



MANAGEMENT OF RESOURCES IN MULTIUNIT CONSTRUCTION PROJECTS WITH THE USE OF A TABU SEARCH ALGORITHM

Michał PODOLSKI

*Faculty of Civil Engineering, Wrocław University of Technology,
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland*

Submitted 25 Nov 2014; accepted 30 Mar 2015

Abstract. The paper presents the problem of optimal management of the resources in multiunit construction project. During the creation of the models of this kind of the project the flow-shop system is rarely used. Flow-shop system now is widely applied in modelling of industrial processes, computer systems. In the model presented in the paper flow shop system with teams of working groups is used. It allows its users to shorten the time of realization of multiunit construction projects in a significant way, because works in the project can be realized by teams of working groups of any cardinality. This is new model of multiunit construction project which has not been examined in field of scheduling of construction project yet. The presented optimization model of multiunit construction project with teams of working groups performing one type of work is NP-hard optimization problem. During the scheduling of such projects there was an artificial intelligence tool used, i.e. metaheuristic tabu search algorithm. Tabu search algorithm can provide solutions of very good quality. The paper also presents a calculation example of the above mentioned problem. The obtained in calculation example result is fully satisfactory.

Keywords: scheduling, tabu search, flow shop, construction project.

Introduction

In the field of construction practice it is not uncommon to encounter cases of realization of such projects, which involve the implementation of either a number of building structures or their separate parts (which constitute the project). The project may include the following examples: multi-storey buildings, residential single-family houses or group of buildings, motorways, roads, pipelines. Such projects are characterized by the repetition of the execution of works on each unit of project. This feature results in the need to specialize working groups in such a way that they carry out works of one type. Working groups in such construction projects move from the previous unit to the next realizing only designated scope of work. Construction projects of this kind are called *repetitive projects* or *multiunit projects* (Harris, Ioannou 1998).

Research on the scheduling of the above mentioned projects give us many models providing their accurate description. The models show detailed solutions appearing in it. Such projects as multi-storey buildings, single family houses or a group of other buildings are dominated by methods based on the so-called “*Line of Balance*” concept – LOB (Arditi *et al.* 2002). Many techniques based on assumptions of this method include: LSM (*Linear Scheduling Method*) by Chrzanowski and Johnston

(1986), RPM (*Repetitive Project Model*) by Reda (1990), VPM (*Vertical Production Method*) by O’Brien (1975). There are also techniques for monitoring and updating of scheduling for repetitive projects using the idea of “critical path” occurring in CPM/PERT methods (e.g. Hamerlink, Rowings 1998; Lucko 2009). As part of the models created for repetitive projects there are optimization problems with different objective functions and constraints often solved. Most problems being solved are optimization problems with such criteria as: minimizing project duration, minimizing total outage time of working groups, minimizing the cost of the project taking into account the imposed deadline conditions. In order to solve this type of optimization tasks, the following tools are applied: linear programming (e.g. Ipsilandis 2007), dynamic programming (e.g. El-Rayes, Mosehli 2001), neural networks (e.g. Adeli, Karim 1997), genetic (e.g. Huang, Sun 2009; Agrama 2014), simulated annealing (e.g. Chen, Shahandashti 2007), or hybrid algorithms (e.g. Ezeldin, Soliman 2009).

Other approaches include models that use system analogous to the flow shop problem encountered in classical scheduling theory. Such scheduling models now are widely used in industrial application, e.g. chemical branches, computer systems, telecommunication networks,

in precast production (e.g. Ko, Wang 2010), but rarely in construction projects (e.g. Bozejko *et al.* 2012). One of the most commonly used models in scheduling theory is permutation flow shop system (Gupta, Stafford 2006). This system can be applied in scheduling of repetitive construction projects when it is a possibility to take into account units processing order in creation an optimal project schedule. One of the constraints in this model is the use of a set of m single working groups (machines in scheduling theory) from the set $B = \{B_1, B_2, B_3, \dots, B_k, \dots, B_m\}$, in which each group B_k can perform work of one k kind. This constraint follows directly from the principles of the flow shop system analyzed in scheduling theory. In practice, we can often encounter the situations when we need to reduce project duration. Project duration can be reduced by assigning more labor to project, in the form of over time or by assigning more resources (working groups). Consequently, below there will be presented a new model of multiunit construction project with use of the flow shop system, which allows its users to use any number of working groups to carry one type of work in project. This process will enable a significant acceleration of the works in the project, while meeting the accepted constraints.

1. Optimization model of the considered multiunit project

The basis for the creation of the models used in many industrial applications is the *permutation flow shop problem* (FP), which is considered in scheduling theory. As it has been mentioned before in these models a given set of work is carried out only by one working group (machine in scheduling theory). This system is shown schematically in Figure 1.

As mentioned above, one of the constraints in permutation flow shop system is the use of a set of m single working groups (machines in scheduling theory) from the set $B = \{B_1, B_2, B_3, \dots, B_k, \dots, B_m\}$, in which each group B_k can perform work of one k kind. In new presented model it is possible to use any number of working groups to carry one type of work in the project. In models which use permutation flow shop problem the decisive variable is a sequence (permutation) of object realization: $\pi = (\pi(1), \pi(2), \dots, \pi(k), \dots, \pi(n))$, where n – number of units. This permutation is constant in process of carry-

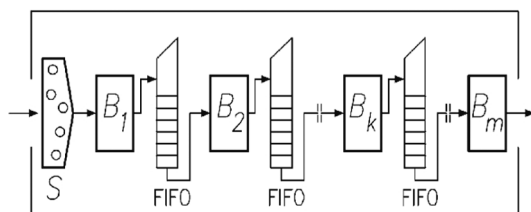


Fig. 1. Permutation flow shop system: S – input sequence (permutation) of n jobs (units of construction project) to the system of working groups B_1, \dots, B_m . The cardinality of the set of possible solutions is $n!$ (n – number of units)

ing out of all works in project. In turn, in the new model of multiunit construction project presented in the paper, there is the adopted assumption that the units processing order can be different for types of works in project. In addition, there is a particular type of assignment of a given work performed in the specified unit to be implemented by the selected working group from the entire team of working groups. The assumed allocation of resources to realization of works in units significantly increases the number of possible schedules and implies a possibility of acceleration of works in project. In the model presented in the paper decisive variable will be a set of permutations with detailed allocation of working groups to performing works in units of project (see decisive variable in the model).

