# COST OPTIMIZATION OF MULTIUNIT CONSTRUCTION PROJECTS USING LINEAR PROGRAMMING AND METAHEURISTIC-BASED SIMULATED ANNEALING ALGORITHM

Michał PODOLSKI [ID][1*], Bartłomiej SROKA [ID][2]

[1]*Faculty of Civil Engineering, Wrocław University of Technology, Wrocław, Poland*
[2]*Faculty of Civil Engineering, Cracow University of Technology, Cracow, Poland*

**Abstract.** The article presents the cost optimization model for multiunit construction projects. Multiunit projects constitute a special case of repetitive projects. They consist in the realization of many different, when it comes to size, types of residential, commercial, industrial buildings or engineering structures. Due to the specific character of construction works, actual schedules of such projects should not only take into account real costs of construction, but also be subject to specific restrictions, e.g. deadlines for the completion of units imposed by the investor. To solve the NP-hard problem of choosing the order of units' construction there was metaheuristic algorithm of simulated annealing used. The objective function in the presented optimization model was the total value of the project cost determined on the basis of the mathematical programming model, taking into account direct and indirect costs, costs of missing deadlines and costs of work group discontinuities. In the article, an experimental analysis of the proposed method of solving the optimization task was carried out in a model that showed high efficiency in obtaining suboptimal solutions. In addition, the operation of the proposed model has been presented on a calculation example. The results obtained in it are fully satisfying.

**Keywords:** repetitive construction projects, scheduling, optimization, linear programming, simulated annealing, flow shop, time-cost trade-off.

## Introduction

Problems of scheduling construction projects have currently been the subject of many studies. Due to the method of conducting works, we usually divide them into undertakings with non-repetitive character and repetitive projects. Repetitive projects are primarily characterized by the possibility of their division into parts (units), which may be, for example, work zones, sections of a certain size (e.g. length), individual floors of buildings or entire buildings or structures. Each part of a given undertaking requires the implementation of the same or a similar set of activities using the available resources, which are construction crews. Examples of repetitive projects include single-family housing estates, groups of buildings, high-rise multi-story buildings, pipeline networks, roads and highways.

In order to schedule repetitive projects there is a concept of the Line of Balance used (Arditi, Tokdemir, & Suh, 2002). On its basis, the following techniques have been developed: Linear Scheduling Method (Chrzanowski &

Johnston, 1986), Repetitive Project Model (Reda, 1990), Linear Scheduling Model (Harmelink & Rowings, 1998), Repetitive Scheduling Method (Harris & Ioannou, 1998). The features of the above techniques include, among others, the ability to distinguish critical activities or their parts in the project schedule, the possibility of taking into account non-linear (centered) activities in the schedule (Harmelink & Rowings, 1998; Harris & Ioannou, 1998), or ensuring the continuity of activities (Chrzanowski & Johnston, 1986; Reda, 1990). Current research on the methods of scheduling repetitive projects most often focus on the problem of optimal creation of their schedules. In the optimization models of these projects, different parameters, constraints and accepted criteria may be taken into account. Due to their nature, while optimizing their schedules, mathematical methods are currently most commonly used (e.g. linear programming (e.g. Reda, 1990; Ipsilandis, 2007; Biruk & Jaśkowski, 2017; Radziszewska-Zielina & Sroka, 2017a, 2017b, 2018), dynamic programming (e.g.

*Corresponding author. E-mail: *michal.podolski@pwr.edu.pl*

Moselhi & Hassanein, 2003), neural networks (e.g. Adeli & Karim, 1997)) or metaheuristics (e.g. genetic (Hegazy, Elhakeem, & Elbeltagi, 2004; Agrama, 2014), evolutionary (e.g. Rogalska, Bożejko, & Hejducki, 2008), hybrid (e.g. Ezeldin & Soliman, 2009), simulated annealing (e.g. Chen & Shahandashti, 2007) algorithms). They are mostly used to solve optimization problems related to, among others, the duration of the project, its cost, taking into account the condition of the imposed deadline for its completion, the total time of downtime of the construction crews carrying out the project.

As part of repetitive undertakings, multiunit projects can be distinguished, which include the construction of many residential, commercial, industrial buildings or engineering structures. The most common assumption in this type of projects is the constant and unchanging sequence of the implementation of units. This assumption may be dictated by existing organizational constraints or limitations imposed by the investor in connection with, for example, a specific sales order of the project units. However, in the general case, the order can be arbitrary. Assuming any size of works that make up a particular object, we get the opportunity to create an optimal schedule for a given multiunit construction project. In this case, the contemporary achievements of the job scheduling theory in the field of flow shop problems will be helpful (Gupta & Stafford, 2006), which are NP-hard already for $n = 3$ objects of the project. These problems are mainly used in electronics, the chemical industry, the automotive industry, and less frequently in construction (Bożejko, Hejducki, & Wodecki, 2012; Bożejko, Hejducki, Uchroński, & Wodecki, 2014; Podolski, 2016, 2017). The article (Bożejko et al., 2012) presents the problem of a multiunit construction projects, in which a solution was sought to minimize costs, including only penalties for exceeding the imposed deadlines for the construction of units. The solution, which was the order of units' realization, was obtained using the scatter search algorithm. Bożejko et al. (2014) presents the problem of a multiunit construction project, in which there was the possibility of overlap of works performed in one unit, i.e. there was, in certain periods of time, the possibility of simultaneously carrying out two different works in one unit. The solution, which was also the order of units' realization, was obtained using the construction and tabu search algorithms. In the article (Podolski, 2016) a discrete problem of optimization of a two-criteria cost/time was presented, in which the decision variables were: the order (permutation) of realization of units (buildings) and a discrete set of ways (subcontractors' offers) of works realization. Each variant included exact cost and time of the works in a given unit. The optimization task consisted in solving the single-criterion task of discrete optimization – minimizing the cost of the project with the specified limitation for the time of its realization (the sum of the selected subcontractors' offers). This problem was solved using a developed by author, two-stage, suboptimal algorithm using the simulated annealing algorithm. In the article (Podolski, 2017) a model of scheduling a multiunit project was presented, in which there is a possibility of applying more than one working group to perform a given work in the unit and order dependencies between works in a given unit in the form of a network as in the CPM method. The task of the minimization of the project duration was solved using the program developed by author in the *Mathematica* system, using the tabu search algorithm similarly to the flow shop problems with the parallel machines (hybrid flow shop).

