_{i,1} & \cdots & t_{i,j} \end{bmatrix}; i = 1, \dots, m; j = 1, \dots, n \tag{2.2}$$

$$TF_{i,j}^{(k)} = \begin{bmatrix} tf_{1,1} & \cdots & tf_{1,j} \\ \vdots & \ddots & \vdots \\ tf_{i,1} & \cdots & tf_{i,j} \end{bmatrix}; i = 1, \dots, m; j = 1, \dots, n \tag{2.3}$$

where:

$T_{i,j}^{(k)}$ – matrix of task duration,

$TF_{i,j}^{(k)}$ – matrix of total float,

$k$ – number of iterations,

$j$ – lot number,

$n$ – number of work lots.

## 2.3. STEP 2

The goal of the second step of the algorithm is the extraction of work crew stoppage, which needs to be eliminated. In this presented research it is proposed that four rules can be used to eliminate crew idle time

### 2.3.1. RULE 1

It is accepted in this rule that the schedule will be searched "top to bottom" and "left to right." This is the most commonly used assumption when searching matrices, for example when searching for the highest value. In order to identify a crew with idle time, the algorithm below should be used:

1. $i = 1$,
2. Set the value of $tf_{i,1}^{(k)}$,
3. If the value of $tf_{i,1}^{(k)} = 0$ then $i \coloneqq i + 1$ needs to return to step 2 of the the algorithm.
4. Value $i$ for which $tf_{i,1}^{(k)} \neq 0$ returns the crew number with idle time.

We learn in this step of the algorithm that the selected crew has downtime. We know that downtime is caused by work taking long by the preceding crew. Having a crew with downtime identified, we must also identify its first stop. In order to accomplish this, the author recommends using the following short algorithm:

1. The value of $i$ is constant and is set by the previous short algorithm, $j = 1$,
2. Set 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 $j \coloneqq j + 1$ and return to step 2 of the algorithm.
4. The value $j$ for which $tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)} \neq 0$ produces the project number after which crew number $i$ has downtime.

### 2.3.2. RULE 2

The crew selection is done using a matrix of overall time stores. The largest period of downtime will be selected first. Crew selection needs to be done based on this dependency:

$$\mathrm{tf}_{\mathrm{i,j}}^{(\mathrm{k})} - \mathrm{tf}_{\mathrm{i,j+1}}^{(\mathrm{k})} = \max_{\mathrm{i,j}} \left[\mathrm{tf}_{\mathrm{i,j}}^{(\mathrm{k})} - \mathrm{tf}_{\mathrm{i,j+1}}^{(\mathrm{k})}\right]; \mathrm{i} = 1, \dots, \mathrm{m}; \mathrm{j} = 1, \dots, \mathrm{n} \tag{2.4}$$

If a few periods of downtime have the same length in time, the first one found will be selected. For the selected$\mathrm{tf}_{\mathrm{i,j}}^{(\mathrm{k})} - \mathrm{tf}_{\mathrm{i,j+1}}^{(\mathrm{k})}$ the value j gives the project number after which crew i has downtime.

### 2.3.3. RULE 3

This is another rule that is similar to the one described above. Similarly as with the previous crew, the selection is performed using matrices of overall time stores. The smallest period of downtime will be selected first. Crew selection needs to be performed using the formula below:

$$tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)} = \min_{i,j} \left[\left(tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)}\right) > 0\right]; i = 1, \dots, m; j = 1, \dots, n \tag{2.5}$$

f downtime periods will be of the same length, the first one found should be selected. For the chosen $tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)}$ value $j$ sets the project number after which crew $i$ has a period of downtime.

### 2.3.4. RULE 4

This rule is based on the assumption that periods of downtime will be chosen randomly. The rule depends on the approach that a random downtime period is selected from the total pool of downtime

periods, then moves onto the next step of the algorithm. Thus, one must randomly select the following

$$tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)} > 0 \tag{2.6}$$

For selected $tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)}$ value $j$ sets the project number after which crew $i$ has a period of downtime.