The next constraint in the models which use permutation flow shop problem is sequence of carrying out of works in project. It is assumed in these models each works should be realized in sequence, without additional interruption, i.e. next work should be realized after ending of previous work. At any time only one type of work may be carried out. In the new model presented in paper the new type of sequence dependences of works in units are introduced. These sequence dependence for each unit are given on the example of directed acyclic graph:

$$K = (M, F), \quad (1)$$

where M is a set of nodes representing works in project; F is a set of arcs. Arcs are used to show the precedence relationships that exist between the works in units. It is assumed that relationships can be as follows (PMBOK®PMI 2013): SS (start – start), FF (finish – finish), FS (finish – start) with the accepted values of lag or lead, which can take any value. The form of the graph K is the same for each unit. Values of lags or leads can be different in units of project.

Additional constraint in the new model presented in the paper is the possibility to take into account additional times required for the movement of the working groups between units, depending on the type of the working group and the sequencing of units' realization. This situation can be happen when units of project are distant from each other. Above described constraints in model presented in paper are new in research of scheduling models of multiunit construction projects, which use flow shop processing.

The following optimization model of multiunit project is defined. Adopted assumption is that any numbers of specialized working groups perform the work of one type.

There are the following parameters defined:

- The project is constituted by a set of units:

$$Z = \{Z_1, Z_2, Z_3, \dots, Z_j, \dots, Z_n\}. \quad (2)$$

- For the execution of the works in the project, there are teams of working groups created, each of which performs one type of work. They form a set of:

$$B = \{B_1, B_2, B_3, \dots, B_k, \dots, B_m\}. \quad (3)$$

- In each team of working group $B_k \in B$ there is $m_k \geq 1$ of the same working groups (of the same performance and constitution):

$$B_k = \{B_{k1}, B_{k2}, B_{k3}, \dots, B_{ki}, \dots, B_{km_k}\}. \quad (4)$$

- Each unit $Z_j \in Z$ requires performance of m works which constitute the set:

$$O_j = \{O_{j1}, O_{j2}, O_{j3}, \dots, O_{jk}, \dots, O_{jm}\}. \quad (5)$$

- It is assumed that work $O_{jk} \in O_j$ can be realized by the working group $B_{ki} \subset B_k$. Work duration time O_{jk} performed by the group B_{ki} equals $p_{jk} > 0$. The set of duration of p_j works from the set O_j is defined by vector:

$$P_j = \{p_{j1}, p_{j2}, p_{j3}, \dots, p_{jk}, \dots, p_{jm}\}. \quad (6)$$

- The sequence dependence of O_j works for unit Z_j are given on the example of directed acyclic graph $K = (M, F)$ (2). M is a set of nodes representing works O_j of unit Z_j ($|M| = mn$). F is a set arcs. Arcs are used to show the precedence relationships that exist between the works in units. It is assumed that relationships can be as follows: SS (start – start), FF (finish – finish), FS (finish – start) with the accepted values of lag or lead, which can take any value.

- Additional time required for the movement of the working groups between units, depending on the type of the working group and the sequencing of units realization is defined by matrix $S^S_k = [s^S_{gh}]_{n \times m}$ where $g \in 1..n, h \in 1..m, k \in 1..m$.

The following constraints are defined:

- It is assumed that at any time, each working group from B_k team can perform only one work.
- It is assumed that work $O_{jk} \in O_j$ is performed without any stops by one working group from B_k team in $p_{jk} > 0$ time.

Decisive variable is units processing order, which is defined by m -tuple:

$$\pi = (\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_m). \quad (7)$$

Performing of the work k by the team of working group B_k in all units in project is defined by set:

$$\pi_k = (\pi_{k1}, \pi_{k2}, \dots, \pi_{ki}, \dots, \pi_{km_k}), \quad (8)$$

where: $\pi_{ki} = (\pi_{ki}(1), \pi_{ki}(2), \dots, \pi_{ki}(l), \dots, \pi_{ki}(n_{ki}))$. π_{ki} determines the processing order of units by working group B_{ki} from the team of working groups $B_k \subset B$ of size m_k .

In the new presented model the task is to find a schedule (S, A) , (where $S = [S_{jk}]_{n \times m}$, $A = [a_{jk}]_{n \times m}$ and S_{jk} defines earliest start time for carrying out works in unit Z_j by team of working groups B_k and a_{jk} defines number of working group allocated to carrying out works in unit Z_j) in order to minimize or maximize value of objective function, while meeting the accepted constraints.

Criterion (objective function) is a term C_{max} of implementation of all works in all units. Optimization task in the model is to find a schedule for the work realization which minimizes the value of the objective function C_{max} , satisfying the constraints given above.

The considered model can be represented in the form of a disjunctive graph $G(\pi)$. The form of the graph is dependent upon an established decisive variable π (an example of such a graph is given in Figure 2): $G(\pi) = (N', E(\pi))$, where N' is a set of nodes, $E(\pi)$ – set of arcs. It is assumed that $N' = N \cup \{(start), (finish)\}$, where $N = \{1, \dots, k, \dots, m\} \times \{1, \dots, j, \dots, n\}$ is a set of nodes representing works in units of project. Nodes $(start), (finish)$ is fictitious nodes representing start and finish operation. Weight of node (k, j) is equal to the time of work execution $p_k, \pi_{ki}(l)$, weights of nodes $(start), (finish)$ equal 0. The set of $E(\pi) = E^F \cup E^S(\pi)$ depends on the assumed decisive variable π . Vertical arcs (sequential, representing processing order of units) from the set $E^S(\pi)$ are between nodes $\pi_{ki}(l-1)$ and $\pi_{ki}(l)$, where $i = 1, \dots, m_k, k = 1, \dots, m, l = 2, \dots, n_{ki}$. Horizontal arcs (technological) from the set E^F are between node standing for work k and nodes which are predecessors of k work for unit $\pi_{ki}(j)$. The weight of the vertical arc from set $E^S(\pi)$ is $s^S_{k, \pi_{ki}(l-1)\pi_{ki}(l)}$. The weight of the horizontal arc from set E^F is the lag or lead time depending on assumed relationships: SS, FF, FS.