Due to the specificity of construction works carried out by construction crews, actual schedules of multiunit construction projects should not only take into account the real costs of works implementation, but also be subject to specific constraints. These include conditions regarding the dates of completion of the facilities imposed by the investor, additional costs caused by the failure to maintain continuity of works performed by construction crews. The work will present a new flow shop model of a multiunit construction project, in which the actual (direct and indirect) costs of the project's implementation are taken into account, with reference to: costs of missing deadlines for construction of units, costs associated with failure to keep the continuity of works being done and variable sequence of units realization. Consideration of the aforementioned parameters and limitations required the use of a metaheuristic simulated annealing algorithm. While calculating the value of the objective function in this algorithm, which was the total cost of a multiunit construction project, the cost optimization of the schedule was made. A novelty in this article in relation to the previous ones is the modeling of cost / time dependence as a continuous function, and the assume of the actual costs of a multiunit project (direct costs of the works, indirect costs, costs of failure to meet the deadlines for the realization of works in the units, and costs of downtime for working groups) in combination with a discrete function of the order of realized units. In this article, due to the lack of research in the literature regarding the presented form of the simulated annealing algorithm, an experimental analysis of the developed metaheuristic confirming the high efficiency of the optimization algorithm was presented. The presented optimization model of a multiunit project was illustrated by a computation example.

## 1. Description of the multiunit construction project considered in the work

The multiunit construction project model considered in the work can be regarded as deterministic, when all parameters and constraints of the model are precisely defined and known from the beginning. Each of the project units requires the same set of works to be done, which are carried out in the same order. For these units, the following works will be performed successively in line with the same technology, for instance earthworks, foundation works, walls with ceilings, roof trusses. The fixed sequence

of works is characteristic of buildings with a simple construction such as: single-family houses, simple industrial or engineering structures. Works in the units will be carried out by construction crews specialized to perform only one type of work, moving from the previous unit to the next one. It is assumed that the size of units included in the project will be arbitrary. They are not characterized by proportionality between the size of units and the same labor consumption of one kind of works. It follows from the above conclusion that the shape of the schedule of a multiunit project defined in this way will be influenced by the order in which the units are constructed. In the considered project model, each of the works of a given unit will be described by its normal duration and its normal cost. It is assumed that it is possible to shorten the normal time of a given work to the border time using more efficient employees, technologies, the involvement of additional workers or machines. It is connected with raising the cost of its implementation to the boundary cost. It is assumed that the dependence of the cost per time for a given work in the project site is linear. For the contractor of the project, the criterion of choosing the optimal schedule will be the total cost of the project. It will include a direct cost understood as the sum of costs of all works in project, indirect costs depending on the duration of the project, penalties for failure to comply with the deadlines for the implementation of units imposed by the investor and additional costs of the contractor due to discontinuity of works of one type in units by individual construction crews.

## 2. Optimization model of the multiunit construction project under consideration

The following optimization model of the multiunit construction project considered in the work was defined:

Parameters:
- The project constitutes a set of units (buildings, structures) $Z = \{Z_1, Z_2, Z_3, ..., Z_i, ..., Z_n\}$.
- In order to perform the works there are construction crews doing work of one kind and forming a set of $B = \{B_1, B_2, B_3, ..., B_j, ..., B_m\}$.
- Each unit $Z_i \in Z$ requires execution of works that form a set of $O_i = \{O_{i1}, O_{i2}, O_{i3}, ..., O_{ij}, ..., O_{im}\}$.
- Work $O_{ij} \in O_i$ can only be carried out by the crew $B_j$. The normal duration of the work $O_{ij}$ performed by the crew is $tn_{i,j} > 0$. The normal duration of work $O_{ij}$ corresponds to the normal cost of its implementation equal to $kn_{i,j} > 0$. Due to the change in the amount of resources used or the change in the technology of works, it is possible to shorten each normal time $tn_{i,j}$ to the value of the boundary time $tb_{i,j} > 0$, where $tb_{i,j} < tn_{i,j}$, which involves raising the cost to boundary cost equaling $kb_{i,j} > kn_{i,j}$.
- Indirect unit cost per day is assumed at the level of $ki$.
- The implementation dates of each of the $n$ units imposed by the investor are assumed and equals $Td_i$.

- Unit costs are assumed for failing to meet the deadlines of the individual units per day and equals $kl_i$.
- Unit costs of discontinuities (downtimes) of crews are assumed and equal $kc_j$.

Constraints:
- The order in which the works are performed resulting from the technology is assumed:
$$O_{i,j-1} \prec O_{i,j} \prec O_{i,j+1}.$$
- It is assumed that at any time, each crew $B_j$ can perform only one work at a time and that only one work $O_{ij}$ can be performed at any time in the unit $Z_i$.
- It is assumed that the $O_{ij} \in O_i$ work is carried out continuously by the crew $B_j$.

For such adopted parameters and constraints there is a mathematical programming model formulated, in the following form:
- Objective function:

$$K(\pi) = K_d + K_i + K_l + K_c \rightarrow \min; \qquad (1)$$

$$K_d = \sum_{i=1}^{n} \sum_{j=1}^{m} kd_{i,j}(t_{i,j}); \qquad (2)$$

$$K_i = Tf_{n,m} \cdot ki; \qquad (3)$$

$$K_l = \sum_{i=1}^{n} l_i \cdot kl_i; \qquad (4)$$

$$K_c = \sum_{j=1}^{m} \left( Tf_{n,j} - Tf_{1,j} - \sum_{i=2}^{n} t_{i,j} \right) \cdot kc_j. \qquad (5)$$

- Decision variables: $t_{i,j}$, $Tf_{i,j}$, $l_i$, $\pi$, where:

$$tb_{i,j} \leq t_{i,j} \leq tn_{i,j}; \qquad (6)$$

$$Tf_{1,1} \geq t_{1,1}; \qquad (7)$$

$$Tf_{i,j} \geq Tf_{i-1,j} + t_{i,j}; \qquad (8)$$

$$Tf_{i,j} \geq Tf_{i,j-1} + t_{i,j}; \qquad (9)$$

$$Tf_{i,m} - l_i \leq Td_i; \qquad (10)$$

$$Tf_{i,j} \geq 0; \qquad (11)$$

$$t_{1,1} \geq 0; \qquad (12)$$

$$l_i \geq 0. \qquad (13)$$

The objective function (1) is the sum of direct costs $K_d$, indirect costs $K_i$, costs related to failure to meet the deadlines for the units $K_l$ and the costs of downtime for crews $K_c$. This cost should be minimized. Direct costs (2) are the sum of the costs of all work performed by all crews in all units depending on the duration of the work. The function of works costs depending on the duration of this activity $kd_{i,j}(t_{i,j})$ is a decreasing linear function. In fact, the dependence of direct costs on time is a discrete dependence determined on the basis of negotiations between the subcontractor and the general contractor. The linear dependence between the time of work and its cost

is a fairly good approximation. The equation of the linear function of the direct cost of a given work to its duration can be determined from the Eqn (14), while the value of the angular coefficient and the free expression from the Eqns (15) and (16).