## 2.4. STEP 3

The period of downtime is generated by activity being performed by a preceding crew. In order to eliminate this downtime, time to execute a task must be shortened by a value equal to $tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)}$.

In order to determine all possible variants of downtime elimination, it is necessary to create collection $A^{(k)}$containing days during which downtime can be eliminated, that is the start date of the preceding task to its completion. Boundaries of this collection can be set with the using the following dependency:

$$Lower\ boundary = ES_{i-1,j+1}^{(k)}, \tag{2.7}$$

$$Upper\ boundary = EF_{i-1,j+1}^{(k)}, \tag{2.8}$$

Having collection $A^{(k)}$ populated, we must move onto populating set $B_x^{(k)}$ where information information about subsequent periods of downtime are contained. Downtime information for remaining work crews can be used to reduce work time for the preceding crew. Index x tells us about the identification number of the next variant. In order to set variant boundaries, the following dependencies should be used:

$$Lower\ boundary - EF_{i,j}^{(k)}; i = i, \dots, n;\ j = 1, \dots, n-1, \tag{2.9}$$

$$Upper\ boundary - ES_{i,j+1}^{(k)}; i = i, \dots, n;\ j = 1, \dots, n-1, \tag{2.10}$$

Having all collections $B_x^{(k)}$ set, the following needs to be determined:

(2.11) $$D_x^{(k)} = A^{(k)} \cap B_x^{(k)}; x = 1, \dots, z,$$

where:

$z$ – the number of earlier set variants.

Consequently, the value of $\left|D_x^{(k)}\right|$ must be set describing the cardinal number of collection $D_x^{(k)}$. The penultimate element of the third step is the designation of possible variants of shortening the preceding task. These variants are set by the dependency below:

(2.12) $$X \wedge Y \wedge Z$$

where:

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

where:

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

The first term of the formula is responsible for a situation where the possible time reduction is less than or equal to half of the time length of a task being shortened. The use of rounding down ($\lfloor \ \rfloor$) has the purpose of keeping the basic unit of the schedule as an integer.

The second term of the formula is responsible for the situation where a length of time is longer than the task duration. Such an operation does not make sense because we would stretch out work time for the supporting crew without shortening work time for the crew being supported.

(2.14) $$Y = \left( \left( \left|D_x^{(k)}\right| > \left\lfloor \frac{\left|A^{(k)}\right|}{2} \right\rfloor \rightarrow \begin{cases} ec_{p,i} = 1 \rightarrow O_x^{(k)+} = \left\lfloor \frac{\left|D_x^{(k)}\right|}{2} \right\rfloor; O_x^{(k)-} = \left\lfloor \left\lfloor \frac{\left|D_x^{(k)}\right|}{2} \right\rfloor * ec_{p,i} \right\rfloor \\ ec_{p,i} \neq 1 \rightarrow O_x^{(k)+} = \left\lceil \frac{\left|D_x^{(k)}\right|}{2} \right\rceil; O_x^{(k)-} = \left\lfloor \left\lceil \frac{\left|D_x^{(k)}\right|}{2} \right\rceil * ec_{p,i} \right\rfloor \end{cases} \right) \wedge \left( O_x^{(k)-} = 0 \rightarrow O_x^{(k)+} = 0 \right) \right)$$

Two elements are contained in the second term. The first one is responsible for shortening the duration of work for the crew being supported and lengthening the duration of work for the supporting crew. The second element is responsible for the elimination of a case where the duration of work of the supporting is extended without shortening the duration of work of the supported crew.

The role of the last term of the dependency is the case where work of the supported crew is excessively shortened as a result of the mathematical model. Such a shortening of work time would cause unnecessary involvement by the supporting crew.

$$Z = \left( O_x^{(k)-} > \left( tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)} \right) \rightarrow \begin{cases} O_x^{(k)+} = \left\lceil \frac{\left( tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)} \right)}{ec_{p,i}} \right\rceil \\ O_x^{(k)-} = tf_{i,j}^{(k)} - tf_{i,j+1}^{(k)} \end{cases} \right) \tag{2.15}$$

The use of rounding up (⌈ ⌉) has the purpose of keeping the base unit of the schedule as an integer.