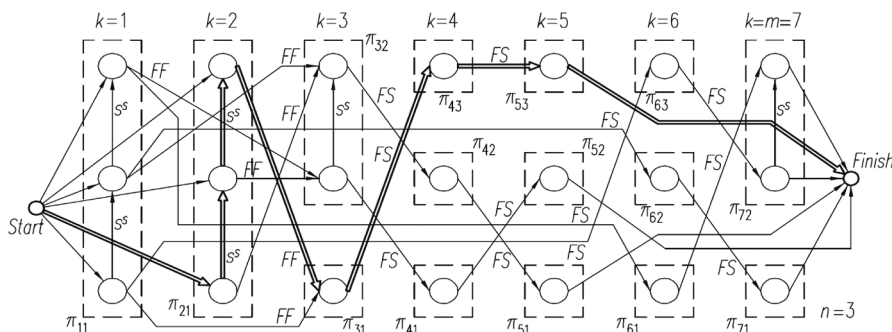


Fig. 2. Example of a graph $G(\pi)$ for model of multiunit project presented in the paper ($m = 7, n = 3$) with a marked critical path

The earliest finish time for works can be determined from the recursive formula:

$$C_{k,\pi_{ki}(l)} = \max \left\{ C_{k,\pi_{ki}(l-1)} + S_{k,\pi_{ki}(l-1)\pi_{ki}(l)}^S, \max_f \{ C_{f,\pi_{ki}(l)} + r_{f,\pi_{ki}(l)} \} \right\} + P_{k,\pi_{ki}(l)}, \quad (9)$$

where: $j = 1, \dots, n, l = 1, \dots, n_{ki}, i = 1, \dots, m_k, k = 1, \dots, m, \pi_{ki}(0) = 0, C_{k,0} = 0, C_{0,j} = 0, f = 1, \dots, m_{poprz}$. Variable m_{poprz} defines the number of predecessors of k work, whereas the variable f denotes the number of works on the list of predecessors of k works. Variable $r_{f,\pi(j)}$ determines the value of the lag or lead time for completion of f work on the list predecessors of k work resulting from the adoption of the relationships type (SS, FF, FS). Any graph $G(\pi)$ for the present model has the property of the critical path of length C_{\max} .

The presented new model of multiunit construction project has not been examined in field of scheduling of construction project yet. In the scheduling theory presented model is a kind of flow shop problem with parallel machines (i.e. hybrid flow shop problem). If we assume that works in units are realized in sequence like in classical permutation flow shop problem (not in directed acyclic graph form), relationships are always finish-start with lag 0 and $S_{k,\pi_{ki}(l-1)\pi_{ki}(l)}^S = 0$ we get classical flow shop problem with parallel machines with the C_{\max} criterion. In literature this problem is strongly NP-hard. From this fact it follows that the presented optimization model of multiunit project with multiple working groups performing one type of work is also NP-hard and because of the constraint (any value of lags in the adopted relationships – SS, FF, FS) it is not possible to create a branch-and-bound algorithm for solving optimization problems. Such an algorithm would enable its users to solve a given discrete optimization task in an accurate way, however, it would require a computational time in the form of very rapidly increasing exponential function along with an increase in the size of the problem. The solution to this problem adopted in the analyzed model of multiunit project was with the use of an approximate, metaheuristic tabu search algorithm.

2. Solution to computational problem with the use of tabu search algorithm

Nowadays a tabu search algorithm (abbreviated as TS) is successfully used to solve various optimization problems due to the very good quality of delivered solutions (Glover, Laguna 1997; Nowicki, Smutnicki 1998; Bozejko *et al.* 2013). It replicates the natural searching process to find solutions to problems posed by men. Basic version of the TS algorithm starts its working with a particular startup solution. Then, for this solution, there is neighborhood found. It is defined as a set of solutions that can be created after performing of moves in a

given solution, i.e. transformation of a given solution into another one according to established rules. The solution with the smallest value of the objective function is sought in the neighborhood. This solution is a base solution for the next iteration. The result of the algorithm is the best solution of the whole search trajectory. Below there is a general algorithm of TS method used to solve flow shop problems in scheduling theory.

Let $\pi \in \Pi$ be any permutation, LT – tabu list, c – adopted objective function, whereas π^* – the best so far found solution (in the beginning we adopt permutation π for π^*).

Step 1. Determine neighborhood N_π of permutation π which does not contain a list of items prohibited by LT ;

Step 2. Find permutation $\delta \in N_\pi$ such that:

$$c(\delta) = \min \{ c(\beta) : \beta \in N_\pi \};$$

Step 3. If $c(\delta) < c(\pi^*)$, to $\pi^* \leftarrow \delta$;

Put attributes d on LT list;

$$\pi \leftarrow \delta;$$

Step 4. If Finish_Condition, then STOP otherwise go to Step 1.

Tabu search method has many degrees of freedom: the choice of a move and determining the neighborhood, forms implementing the mechanism of tabu (e.g. the length of the tabu list, form of attributes), search strategy. Currently TS algorithm is one of the most effective tools used in the scheduling theory. Its form used to solve the optimization problem in the presented model is based on the work (Nowicki, Smutnicki 1998).