$$kd_{i,j}(t_{i,j}) = a_{i,j} \cdot t_{i,j} + b_{i,j}; \tag{14}$$

$$a_{i,j} = \frac{kb_{i,j} - kn_{i,j}}{tb_{i,j} - tn_{i,j}}; \tag{15}$$

$$b_{i,j} = kb_{i,j} - a_{i,j} \cdot tb_{i,j}. \tag{16}$$

Indirect costs (3) depend on the amount of indirect unit costs and the duration of the entire project $Tf_{n,m}$. Costs related to non-compliance with the directive deadlines (4) are calculated as the sum of unit cost products for non-compliance with the deadline and the duration of this delay for all units. The unit cost for non-compliance with the directive deadline may vary for each of the units. Penalties are specified in the contract between the investor and the general contractor and usually amount to between 0.05% and 0.2% of the gross contract value for each day of delay. Costs of downtime for crews (5) are calculated as a sum of product of unit cost for downtime in crews work and duration of downtime for all crews. Duration of downtime is calculated on the basis of completion date of crew work in all units, date of termination of work on the first unit and duration of crew work on remaining units.

The decision variables in the model are: $t_{i,j}$, denotes the assumed duration of work $O_{i,j}$, $Tf_{i,j}$ denotes the deadline for completion of work $O_{i,j}$, $l_i$ denotes the time of exceeding the directive deadline for individual units, and $\pi$ denotes the order of executing units from the set $Z$: $\pi = (\pi(1), \pi(2), ..., \pi(i), ..., \pi(n))$.

The model includes the following constraints. Eqn (6) limits the duration of activities to the interval between the limit (minimum) and normal (maximum) time. Eqns (7), (8), (9) specify the assumptions regarding the order of works execution. The deadline for completion of work on the first site by the first working group must be greater than the duration of this work (7). The deadline for completing the work of the $j$-th crew on the $i$-th unit must be larger than the completion date of the work of the same crew in the previous unit enlarged by the time of duration of the crew's work in the $i$-th unit (8). The deadline for completing the work of the $j$-th crew on the $i$-th object must be larger than the completion date of the work of the previous crew in the same unit enlarged by the time of duration of work in this unit by the $j$-th crew (9). The condition (10) allows its users to determine the value of the variable $l_i$, which determines the time of exceeding the directive deadlines. Conditions (11), (12) and (13) limit the values of decision variables to non-negative numbers.

The duration of activities and the corresponding costs of performing activities are linear in nature. For the set order of unit execution, the presented optimization model becomes a linear programming model that can be solved using, for example, the Simplex method. However, considering any order $\pi$ in which units are executed causes the fact that the presented model becomes an NP-hard problem of discrete optimization, which has not been examined in field of scheduling of construction project yet. The NP-hardness of the problem in the model under consideration results from the following assumption: $tn_{i,j} = tb_{i,j}$, $kn_{i,j} = kb_{i,j} = 0$, $ki = 0$, $kl_i = 0$, $kc_j = 0$, and the objective function is $Tf_{n,m}$, whereas we are looking for its minimum value. Taking into consideration the above assumptions we obtain a classic permutational flow shop problem $FP||C_{max}$, which is strongly NP-hard (Gupta & Stafford, 2006). In order to find the optimal solution, the considered model requires the use of an approximate algorithm that will allow us to obtain solutions close to optimal. In this work, a metaheuristic simulated annealing algorithm was used to obtain solutions in the model.

## 3. Method for solving the optimization problem

The problem presented above in the optimization model is an NP-hard problem of discrete optimization. An important difficulty in solving it is the fact that two different groups of decision variables can be distinguished in it. The first of them is related to the selection of optimal durations of the works $t_{i,j}$ and the corresponding costs of performing the activities in the project taking into account the accepted objective function of the total cost. Due to the continuous and linear nature of these variables, this problem will be solved using the Simplex method, which in the considered model allows us to obtain an optimal solution. Such a solution in the form of optimal durations of activities $t_{i,j}$ is possible only for the determined order of executing construction units $\pi$. The second group of decision variables is represented by the order in which units are executed. The nature of this decision variable is discrete, and the problem of searching for the optimal order of performing objects is NP-hard. To solve this problem, an approximate, metaheuristic simulated annealing (SA) algorithm was chosen because of its confirmed high effectiveness in solving of optimization tasks in multiunit construction projects (Podolski, 2008).

SA algorithm has been proposed in the work of Kirkpatrick, Gelatt, and Vecchi (1983). This algorithm uses analogous to the thermodynamic process of cooling the solid in order to introduce the trajectory of the search of the local extremum. States of solid matter are seen analogously as individual solution to the problem, whereas the energy of the body as the value of the objective function. During the physical process of cooling the temperature is reduced slowly in order to maintain energy balance. The SA algorithm starts with the initial solution, usually chosen at random. Then, in each iteration, according to established rules or randomly, there is solution $\pi'$ selected from the base neighborhood $\pi$. It becomes the base solution in the next iteration, if the value of the objective function is better than the current base solution or if it otherwise may become the base solution with the probability of:

$p = \exp(-\Delta/T_i)$, where $\Delta = c(\pi') - c(\pi)$, $T_i$ – the temperature of the current iteration $i$, $c$ – the objective function. In each iteration there are $m$ draws from the neighbourhood of the current basic solution performed. The parameter called the temperature decreases in the same way as in the natural process of annealing. The most frequently adopted patterns of cooling are:

– geometrical $T_{i+1} = \lambda_i T_i$ ,

– logarithmic $T_{i+1} = T_i / (1 + \lambda_i T_i)$,

where $i = 0, ..., N - 1$, $T_0$ – the initial temperature, $T_N$ – the final temperature, $N$ – number of iterations, $\lambda_i$ – parameter. In the algorithm there are usually parameter values $T_0$, $T_N$, $N$ adopted and parameter $\lambda_i$ is calculated. The relationship $T_0 > T_N$ should take place, whereas $T_N$ should be small, close to zero. Below, there is presented a general method of SA algorithm used to solve the flow shop problem.