The last element of the third step is the selection of the optimal variant. In order to choose the best one, the following rules must be followed:

- The variant that allows a task be shortened the most should be chosen
- If a few variants shorten the preceding task by the same amount, the one with the highest efficiency rate should be selected
- However, if a situation arises where variants are equivalent in terms of both time savings and efficiency, a variant where the supporting crew has a lower index i, meaning the crew that occurs first

If is also possible that a situation arises where no option to shorten a crew's downtime is produced. In that case, it is necessary to return to the third step of the algorithm and search for the next period of downtime. All operations from the step 3 of the algorithm must then be performed on this downtime period.

## 2.5. STOP CONDITION

There is also a situation where a schedule cannot have any periods of downtime eliminated for a particular crew. In that instance the next crew, with an index number $i$ increased by 1, should start being evaluated by returning to the second step of the algorithm.

If $i = m$ is reached and $j = n$ and no time optimizations are found, that means that no further optimization is possible and the algorithm must be stopped.

If however an opportunity to shorten a task is yielded, it is necessary to move onto step 4 of the algorithm.

## 2.6. STEP 4

Task duration adjustments occur during the 4th step of the algorithm. These adjustments are performed using the dependency below:

$$t_{i-1,j+1}^{(k+1)} = t_{i-1,j+1}^{(k)} - O_x^{(k)-}, \tag{2.16}$$

The task $ADt$ must be added to the network model and its length will be equal to $O_x^{(k)+}$. The task must be put in place of period of downtime $B_x^{(k)}$.

$$ADt = O_x^{(k)+}. \tag{2.17}$$

If the condition "X" was used in order to set the length of time being shortened, it is necessary to check an additional condition based on the new network model:

$$EF\left(t_{i-1,j+1}^{(k+1)}\right) < EF(ADt), \tag{2.18}$$

If it is true, the reduction must be set based on the condition "Y" and compared with other variants. With the best variant being selected proceed to the beginning of step 4, calculate dates, and rebuild the network model.

## 3. EXAMPLES OF USAGE

In the table below, task durations are shown on following workspaces. The sorting was achieved by using KASS v.2.2 software. [4]

Table 1. Input data based on KASS v.2.2

| | Workspace 1 | Workspace 2 | Workspace 3 | Workspace 4 |
|---|---|---|---|---|
| Plaster board | 6 | 5 | 6 | 5 |
| Cover strips | 3 | 2 | 3 | 2 |
| Windows | 1 | 1 | 1 | 1 |
| Doors | 1 | 1 | 1 | 1 |

The initial schedule is featured on the illustration below. It will be subjected to optimization with the use of the earlier mentioned model.

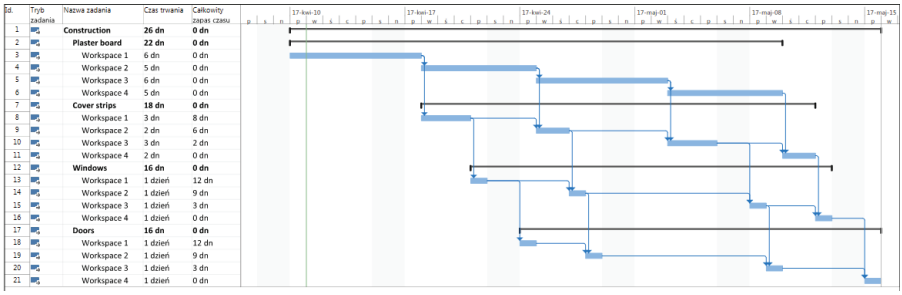


Fig. 2. Initial work schedule

Before we move onto the next step of the algorithm, we must define the matrix of efficiency rates $EC_{p,i}$, which is shown below:

$$EC_{p,i} = \begin{bmatrix} 1 & 0{,}8 & 0{,}7 & 0{,}7 \\ 0 & 1 & 0{,}7 & 0{,}7 \\ 0 & 0 & 1 & 0{,}8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Next, we can move onto the first step of the first iteration of the algorithm.