Algorithm tabu search used in presented model starts with the initial solution. It is assumed that initial solution (a m -tuple π , – see decide variable) is obtained randomly. Then we should to assume the following basic elements of algorithm: movement, neighbourhood, tabu list, completion condition. Movement is a function transforming one solution into another. Neighbourhood is a set of solution which is possible to obtain from a determined solution by means of a class of movements. In presented algorithm TS it is assumed that movement is “insertion” type. The “insertion” move operates on set π_k (Eqn (10)) and removes a job placed at a position in this set and inserts it in another position of this set. More precisely “insertion” move is assumed as $v = (k, a, y, b, z)$. It means that at center k (i.e. team of working groups) the job (i.e. unit number) at position y on machine (i.e. working group) a will be move at position z on machine b . Instance: let $\pi_k = ((2, 3, 1), (5, 4))$, “insertion” move $v = (k, 1, 2, 2, 1)$. After “insertion” move v we get $\pi_k^v = ((2, 1), (3, 5, 4))$. In presented algorithm TS the neighbourhood N_π consists of unit processing orders π (Eqn (9)) with the set π_k^v generated by all possible “insertion” movements in a given randomly chosen set π_k .

The next basic element in presented algorithm TS is the tabu list LT . The tabu list LT is a list including attributes of movements of recently examined solutions. For

movement $v = (k, a, y, b, z)$ it is assumed the following attributes:

$$AV(\pi, v) = \begin{cases} \{(\pi_{ka}(y), \pi_{ka}(y+1))\}, & b = a, y < z \\ \{(\pi_{ka}(y-1), \pi_{ka}(y))\}, & b = a, y > z \\ \{(\pi_{ka}(y-1), \pi_{ka}(y))\}, \{(\pi_{ka}(y), \pi_{ka}(y+1))\} & b \neq a. \end{cases} \quad (10)$$

The important elements of tabu list LT are conditions when movement v for unit processing order π (Eqn (7)) is forbidden according to tabu list LT (to prevent generating a cycle). For movement $v = (k, a, y, b, z)$, where $a = b$ it is assumed following conditions: movement $v = (k, a, y, b, z)$ which is realized to the right (i.e. $y < z$) is forbidden when the tabu list LT includes at least such the pair of jobs (i.e. units) (j_1, j_2) , where $j_1 = \pi(i)$ for $y < i \leq z$ and $j_2 = \pi(y)$. Respectively, movement $v = (k, a, y, b, z)$ which is realized to the left (i.e. $y > z$) is forbidden when the tabu list LT includes at least such the pair of jobs (j_1, j_2) , where $j_1 = \pi(i)$ for $z \leq i < y$ and $j_2 = \pi(y)$. For movement $v = (k, a, y, b, z)$, where $a \neq b$ it is assumed following conditions: movement $v = (k, a, y, b, z)$ is forbidden when the tabu list LT includes at least such the pair of jobs (i.e. units) (j_1, j_2) , where $j_1 = \pi_{kb}(i)$ for $1 \leq i < z, j_2 = \pi_{ka}(y)$ or $j_1 = \pi_{ka}(y), j_2 = \pi_{kb}(i)$ for $z \leq i \leq n_{kb}$ (n_{kb} – length of π_{kb}). The only parameter of tabu list is its length (the number of its elements). In presented algorithm TS length of tabu list equals 1/3 nm. The completion condition is as following: the algorithm TS stops after 20000 iterations. Programming implementation of TS algorithm for

the considered model was made by author of the paper in *Mathematica* environment.

3. Case study

On behalf of the investor the contractor is to realize a project in the form of the construction of $n = 12$ petrol stations that are distant from each other. This implies the need to take into account the additional time needed to move working groups between different stations. In the project, each station requires realization of $m = 15$ works, among which the sequence dependence is presented in Table 2. The contractor assumes that it has the following amounts of working groups, for each of the works, which have the same performance and size: $k = 1$ (A – land development) – 3 groups; $k = 2$ (B – earth works) – 3 groups; $k = 3$ (C – foundations) – 4 groups; $k = 4$ (D – tanks and freylits) – 2 groups; $k = 5$ (E – water supply system) – 4 groups; $k = 6$ (F – electrical grid) – 3 groups; $k = 7$ (G – masonry works) – 4 groups; $k = 8$ (H – steelworks) – 3 groups; $k = 9$ (J – roofing) – 4 groups; $k = 10$ (K – roads and square) – 5 groups; $k = 11$ (L – installations) – 4 groups; $k = 12$ (M – plastering) – 3 groups; $k = 13$ (N – floors, tiles) – 3 groups; $k = 14$ (O – painting) – 3 groups; $k = 15$ (P – equipment installation) – 3 groups. On the basis of not only productivity in the individual stations but also the composition and performance of working groups there were durations of works execution established, which are shown in Table 1. The relationships between works are the same in all stations and are shown in Table 2. The types of relationships between works and lag times for each station are shown

Table 1. Duration of works executed by working groups, expressed in working days

Number and type of work $k =$	Station $j =$											
	1	2	3	4	5	6	7	8	9	10	11	12
1 (A – land development)	10	13	15	7	19	17	7	8	17	13	4	9
2 (B – earth works)	12	18	8	22	10	13	22	9	7	16	13	7
3 (C – foundations)	31	33	43	45	44	17	26	39	22	40	16	17
4 (D – tanks and freylits)	5	3	10	7	9	7	6	4	9	10	2	9
5 (E – water supply system)	50	42	39	17	17	39	45	41	18	49	28	39
6 (F – electric al grid)	25	31	17	10	30	26	29	11	16	9	18	18
7 (G – masonry works)	29	33	36	14	55	57	52	37	56	44	51	13
8 (H – steelworks)	15	20	22	29	25	28	10	18	25	10	7	20
9 (J – roofing)	18	16	10	14	25	9	10	30	31	16	28	29
10 (K – roads and square)	33	27	31	40	25	20	15	32	12	28	20	20
11 (L – installations)	24	43	35	19	37	37	39	42	19	43	12	10
12 (M – plastering)	29	16	26	20	29	19	31	24	34	20	26	26
13 (N – floor)	25	39	49	41	41	25	12	25	40	27	33	17
14 (O – paintings)	14	11	25	18	22	15	5	5	14	23	13	7
15 (P – equipment installation)	24	44	32	46	37	31	38	23	10	43	38	20