**Step 0.** Determine the initial solution $\pi^0 \in \Pi$. Substitute $\pi_{SA} = \pi^0$, $k = 0$, $T = T_0$.

**Step 1.** Perform steps 1.1 – 1.3 $x$-times.
  Step 1.1. Substitute $k := k + 1$. Choose random $\pi \in N(V, \pi^{k-1})$.
  Step 1.2. If $c(\pi) < c(\pi_{SA})$ then substitute $\pi_{SA} = \pi$.
  Step 1.3. If $c(\pi) < c(\pi_{k-1})$ then substitute $\pi_k = \pi$. Otherwise, accept solution $\pi$ with a probability of $p = \exp((c(\pi_{k-1}) - c(\pi))/T_{,tj}$. $\pi_k = \pi$, if solution $\pi$ was not accepted.

**Step 2.** Change the temperature $T$ according to a defined pattern of cooling.

**Step 3.** If $T > T_N$, return to step 1, otherwise STOP.

SA algorithms are used to solve many optimization problems, including flow shop problems considered in the context of discrete optimization problems (e.g. Ogbu & Smith, 1995; Ishibuchi, Misaki, & Tanaka, 1995). Good results obtained in applications allow us to treat SA algorithms as one of the strongest tools. The optimization problem considered in the work will be solved using the algorithm presented in Figure 1. Its steps are consistent with the above-mentioned SA algorithm. When calculating the assumed objective function for a given $\pi$ sequence, the problem of minimizing this function is optimally solved using the Simplex method.

In the SA algorithm used in the work, the following assumptions regarding its parameters were adopted:
  – the neighbourhood $N(\pi)$ contains permutations generated from $\pi$ using the "insert" move type,
  – the Boltzmann acceptance function was used,
  – a geometric diagram of cooling was adopted, i.e. $T_{i+1} = \lambda T_i$, the initial temperature $T_0 = 30$, $\lambda = 0.99$, the number of solutions considered at a fixed temperature – 300, minimum temperature $T_N = 0.05$.

The presented algorithm has been implemented in the Python programming language using the PyMathProg linear optimization package, which uses the GLPK (GNU Linear Programming Kit) engine.
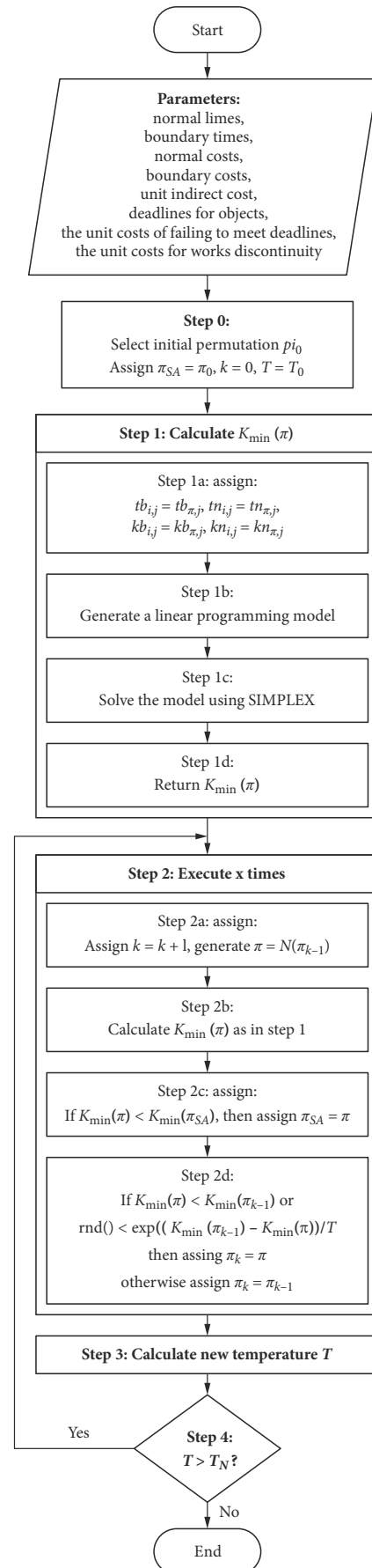


Figure 1. Block diagram of the applied SA algorithm

## 4. Verification of the results obtained with the use of the applied SA algorithm

For the approximate SA algorithm presented above, the authors verified the results obtained with its help. For this purpose, two groups of examples of objects were generated randomly: $n = 7$ objects, $m = 5$ works and $n = 10$ objects, $m = 10$ works. These sizes are similar to the sizes of problems that can be encountered in the practice of multiunit construction projects. There were five examples generated in each group. In each of the examples, a minimum value of the cost of a given project was sought in accordance with the model presented in the work. Each of the examples was resolved five times. The results obtained with the help of the created software are presented in Table 1. Then, the two groups of examples analyzed here were solved optimally by means of the exhaustive search algorithm (ES algorithm). The obtained results were compared with each other by calculating the average relative error $PRD$(SA) of the SA algorithm:

$$PRD(SA) = 100\%(K^{SA} - K^{ES})/K^{ES}, \quad (17)$$

where: $K^{SA}$ – the value of the adopted objective function obtained by means of the SA algorithm, $K^{ES}$ – the value of the assumed objective function obtained by means of an exhaustive search algorithm. The average relative errors of the applied SA algorithm are presented in Table 1. The values of these errors are small and do not exceed 0.5%, which confirms the high effectiveness of the SA algorithm in searching for optimal solutions in the subject.