Iteration 1

Step 1

$$T_{i,j}^{(1)} = \begin{bmatrix} 6 & 5 & 6 & 5 \\ 3 & 2 & 3 & 2 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$TF_{i,j}^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 8 & 6 & 2 & 0 \\ 12 & 9 & 3 & 0 \\ 12 & 9 & 3 & 0 \end{bmatrix}$$

Step 2

Rule 1

$$tf_{2,2}^{(1)} = 6$$

$$tf_{2,2}^{(1)} - tf_{2,3}^{(1)} = 6 - 2 = 4$$

Step 3

$$A^{(1)} = \{12,13,14,15,16,17\}$$

$$B_1^{(1)} = \{10,11\};\ B_2^{(1)} = \{14,15,16,17\};\ B_3^{(1)} = \{21,22\}$$

$$B_4^{(1)} = \{11,12,13\};\ B_5^{(1)} = \{15,16,17,18,19,20\};\ B_6^{(1)} = \{21,22,23\}$$

$$B_7^{(1)} = \{12,13,14\};\ B_8^{(1)} = \{16,17,18,19,20,21\};\ B_9^{(1)} = \{23,24,25\}$$

$$D_1^{(1)} = \{\ \};\ D_2^{(1)} = \{14,15,16,17\};\ D_3^{(1)} = \{\ \}$$

$$D_4^{(1)} = \{12,13\};\ D_5^{(1)} = \{15,16,17\};\ D_1^{(1)} = \{\ \}$$

$$D_7^{(1)} = \{12,13,14\};\ D_8^{(1)} = \{16,17\};\ D_9^{(1)} = \{\ \}$$

$$\left(\left|D_2^{(1)}\right| > \left\lfloor\frac{\left|A^{(1)}\right|}{2}\right\rfloor \rightarrow \begin{cases} ec_{p,1} = 1 \rightarrow Not\ applicable \\ ec_{1,2} \neq 1 \rightarrow O_2^{(1)+} = \left\lceil\frac{\left|D_2^{(1)}\right|}{2}\right\rceil = 2;\ O_2^{(1)-} = \left\lfloor\left\lceil\frac{\left|D_2^{(1)}\right|}{2}\right\rceil * ec_{1,2}\right\rfloor = 1 \end{cases}\right) \wedge$$

$$\left(O_2^{(1)-} = 0 \rightarrow O_2^{(1)+} = 0\right)$$

$$\left|D_4^{(1)}\right| \leq \left\lfloor\frac{\left|A^{(1)}\right|}{2}\right\rfloor \rightarrow \begin{cases} O_4^{(1)+} = \left|D_4^{(1)}\right| = 2 \\ O_4^{(1)-} = \left|\left|D_4^{(1)}\right| * ec_{1,3}\right| = \lfloor 2 * 0{,}7 \rfloor = 1 \end{cases}$$

$$\left|D_5^{(1)}\right| \leq \left\lfloor\frac{\left|A^{(1)}\right|}{2}\right\rfloor \rightarrow \begin{cases} O_5^{(1)+} = \left|D_5^{(1)}\right| = 3 \\ O_5^{(1)-} = \left|\left|D_5^{(1)}\right| * ec_{1,3}\right| = \lfloor 2 * 0{,}7 \rfloor = 2 \end{cases}$$

$$\left|D_7^{(1)}\right| \leq \left\lfloor\frac{\left|A^{(1)}\right|}{2}\right\rfloor \rightarrow \begin{cases} O_7^{(1)+} = \left|D_7^{(1)}\right| = 3 \\ O_7^{(1)-} = \left|\left|D_7^{(1)}\right| * ec_{1,4}\right| = \lfloor 2 * 0{,}7 \rfloor = 2 \end{cases}$$