Table 2. Predecessors and successors of works in example

	Works $k =$														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Predecessors	start	start	1, 2	3	1, 2	5	3	7	8	8,6,4	9	9,7	11	12	10,13,14
Successors	5, 3	5, 3	4, 7	10	6	10	8,12	9, 10	11, 12	15	13	14	14, 15, finish	15	finish

Table 3. Types of relationships and lag times in example expressed in working days

Predecessor $k =$	Successors $k =$	Type of relationship	Lag or lead time for each station $j =$											
			1	2	3	4	5	6	7	8	9	10	11	12
1	3	SS	5	7	8	4	10	9	4	4	9	7	2	5
1	5	SS	5	7	8	4	10	9	4	4	9	7	2	5
2	3	SS	5	7	3	9	4	5	9	4	3	6	5	3
2	5	SS	5	7	3	9	4	5	9	4	3	6	5	3
3	4	FF	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
3	7	FS	-3	-3	-4	-5	-4	-2	-3	-4	-2	-4	-2	-2
4	10	FS	0	0	0	0	0	0	0	0	0	0	0	0
5	6	FF	0	0	0	0	0	0	0	0	0	0	0	0
6	10	FS	0	0	0	0	0	0	0	0	0	0	0	0
7	8	SS	19	21	23	9	36	37	34	24	36	29	33	8
7	12	SS	15	17	18	7	28	29	26	19	28	22	26	7
8	9	SS	5	7	8	10	9	10	4	6	9	4	2	7
8	10	FS	0	0	0	0	0	0	0	0	0	0	0	0
9	11	SS	7	6	4	6	10	4	4	12	12	6	11	12
9	12	FF	-2	-2	-1	-1	-3	-1	-1	-3	-3	-2	-3	-3
10	15	FS	0	0	0	0	0	0	0	0	0	0	0	0
11	13	SS	12	22	18	10	19	19	20	21	10	22	6	5
12	14	FS	0	0	0	0	0	0	0	0	0	0	0	0
13	14	FS	0	0	0	0	0	0	0	0	0	0	0	0
13	15	FS	0	0	0	0	0	0	0	0	0	0	0	0
14	15	SS	6	4	10	7	9	6	2	2	6	9	5	3

in Table 3. The relationships for station $j = 1$ are shown in Figure 3. Data concerning the time required due to the movement of the working groups between stations (depending on the type of the working group and the order of stations realization) are stored in the form of 15 matrices $S_k^{S_k}$ ($k = 1 \dots 15$): $S_2^{S_2} = S_3^{S_3} = S_7^{S_7} = S_8^{S_8} = S_{12}^{S_{12}} = S_{14}^{S_{14}} = S_{15}^{S_{15}} = S_{gh}^{S_{gh}} = 1$, $S_1^{S_1} = S_4^{S_4} = S_5^{S_5} = S_6^{S_6} = S_9^{S_9} = S_{11}^{S_{11}} = S_{13}^{S_{13}} = S_{14}^{S_{14}} = S_{15}^{S_{15}} = S_{gh}^{S_{gh}} = 0$, $S_{10}^{S_{10}} = S_{gh}^{S_{gh}} = 2$.

In connection with such a fixed structure and characteristics of the used resources (working groups), the

contractor has the ability to create an optimal schedule in reference to the criterion of the duration of the entire project which is C_{max} .

There was an reference solution adopted in accordance with the order of numeration of stations (natural), i.e. for the following decisive variable $\pi = (\pi_1, \pi_2, \dots, \pi_{15})$, where:

$$\pi_1 = \pi_2 = \pi_6 = \pi_8 = \pi_{12} = \pi_{13} = \pi_{14} = \pi_{15} = ((1,2,3,4),(5,6,7,8),(9,10,11,12)),$$

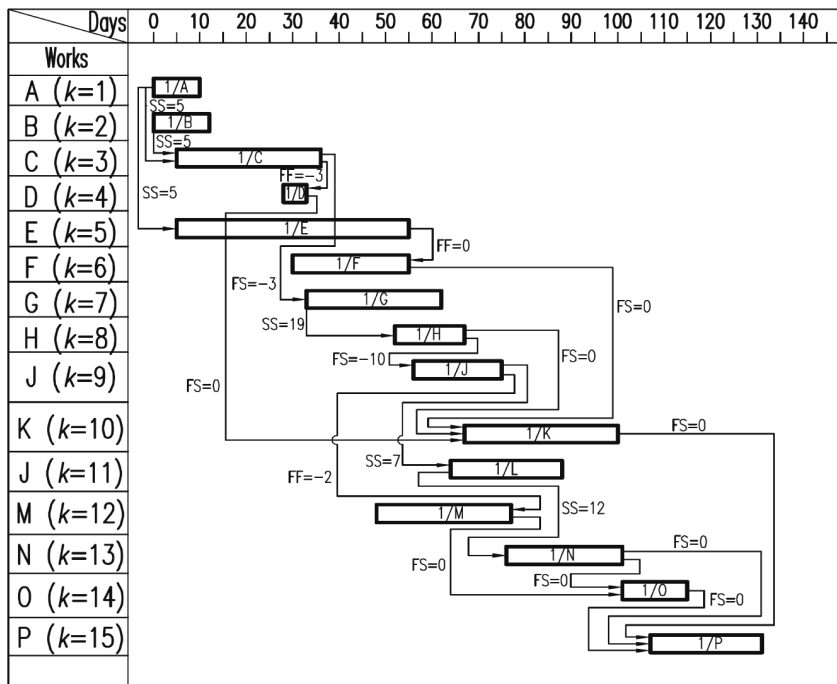


Fig. 3. Relationships between works in station $j = 1$ of project

$$\pi_3 = \pi_5 = \pi_7 = \pi_9 = \pi_{11} = ((1,2,3), (4,5,6), (7,8,9), (10,11,12)),$$

$$\pi_4 = ((1,2,3,4,5,6), (7,8,9,10,11,12)),$$

$$\pi_{10} = ((1,2,3), (4,5,6), (7,8), (9,10), (11,12)).$$

The numbers in parentheses means the numbers of stations in the project. In the double parentheses are given stations processing order for $m = 15$ works and teams of working groups. In the single parentheses are given stations processing order for each working group from team