## 5. Case study

The contractor, at the request of the investor, is to carry out a project consisting in the construction of $n = 12$ residential buildings (in the model called units). Each of them requires the execution of $m = 9$ works executed in a fixed order. The undertaking will be implemented entirely by subcontractors (in the model called construction crews). For each type of works, the contractor received from subcontractors the range of deadlines in which the work in a given building can be performed, and the corresponding costs. This is shown in Table 2. The scope of deadlines is defined by the normal $tn_{i,j}$ and boundary $tb_{i,j}$ date of the execution of a given type of work in individual buildings expressed in working days and corresponding to the normal term – normal cost $kn_{i,j}$ and the boundary date – boundary cost $kb_{i,j}$. Each work can be executed at any time between the limit time and normal time at proportional costs. The contractor also incurs indirect costs related to running and supervising the construction in the amount of 300 € for each day of the project implementation. The contractor has a deadline for the implementation of individual buildings imposed by the investor. The directive deadlines are for subsequent objects (in working days): 160, 200, 240, 280, 320, 360, 400, 440, 480, 520, 560, 600. For failure to meet the deadline, a penalty of 200 € is imposed for the general contractor for each day of delay. In addition, in connection with the concluded contracts with subcontractors, the contractor undertook to ensure the continuity of subcontractors' work $j = 4$ (roof truss,

Table 1. Outcomes of the verification of the results obtained with the use of the SA algorithm

| Example name | Value of the cost $K^{SA}$ obtained with the use of SA algorithm in trial $i$ = | | | | | The average value of the cost $K^{SA}$ | The minimum value of cost $K^{ES}$ obtained with the use of ES algorithm | $PRD$(SA) [%] |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | | | |
| Example $n = 7$ objects. $m = 5$ works | | | | | | | | |
| 7×5_1 | 446921.45 | 442174.65 | 442174.65 | 440752.61 | 440752.61 | 442555.20 | 440752.61 | 0.409 |
| 7×5_2 | 368618.44 | 368618.44 | 368618.44 | 368618.44 | 368618.44 | 368618.44 | 368618.44 | 0.000 |
| 7×5_3 | 282900.16 | 282900.16 | 282900.16 | 282900.16 | 282900.16 | 282900.16 | 282900.16 | 0.000 |
| 7×5_4 | 285931.69 | 287215.79 | 285781.87 | 286511.55 | 285696.49 | 286227.48 | 285150.08 | 0.378 |
| 7×5_5 | 257543.36 | 257543.36 | 257543.36 | 257543.36 | 257543.36 | 257543.36 | 257543.36 | 0.000 |
| Mean $PRD$(SA) for the size 7×5 [%]: | | | | | | | | 0.157 |
| Example $n = 10$ objects. $m = 10$ works | | | | | | | | |
| 10×10_1 | 573764.04 | 573764.04 | 573764.04 | 573764.04 | 573764.04 | 573764.04 | 573764.04 | 0.000 |
| 10×10_2 | 550313.84 | 550452.97 | 550313.84 | 550452.97 | 550313.84 | 550369.49 | 550313.84 | 0.010 |
| 10×10_3 | 845382.82 | 845508.93 | 845508.93 | 845508.93 | 845508.93 | 845483.71 | 845382.82 | 0.012 |
| 10×10_4 | 1062027.95 | 1062027.95 | 1062027.95 | 1062027.95 | 1062027.95 | 1062027.95 | 1062027.95 | 0.000 |
| 10×10_5 | 648180.24 | 648180.24 | 648180.24 | 648180.24 | 648180.24 | 648180.24 | 648180.24 | 0.000 |
| mean $PRD$(SA) for the size 10×10 [%]: | | | | | | | | 0.004 |
| mean $PRD$(SA): | | | | | | | | 0.081 |

cover), $j = 5$ (installations), $j = 7$ (plaster, floors, attic), $j = 9$ (tiling, painting, sanitary assembly). The penalty for each day of discontinuity of work of contractors has been determined with subcontractors and amounts to 300 € for contractors 4 and 5 and 200 € for contractors 7 and 9. Additional technological and organizational limitation is that the work cannot start if the work of the same type in the previous building did not end and if the previous work in the same building was not completed. The number of possible schedules due to the order in which the buildings are realized in the example shown is $12! \approx 4.8 \cdot 10^8$. The number of possible schedules due to the continuity of the cost-time dependency is infinite. Due to the possibility of choosing the time of works' execution, taking into account indirect costs, imposed dates of completion of individual buildings and costs to be incurred due to the lack of continuity of subcontracting work, the contractor will seek to set a schedule to minimize the total project costs.

Table 2. Ranges of execution times and costs of works in $n = 12$ residential buildings for $m = 9$ works. Units: for $tn_{i,j}$ and $tb_{i,j}$ – working days; for $kn_{i,j}$ and $kb_{i,j}$ – thousand euro