$$\left|D_8^{(1)}\right| \leq \left\lfloor \frac{\left|A^{(1)}\right|}{2} \right\rfloor \rightarrow \begin{cases} O_8^{(1)+} = \left|D_8^{(1)}\right| = 2 \\ O_8^{(1)-} = \left\lfloor \left|D_8^{(1)}\right| * ec_{1,4} \right\rfloor = \lfloor 2 * 0{,}7 \rfloor = 1 \end{cases}$$

Selected variant $D_5^{(1)}$ .

Step 4

$$t_{1,3}^{(2)} = t_{13}^{(1)} - O_5^{(1)-} = 6 - 2 = 4$$

$$ADt = O_2^{(1)+} = 3$$

$$\left(EF\left(t_{1,3}^{(k+1)}\right) = 15\right) < (EF(ADt) = 17) \rightarrow O_5^{(1)+} = 2; O_5^{(1)-} = 1$$

Variant $D_7^{(1)}$ must be selected

$$t_{1,3}^{(2)} = t_{13}^{(1)} - O_7^{(1)-} = 6 - 2 = 4$$

$$ADt = O_7^{(1)+} = 3$$

$$\left(EF\left(t_{1,3}^{(k+1)}\right) = 15\right) < (EF(ADt) = 17) \rightarrow O_7^{(1)+} = 2; O_7^{(1)-} = 1$$

This results in the fact that variant $D_2^{(1)}$ must be chosen, which uses a crew with a higher efficiency rate

$$t_{1,3}^{(2)} = t_{13}^{(1)} - O_2^{(1)-} = 6 - 1 = 5$$

$$ADt = O_2^{(1)+} = 2$$

Resulting schedules are shown on the illustrations below, which were produced by using the four rules described in this paper.