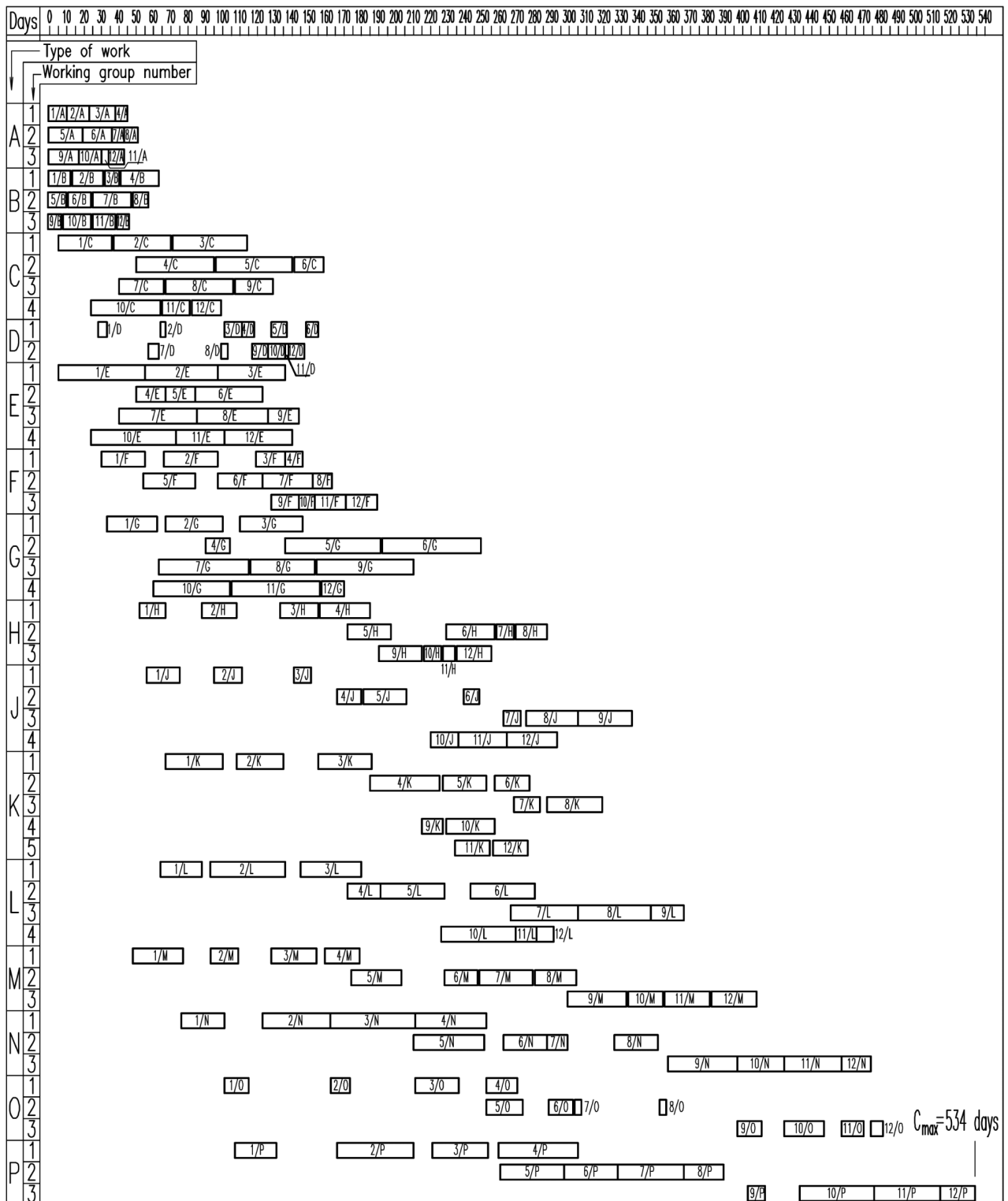


Fig. 4. Schedule of multiunit project for reference solution

of working groups. For instance: for work $m = 1$ (set π_1 – three groups) first working group carries out works in stations 1, 2, 3, 4, second working group – in stations 5, 6, 7, 8, third working group – in stations 9, 10, 11, 12. The duration of the realization of the presented project for the adopted decisive variable π equals 534 working days.

The schedule for the reference solution is shown in Figure 4.

Calculations with TS algorithm were performed three times. The obtained approximate minimum duration time of the entire project as shown in the example (the best value C_{max}) was 264 working days, which was achieved in the 16250 iterations of the algorithm operation (Fig. 5). The value of the objective function for the reference decisive variable was 534 working days. This