| Number and type of work $j$ | | | Building $i =$ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $j =$ | Type of work | Data | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | A – earthworks | $tn_{i,j}$ | 12 | 15 | 15 | 16 | 9 | 11 | 14 | 13 | 12 | 13 | 16 | 10 |
| | | $tb_{i,j}$ | 10 | 10 | 10 | 15 | 5 | 5 | 13 | 12 | 10 | 12 | 15 | 6 |
| | | $kn_{i,j}$ | 1.28 | 1.28 | 1.28 | 1.51 | 1.05 | 1.51 | 1.98 | 0.81 | 1.74 | 0.81 | 1.98 | 1.51 |
| | | $kb_{i,j}$ | 1.40 | 1.51 | 1.51 | 1.86 | 1.40 | 1.86 | 2.33 | 1.28 | 2.33 | 1.16 | 2.44 | 1.98 |
| 2 | B – foundations | $tn_{i,j}$ | 18 | 24 | 15 | 14 | 16 | 18 | 18 | 18 | 18 | 19 | 17 | 17 |
| | | $tb_{i,j}$ | 17 | 19 | 10 | 10 | 12 | 9 | 13 | 13 | 11 | 17 | 14 | 13 |
| | | $kn_{i,j}$ | 5.58 | 3.95 | 5.58 | 8.14 | 6.05 | 3.95 | 5.81 | 7.67 | 5.58 | 5.58 | 5.35 | 8.14 |
| | | $kb_{i,j}$ | 5.93 | 5.58 | 6.98 | 9.77 | 7.79 | 4.42 | 7.09 | 10.47 | 8.26 | 7.21 | 6.51 | 10.70 |
| 3 | C – walls, ceilings | $tn_{i,j}$ | 40 | 48 | 36 | 34 | 28 | 30 | 41 | 44 | 41 | 47 | 52 | 45 |
| | | $tb_{i,j}$ | 22 | 38 | 28 | 17 | 26 | 26 | 25 | 38 | 26 | 29 | 33 | 34 |
| | | $kn_{i,j}$ | 15.23 | 16.86 | 8.26 | 9.42 | 9.19 | 20.35 | 7.56 | 11.74 | 12.91 | 8.95 | 18.95 | 16.40 |
| | | $kb_{i,j}$ | 15.93 | 18.84 | 11.74 | 12.09 | 11.16 | 25.70 | 10.23 | 11.86 | 13.84 | 11.51 | 23.37 | 20.12 |
| 4 | D – roof truss, roofing | $tn_{i,j}$ | 13 | 11 | 16 | 9 | 16 | 11 | 15 | 13 | 12 | 9 | 13 | 16 |
| | | $tb_{i,j}$ | 12 | 5 | 15 | 6 | 13 | 8 | 10 | 10 | 11 | 6 | 12 | 12 |
| | | $kn_{i,j}$ | 8.26 | 11.51 | 11.98 | 4.77 | 12.21 | 8.72 | 11.74 | 4.77 | 5.93 | 7.09 | 9.42 | 11.98 |
| | | $kb_{i,j}$ | 10.35 | 14.19 | 13.26 | 6.98 | 17.44 | 11.51 | 14.77 | 4.88 | 8.26 | 7.56 | 9.77 | 13.26 |
| 5 | E – installations | $tn_{i,j}$ | 24 | 20 | 22 | 24 | 28 | 24 | 18 | 21 | 26 | 17 | 22 | 20 |
| | | $tb_{i,j}$ | 16 | 14 | 15 | 17 | 22 | 17 | 9 | 18 | 19 | 13 | 16 | 18 |
| | | $kn_{i,j}$ | 10.70 | 14.65 | 8.84 | 8.37 | 13.02 | 10.93 | 11.40 | 9.77 | 11.63 | 11.63 | 14.88 | 9.53 |
| | | $kb_{i,j}$ | 12.56 | 20.12 | 11.98 | 8.49 | 16.98 | 16.16 | 15.35 | 11.05 | 12.67 | 14.77 | 18.60 | 12.33 |
| 6 | F – door and window carpentry | $tn_{i,j}$ | 10 | 9 | 11 | 11 | 13 | 10 | 12 | 9 | 12 | 9 | 11 | 10 |
| | | $tb_{i,j}$ | 7 | 7 | 5 | 7 | 9 | 6 | 6 | 5 | 7 | 7 | 5 | 9 |
| | | $kn_{i,j}$ | 8.84 | 5.12 | 8.37 | 10.70 | 4.65 | 13.26 | 5.81 | 10.00 | 5.81 | 10.00 | 5.12 | 9.53 |
| | | $kb_{i,j}$ | 10.70 | 6.74 | 9.77 | 14.77 | 6.16 | 13.95 | 6.40 | 10.35 | 6.86 | 12.56 | 5.70 | 12.33 |
| 7 | G – plaster, floors, attic | $tn_{i,j}$ | 26 | 28 | 29 | 21 | 30 | 18 | 27 | 20 | 26 | 19 | 26 | 35 |
| | | $tb_{i,j}$ | 22 | 23 | 26 | 16 | 20 | 13 | 18 | 15 | 17 | 14 | 15 | 26 |
| | | $kn_{i,j}$ | 6.28 | 5.35 | 5.35 | 9.30 | 7.67 | 3.02 | 5.12 | 4.65 | 7.21 | 9.53 | 7.67 | 4.65 |
| | | $kb_{i,j}$ | 8.14 | 7.33 | 8.02 | 11.98 | 9.53 | 4.53 | 6.05 | 4.77 | 10.35 | 10.70 | 10.47 | 6.63 |
| 8 | H – fences, driveways | $tn_{i,j}$ | 12 | 14 | 16 | 11 | 13 | 14 | 16 | 9 | 11 | 12 | 10 | 14 |
| | | $tb_{i,j}$ | 9 | 9 | 9 | 5 | 9 | 8 | 13 | 5 | 10 | 9 | 9 | 8 |
| | | $kn_{i,j}$ | 5.00 | 4.07 | 5.23 | 7.33 | 6.16 | 3.37 | 7.56 | 2.91 | 2.67 | 6.86 | 5.93 | 4.07 |
| | | $kb_{i,j}$ | 5.35 | 5.58 | 5.70 | 7.79 | 7.44 | 3.84 | 9.88 | 3.49 | 4.07 | 8.84 | 6.28 | 5.58 |
| 9 | J – tiling, painting, sanitary assembly | $tn_{i,j}$ | 18 | 24 | 17 | 18 | 17 | 18 | 13 | 15 | 18 | 20 | 21 | 14 |
| | | $tb_{i,j}$ | 10 | 20 | 15 | 17 | 13 | 15 | 7 | 12 | 15 | 12 | 20 | 10 |
| | | $kn_{i,j}$ | 9.65 | 8.95 | 8.49 | 12.91 | 7.56 | 13.37 | 11.28 | 14.07 | 13.84 | 6.86 | 5.70 | 7.79 |
| | | $kb_{i,j}$ | 10.47 | 13.14 | 9.30 | 16.05 | 10.93 | 17.44 | 15.47 | 17.91 | 14.53 | 9.53 | 6.74 | 9.77 |

For the initial schedule of $\pi_0$ = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12) and assumed times of normal execution of all works, total cost of 1292.91 thousand of euro was obtained. The duration of the entire project is at the level of 625 days. The schedule for the initial $\pi_0$ rank without optimization is shown in Figure 2(a).

For the initial schedule of $\pi_0$ after solving the linear optimization task using the Simplex method and the created software, total costs of 1065.70 thousand Euro were obtained with a realization time of 457 days. This means a 17.6% reduction in costs and an acceleration of the project implementation time by 26.9% compared to the solution without the use of optimization. The schedule for the initial order $\pi_0$ after linear optimization is shown in Figure 2(b).