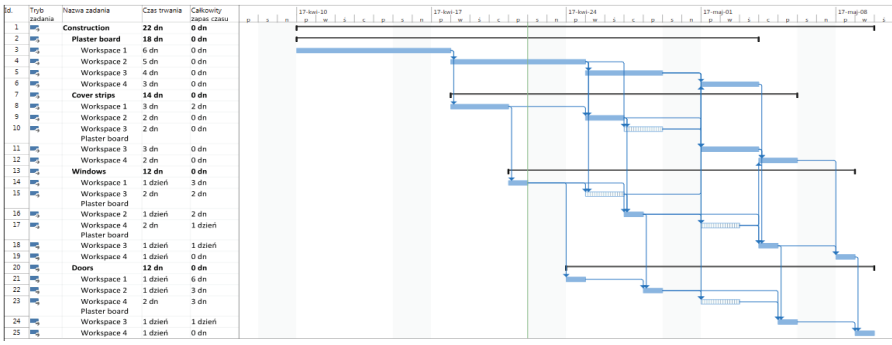


Fig. 3. Resulting schedule with the use of rule number 1

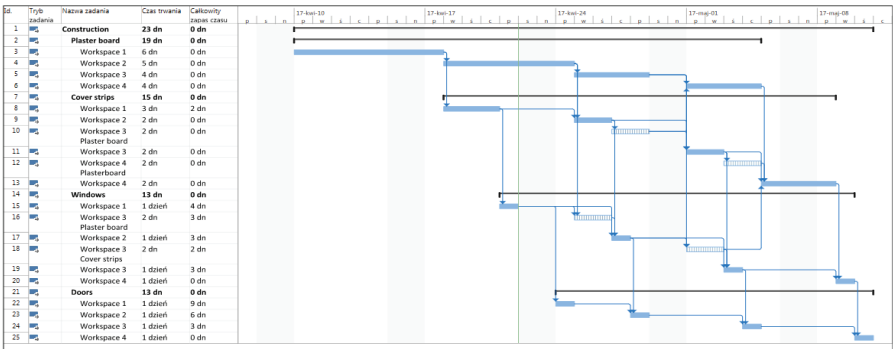


Fig. 4. Resulting schedule with the use of rule number 2

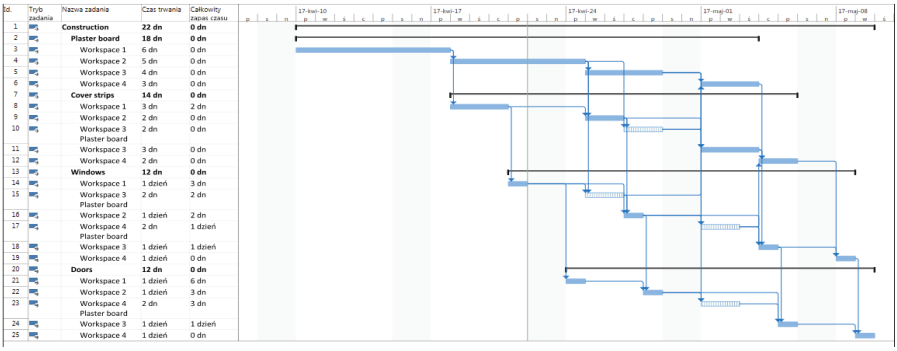


Fig. 5. Resulting schedule with the use of rule number 3

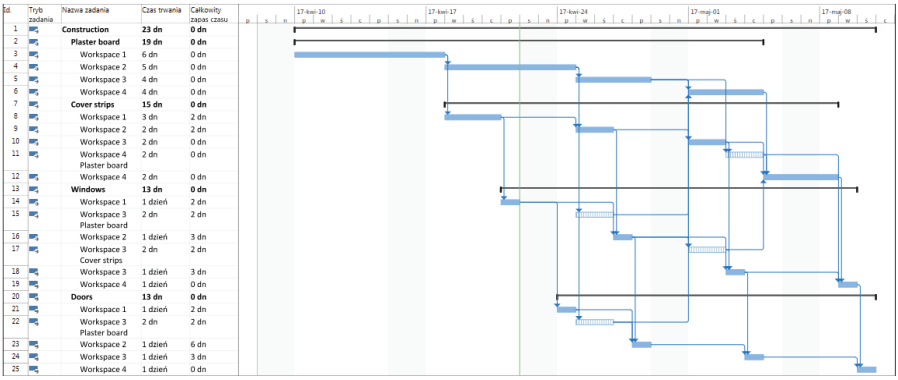


Fig. 6. Resulting schedule with the use of rule number 4

Table 2. Table form summary of obtained results

| Schedule | Duration | Total slack |
|---|---|---|
| Base | 26 | 32 |
| R1 | 22 | 12 |
| R2 | 23 | 16 |
| R3 | 22 | 12 |
| R4 | 23 | 16 |

The results obtained in the presented example demonstrates that instances of downtime can be reduced by anywhere from 50 to 70 percent, depending on the rule being used. This is clearly a very rewarding result. In addition, the overall project duration is shortened by 12 or 15 percent, depending on which rule is used.

## 3. SUMMARY

This paper has presented a newly devised algorithm for optimizing construction project schedules using the flow shop model. The sorting of tasks was not the focus of this research, as this issue has already been solved. The problem being solved by the new algorithm is how to reduce the number of times work is halted for crews on a given project. This paper presents an algorithm for this model. Four methods are offered to search the network model. The developed model allows the use of efficiency dependencies. The model assumes a higher rank to more efficient crews when assigning work. The results presented in table 2 demonstrates that instances of downtime can be reduced by anywhere from 50 to 70 percent, depending on the rule being used. This is clearly a very rewarding result. In addition, the overall project duration is shortened by 12 or 15 percent, depending on which rule is used. Due to the results obtained by this research, it can be ascertained that this model can be successfully used in construction.