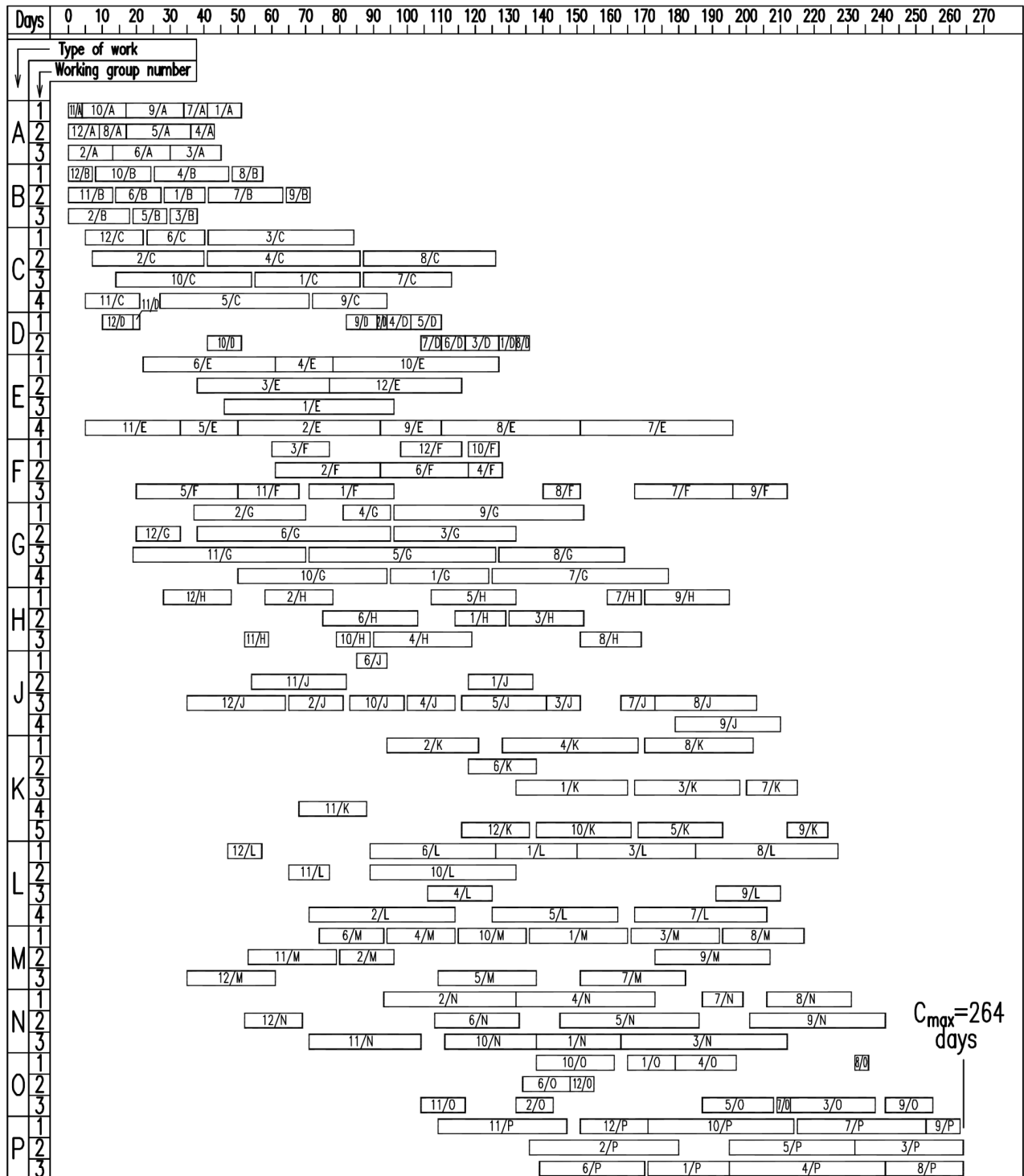


Fig. 5. Schedule of multiunit project for a found suboptimal solution

result was improved by 50.6% with the use of TS algorithm.

Best duration of the project was obtained for the decisive variable $\pi = (\pi_1, \pi_2, \dots, \pi_{15})$, where:

$$\pi_1 = ((11,10,9,7,1),(12,8,5,4),(2,6,3)),$$

$$\pi_2 = ((12,10,4,8),(11,6,1,7,9),(2,5,3)),$$

$$\pi_3 = ((12,6,3),(2,4,8),(10,1,7),(11,5,9)),$$

$$\pi_4 = ((12,11,9,2,4,5),(10,7,6,3,1,8)),$$

$$\pi_5 = ((6,4,10),(3,12),(1),(11,5,2,9,8,7)),$$

$$\pi_6 = ((3,12,10),(2,6,4),(5,11,1,8,7,9)),$$

$$\pi_7 = ((2,4,9),(12,6,3),(11,5,8),(10,1,7)),$$

$$\pi_8 = ((12,2,5,7,9),(6,1,3),(11,10,4,8)),$$

$$\pi_9 = ((6),(11,1),(12,2,10,4,5,3,7,8),(9)),$$

$$\pi_{10} = ((2,4,8),(6),(1,3,7),(11),(12,10,5,9)),$$

$$\pi_{11} = ((12,6,1,3,8),(11,10),(4,9),(2,5,7)),$$

$$\pi_{12} = ((6,4,10,1,3,8),(11,2,9),(12,5,7)),$$

$$\pi_{13} = ((2,4,7,8),(12,6,5,9),(11,10,1,3)),$$

$$\pi_{14} = ((10,1,4,8),(6,12),(11,2,5,7,3,9)),$$

$$\pi_{15} = ((11,12,10,7,9),(2,5,3),(6,1,4,8)).$$

In the double parentheses are given stations processing order for $m = 15$ works and teams of working groups. In the single parentheses are given stations processing order for each working group from team of working groups. For instance: for work $m = 2$ (set π_2 – three groups) first working group carries out works in stations 12, 10, 4, 8, second working group – in stations 11, 6, 1, 7, 9, third working group – in stations 2, 5, 3.

Conclusions

Multiunit projects are special cases of construction projects, in which optimization problems may arise. These problems for general models are usually NP-hard. It means that it is impossible to solve practical problems by algorithms providing accurate solutions in reasonable time. Scheduling theory is a field in which certain tools are used, i.e. metaheuristic algorithms for solving this kind of problems that may also occur in the scheduling of construction projects. Scheduling theory is also a field where we can find certain models for scheduling of construction projects. In this case the flow shop problem with parallel machines was proper to describe new model of multiunit construction project. It is important that in presented new model it is assumed that works in project were carried out with various sequences of units for given kind of work. Such realization of works in case study enabled a reduction in the execution time of the project by more than 50% using algorithm TS. This is

important confirmation not only of the effectiveness of the algorithm but also of the need to search for optimal solutions in the planning of construction projects (Podolski 2008). On one hand such realization of works (various units processing orders for all works) allows a big reduction in execution time of the project. But on the other hand it introduces complication in works organization of multiunit project due to complicated decisive variable.

Presented in the paper new model allows its practitioners to determine the suboptimal schedules of multiunit construction projects. The model can be used in situation when investor sets the deadline of the project. Contractor using presented model has a possibility of creation of optimal schedule which meets the investor's deadline of the project. Suboptimal schedule takes into account limited resources of contractor, i.e. teams of working group of known size.