The next step in the search for an optimal solution was the use of the SA algorithm in order to find the optimal schedule of the project that minimizes the assumed cost goal function, taking into account the possible change in the order of construction objects execution. The stop condition for the algorithm's operation was to achieve the assumed minimum temperature $T_{min}$ = 0.05. The algorithm performed 191100 iterations in the SA loop, which is less than 0.04% of all possible scheduling. The run of SA algorithm is presented in Figure 3. The smallest cost

of the project is 1045.28 thousand Euro obtained for the following scheduling of objects: $\pi_{SA}$ = (6, 7, 10, 2, 3, 9, 1, 5, 11, 12, 4, 8). The deadline for the project implementation was 438 days. In relation to the optimal solution for the initial scheduling $\pi_0$, the total cost was improved by 1.9% (20.42 thousand Euro) and the deadline for implementation was improved by 4.2% (19 days). In relation to the
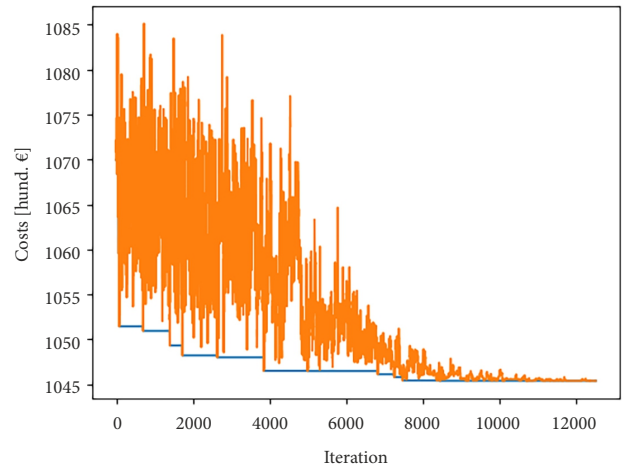


Figure 3. Typical run of the SA algorithm in the case study. The best solution obtained currently indicated in blue
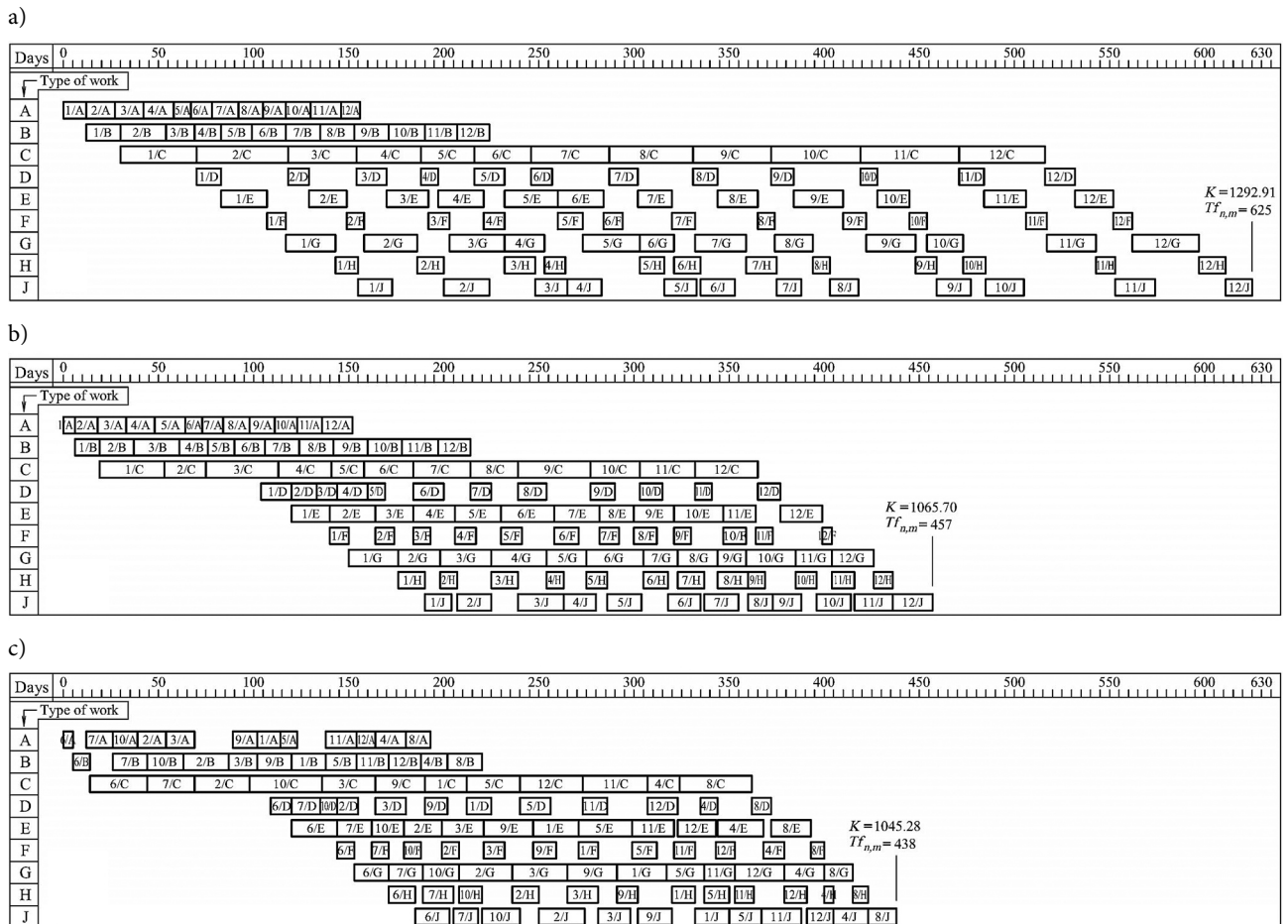
a)



b)



c)



Figure 2. Schedules for the implementation of project in the case study: (a) – for $\pi_0$ = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12) without optimization; (b) – for $\pi_0$ with optimization; (c) – for $\pi_{SA}$ = (6, 7, 10, 2, 3, 9, 1, 5, 11, 12, 4, 8) with optimization using SA algorithm

initial schedule (without optimization) for the schedule $\pi_0$, the project costs were reduced by 19% and the implementation time decreased by 30%. The obtained schedule is shown in Figure 2(c).

## Conclusions

Scheduling of multiunit construction projects, which is a case of repetitive projects, is a problem in which optimization tasks often occur. Due to the specific character of multiunit construction projects, decision variables in such undertakings may be continuous or discrete. An important difficulty in finding solutions introduces simultaneous occurrence of a continuous and discrete decision variables in the project optimization model, which took place in the model presented in the work. The discreteness of the decision variable – the order in which objects are realized in a multiunit project brought the presented model to the NP-hard optimization problem, i.e. the permutational flow shop problem. The search for solutions in it required the use of a metaheuristic simulated annealing algorithm with simultaneous solving the linear optimization task using the Simplex method while calculating the value of the adopted objective function – the total cost of the project. Due to the lack of research in the literature regarding the applied form of the simulated annealing algorithm, the article presents verification of the results provided with its use. It showed its high effectiveness in searching for optimal solutions in the model in question.