## REFERENCES


1. S. Biruk, P. Jaskowski, "Harmonogramowanie przedsięwzięć wieloobiektowych z ciągłą realizacją procesów na działkach roboczych" Zeszyty Naukowe WSOWL Nr 3 (157): 340-349, 2010
2. Z. Hejducki, "Sprzężenia czasowe w metodach organizacji złożonych procesów budowlanych", Prace Naukowe Instytutu Budownictwa Politechniki Wrocławskiej, Monografie nr 34, 2000.
3. Z. Hejducki, "Scheduling model of construction activity with time coupulings", Journal of Civil Engineerin and Management, 9:4: 284-291, 2003.
4. M. Krzemiński, "Use of the KASS program in scheduling", Technical Transaction, ISSUE 2-B(6): 217-224, 2014.
5. M. Krzemiński, "KASS v.2.2. Scheduling Software for Construction with Optimization Criteria Description", Acta Physica Polonica A, Vol. 130 No. 6: 1439 – 1442, 2016.
6. R. Marcinkowski, "Metody rozdziału zasobów realizatora w działalności inżynieryjno – budowlanej", WAT, 2002
7. M. L. Pinedo, "Scheduling: Theory, Algorithms, and Systems" Springer, 2012
8. M. Rogalska, W. Bożejko, Z. Hejducki, "Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling", Automotion in Construction, 18: 24-31, 2008.


### LIST OF FIGURES AND TABLES:



*Received 28.09.2017*

*Revised 08.12.2017*

# OPTYMALIZACJA HARMONOGRAMÓW WYKONYWANYCH METODĄ POTOKOWĄ PRZY ZAŁOŻONEJ WIELOZADANIOWOŚCI BRYGAD ROBOCZYCH

*Słowa kluczowe*: harmogramowanie, modele potokowe, wielozadaniowość brygad roboczych, ciągłość pracy brygad

**STRESZCZENIE**

Harmonogramowanie przedsięwzięć wieloobiektowych jest tematem wielu prac naukowych z zakresu inżynierii przedsięwzięć budowlanych. Podstawowym założeniem jest organizacja robót przy zastosowaniu metody potokowej, a więc takiej w której mamy do czynienia z brygadami wykonującymi prace kolejno po sobie i przechodzącymi pomiędzy poszczególnymi działkami roboczymi. Pierwszym krokiem jaki należy wykonać jest opracowanie harmonogramu bazowego. Chcąc wykonać jego optymalizację w pierwszej kolejności wykonuje się szeregowanie zadań mające na celu ustalenie właściwej kolejności przechodzenia brygad pomiędzy działkami roboczymi, czyli ustalenie kolejności działek roboczych. Tą część optymalizacji można wykonać m. in. z zastosowaniem sprzężeń czasowych, można posłużyć się kolejnościowaniem z zastosowaniem kryterium czasowo-kosztowego, możliwe jest również na przykład zastosowanie zagadnienia komiwojażera. Swoje zastosowanie mogą znaleźć tutaj zarówno modele stosowane w produkcji przemysłowej jak również modele oparte na metodach z zakresu metod sztucznej inteligencji. Próbę rozwiązania zadania z zastosowaniem techniki przeglądu zupełnego podejmował również autor w swoich wcześniejszych pracach.

Zastosowanie wymienionych wcześniej metod nie gwarantuje jednak uzyskania ciągłości w pracy brygad przy równoczesnej ciągłości w pracach na frontach roboczych. Przy odpowiednio niekorzystnych danych wejściowych, wynikających z technologicznych zależności budowlanych, ciągle możliwe jest wystąpienie przestoi w pracach. Chcąc je wyeliminować najprostszym zabiegiem może być zmiana wielkości zatrudnienia środków produkcji mająca na celu przyspieszenie procesów generujących przestoje. Może się jednak okazać że taki zabieg nie jest możliwy z różnych względów takich jak ograniczona wielkość zaplecza nie pozwalająca na zwiększenie zatrudnienia czy też bariery technologiczne.