Presented model can be used in deterministic situations. The technological and organizational conditions of the project, bill of quantities are known in project. Resources (working groups) are available all the time in the project. Works in the project are carried out by working groups with adopted productivity and quality. The next assumption of using presented model is that every unit in project requires performance of the same set of works. The model can be used when investor needs to carry out all works in all units in the project. The model can be applied for large projects with group of buildings that are distant from each other such as e.g. residential single-family houses, petrol stations, commercial buildings.

Acknowledgements

The author would like to thank the referees for their constructive criticism on an earlier version of this paper.

References

- Adeli, H.; Karim, A. 1997. Scheduling/cost optimization and neural dynamic model for construction projects, *Journal of Construction Engineering and Management* 123(4): 450–458.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1997\)123:4\(450\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1997)123:4(450))
- Agrama, F. 2014. Multi-objective genetic optimization for scheduling a multi-storey building, *Automation in Construction* 44: 119–128.
<http://dx.doi.org/10.1016/j.autcon.2014.04.005>
- Arditi, D.; Tokdemir, O. B.; Suh, K. 2002. Challenges in line-of-balance scheduling, *Journal of Construction Engineering and Management* 128(6): 545–556.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2002\)128:6\(545\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2002)128:6(545))
- Bozejko, W.; Hejducki, Z.; Wodecki, M. 2012. Applying metaheuristic strategies in construction projects management, *Journal of Civil Engineering and Management* 18(5): 621–630.
<http://dx.doi.org/10.3846/13923730.2012.719837>
- Bozejko, W.; Pempera, J.; Smutnicki, C. 2013. Parallel tabu search algorithm for the hybrid flow shop problem, *Computers & Industrial Engineering* 65: 466–474.
<http://dx.doi.org/10.1016/j.cie.2013.04.007>
- Chen, P.-H.; Shahandashti, S. M. 2007. Simulated annealing algorithm for optimizing multi-project linear scheduling

- with multiple resource constraints, in *Proc. of the 24th International Symposium on Automation and Robotics in Construction (ISARC 2007)*, 19–21 September 2007, Madras, India, 429–434.
- Chrzanowski, E. N.; Johnston, D. 1986. Application of linear construction, *Journal of Construction Engineering and Management* 112(4): 476–491.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1986\)112:4\(476\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1986)112:4(476))
- El-Rayes, K.; Moselhi, O. 2001. Optimizing resource utilization for repetitive construction projects, *Journal of Construction Engineering and Management* 127(1): 18–27.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2001\)127:1\(18\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2001)127:1(18))
- Ezeldin, A. S.; Soliman, A. 2009. Hybrid time–cost optimization of nonserial repetitive construction projects, *Journal of Construction Engineering and Management* 135(1): 42–55.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2009\)135:1\(42\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2009)135:1(42))
- Glover, F.; Laguna, M. 1997. *Tabu search*. Massachusetts, USA: Kluwer Academic Publishers. 382 p.
<http://dx.doi.org/10.1007/978-1-4615-6089-0>
- Gupta, J.; Stafford, E. F. Jr. 2006. Flowshop scheduling research after five decades, *European Journal of Operational Research* 169(3): 699–711.
<http://dx.doi.org/10.1016/j.ejor.2005.02.001>
- Hamerlink, D. J.; Rowings, J. E. 1998. Linear scheduling model: development of controlling activity path, *Journal of Construction Engineering and Management* 124(4): 263–268.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1998\)124:4\(263\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1998)124:4(263))
- Harris, R. B.; Ioannou, P. G. 1998. Scheduling projects with repeating activities, *Journal of Construction Engineering and Management* 124(4): 269–278.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1998\)124:4\(269\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1998)124:4(269))
- Huang, R.-y.; Sun, K.-S. 2009. A GA optimization model for workgroup-based repetitive scheduling (WoRSM), *Advances in Engineering Software* 40(3): 212–228.
<http://dx.doi.org/10.1016/j.advengsoft.2008.01.010>
- Ipsilandis, P. G. 2007. Multiobjective linear programming model for scheduling linear repetitive projects, *Journal of Construction Engineering and Management* 133(6): 417–424.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2007\)133:6\(417\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2007)133:6(417))
- Ko, C.-H.; Wang, S.-F. 2010. GA-based decision support systems for precast production planning, *Automation in Construction* 19(7): 907–916.
<http://dx.doi.org/10.1016/j.autcon.2010.06.004>
- Lucko, G. 2009. Productivity scheduling method: linear schedule analysis with singularity functions, *Journal of Construction Engineering and Management* 135(4): 246–253.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(2009\)135:4\(246\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(2009)135:4(246))
- Nowicki, E.; Smutnicki, C. 1998. The flow shop with parallel machines: A tabu search approach, *European Journal of Operational Research* 106: 226–253.
[http://dx.doi.org/10.1016/S0377-2217\(97\)00260-9](http://dx.doi.org/10.1016/S0377-2217(97)00260-9)
- O'Brien, J. J. 1975. VPM scheduling for high rise buildings, *Journal of the Construction Division* 101(4): 895–905.
- PMBOK®PMI 2013. *A guide to the project management body of knowledge (Pmbok guide)*. 5th ed. Project Management Institute. 589 p.
- Podolski, M. 2008. *Analiza nowych zastosowań teorii szeregowania zadań w organizacji robót budowlanych* [Analysis of new applications of job scheduling theory in construction work organization]: PhD thesis. Wrocław University of Technology, Wrocław, Poland (in Polish).
- Reda, R. M. 1990. RPM: Repetitive project modelling, *Journal of Construction Engineering and Management* 116: 316–330.
[http://dx.doi.org/10.1061/\(ASCE\)0733-9364\(1990\)116:2\(316\)](http://dx.doi.org/10.1061/(ASCE)0733-9364(1990)116:2(316))

Michał PODOLSKI. Phd, Eng, Assistant Professor at the Faculty of Civil Engineering of Wrocław University of Technology, Poland. Obtained his diploma in Civil Engineering at WUT in 1998. His research interests include: optimization of schedules in construction projects, applications of scheduling theory in construction projects, artificial intelligence methods (simulating annealing, genetic algorithm, tabu search).