The presented model of multiunit construction project scheduling can be used when determining the optimal work schedule for construction companies undertaking such projects. It enables to take into account the situation when companies use their own working crews to implement the works or intend to select specialized subcontractors who best meet the imposed conditions. The developed model can be easily implemented in scheduling programs and can support the planner when designing multiunit construction projects, helping to minimize the cost of the planned investment realization. Due to the development of dedicated software for solving optimization tasks, it is possible to extend the current model with additional technological and organizational constraints, additional factors affecting the cost of the project and taking into account other objective functions.

## Author contributions

Michał Podolski was responsible for literature review, description of the model, partially for optimization model, description of applied SA algorithm, concept of the algorithm verification and wrote the introduction, sections 1, 3 and 4, partially sections 2 and 5, conclusions of the article. Bartłomiej Sroka was responsible for concept, computer implementation, verification and case study of the presented model, partially for optimization model, wrote partially sections 2 and 5.

## Disclosure statement

Authors have not any competing financial, professional, or personal interests from other parties.

## References

Adeli, H., & Karim, A. (1997). Scheduling/cost optimization and neural dynamic model for construction. *Journal of Construction Engineering and Management*, *123*(4), 450-458. https://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. https://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. https://doi.org/10.1061/(ASCE)0733-9364(2002)128:6(545)

Biruk, S., & Jaśkowski, P. (2017). Scheduling linear construction projects with constraints on resource availability. *Archives of Civil Engineering*, *63*(1), 3-15. https://doi.org/10.1515/ace-2017-0001

Bożejko, W., Hejducki, Z., & Wodecki, M. (2012). Applying metaheuristic strategies in construction projects management. *Journal of Civil Engineering and Management*, *18*(5), 621-630. https://doi.org/10.3846/13923730.2012.719837

Bożejko, W., Hejducki, Z., Uchroński, M., & Wodecki, M. (2014). Solving resource-constrained construction scheduling problems with overlaps by metaheuristic. *Journal of Civil Engineering and Management*, *20*(5), 649-659. https://doi.org/10.3846/13923730.2014.906496

Chen, P. H., & Shahandashti, S. M. (2007). Simulated annealing algorithm for optimizing multi-project linear scheduling with multiple resource constraints. In *Proceedings of 24th International Symposium on Automation and Robotics in Construction* (*ISARC 2007*) (pp. 429-434). Kochi, India. https://doi.org/10.22260/ISARC2007/0071

Chrzanowski, E. N., & Johnston, D. (1986). Application of linear construction. *Journal of Construction Engineering and Management*, *112*(4), 476-491. https://doi.org/10.1061/(ASCE)0733-9364(1986)112:4(476)

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. https://doi.org/10.1061/(ASCE)0733-9364(2009)135:1(42)

Gupta, J., & Stafford, E. F. Jr. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, *169*(3), 699-711. https://doi.org/10.1016/j.ejor.2005.02.001

Harmelink, D. J., & Rowings, J. E. (1998). Linear scheduling model: Development of controlling activity path. *Journal of Construction Engineering and Management*, *124*(4), 266-268. https://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. https://doi.org/10.1061/(ASCE)0733-9364(1998)124:4(269)

Hegazy, T., Elhakeem, A., & Elbeltagi, E. (2004). Distributed scheduling model for infrastructure networks. *Journal of Construction Engineering and Management*, *130*(2), 160-167. https://doi.org/10.1061/(ASCE)0733-9364(2004)130:2(160)

Ipsilandis, P. G. (2007). Multiobjective linear programming model for scheduling linear repetitive projects. *Journal of Construction Engineering and Management*, *133*(6), 417-424. https://doi.org/10.1061/(ASCE)0733-9364(2007)133:6(417)

Ishibuchi, H., Misaki, S., & Tanaka, H. (1995). Modified simulated annealing algorithms for the flow shop sequencing problem. *European Journal of Operational Research*, *81*, 388-398. https://doi.org/10.1016/0377-2217(93)E0235-P

Kirkpatrick, S., Gelatt, C. D., & Vecchi M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671-680. https://doi.org/10.1126/science.220.4598.671

Moselhi, O., & Hassanein, A. (2003). Optimised scheduling of linear projects. *Journal of Construction Engineering and Management*, *129*(6), 667-673. https://doi.org/10.1061/(ASCE)0733-9364(2003)129:6(664)

Ogbu, F. A., & Smith, D. K. (1995). The application of the simulated annealing algorithm to the solution of the n/m/Cmax flowshop problem. *Computers* & *Operations Research*, *17*(3), 243-253. https://doi.org/10.1016/0305-0548(90)90001-N

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). Retrieved from https://dbc.wroc.pl/Content/2515/PDF/Podolski_Analiza_PhD.pdf

Podolski, M. (2016). Scheduling of job resources in multiunit projects with the use of time/cost criteria. *Archives of Civil Engineering*, *62*(1), 143-158. https://doi.org/10.1515/ace-2015-0057

Podolski, M. (2017). Management of resources in multiunit construction projects with the use of a tabu search algorithm. *Journal of Civil Engineering and Management*, *23*(2), 263-272. https://doi.org/10.3846/13923730.2015.1073616

Radziszewska-Zielina, E., & Sroka B. (2017a). Liniowy model optymalizacji czasowo-kosztowej planowania realizacji inwestycji wieloobiektowych. *Acta Scientiarum Polonorum. Architectura*, *16*(2), 3-12 (in Polish). https://doi.org/10.22630/ASPA.2017.16.2.01

Radziszewska-Zielina, E., & Sroka, B. (2017b). Priority scheduling in the planning of multiple-structure construction projects. *Archives of Civil Engineering*, *63*(4), 21-33. https://doi.org/10.1515/ace-2017-0038

Radziszewska-Zielina, E., & Sroka, B. (2018). Planning repetitive construction projects considering technological constraints. *Open Engineering*, *8*(1), 500-505. https://doi.org/10.1515/eng-2018-0058

Reda, R. M. (1990). RPM: Repetitive project modelling. *Journal of Construction Engineering and Management*, *116*, 316-330. https://doi.org/10.1061/(ASCE)0733-9364(1990)116:2(316)

Rogalska, M., Bożejko, W., & Hejducki, Z. (2008). Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling. *Automation in Construction*, *18*(1), 24-31. https://doi.org/10.1016/j.autcon.2008.04.002