Wymienione wcześniej metody opierają się o założenie jednozadaniowości brygad roboczych czyli że dany zespół przypisany jest do jednego rodzaju zadania. W praktyce budowlanej można jednak zauważyć że często brygada która powoduje przestój w pracy brygady następującej jest przez nią wspomagana. Zdaniem autora warto zatem opracować model pozwalający na przyporządkowanie dodatkowych prac brygadom z przestojami już na etapie projektowania harmonogramu wykonywania robót. Model taki został zaprezentowany w dalszej części artykułu.

Założeniem metody jest że kiedy brygada wspomaga inną brygadę jej wydajność na wykonywanym procesie może być taka sama jak brygady podstawowej lub mniejsza. Wydajności charakteryzuje macierz wydajności $EC_{p,i}$

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

gdzie:

$ec_{p,i}$ – oznacza wydajność brygady $p$ przy wykonywaniu prac na procesie $i$

$p$ – numer wspomaganego procesu,

$i$ – numer brygady (wspomagającej),

$m$ – łączna liczba brygad roboczych.

Na poniższym rysunku przedstawiono algorytm metody likwidacji przestojów w pracach brygad roboczych.

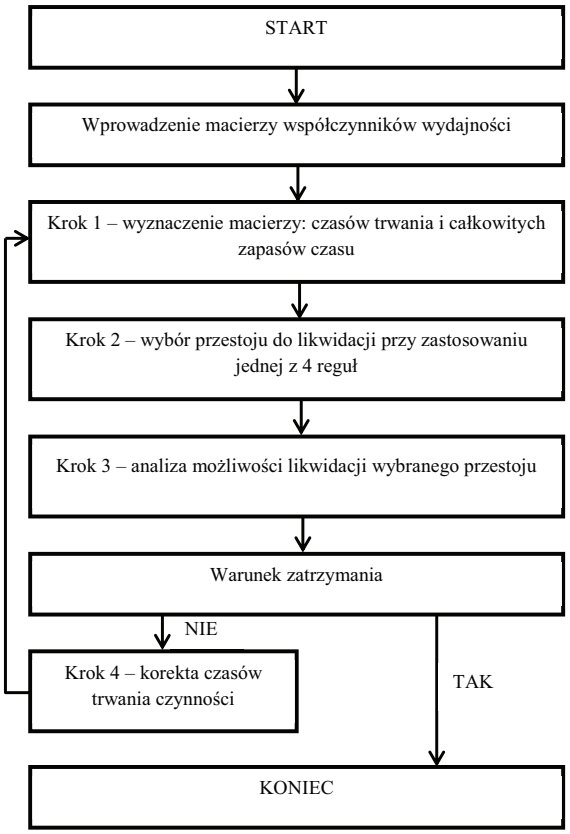


Fig. 1. Algorytm metody likwidacji przestojów

W przedstawionym artykule zaprezentowano nowo opracowany algorytm optymalizacji harmonogramów budowlanych wykonywanych przy założeniu organizacji metodą potokową. Zadaniem nowego algorytmu jest takie przypisanie pracy brygadom mającym przestoje które spowoduje zmniejszenie ich całkowitej liczby. Zaproponowano cztery różne reguły przeszukiwania modelu sieciowego. Opracowany model pozwala na wprowadzenie zależności wydajnościowych. Model w swoim założeniu przyporządkowuje w pierwszej kolejności prace brygadom o największej wydajności. Zaprezentowane wyniki pokazują że w zależności od zaproponowanej reguły zmniejszenia przestoi wahają się od 50 do 75 procent co można z pewnością uznać za wynik satysfakcjonujący. Dodatkowo skróceniu uległ całkowity czas realizacji przedsięwzięcia, tu wartości wynoszą odpowiednio 12 i 15 procent. Ze względu na otrzymane wyniki można stwierdzić że opracowany model może z powodzeniem być zastosowany w budownictwie.