JOURNAL OF SCIENCE OF THE MILITARY ACADEMY OF LAND FORCES

Volume 47 Number 1 (175) 2015

DOI: 10.5604/17318157.1158558

SCHEDULING OF MULTIUNIT PROJECTS USING TABU SEARCH ALGORITHM

Michał PODOLSKI*

* *Wrocław University of Technology*
e-mail: michal.podolski@pwr.edu.pl

Received on 18 July 2014; accepted in revised in December 2014

Abstract:

The paper describes problems of discrete optimisation in scheduling of multiunit projects. A model of this kind of project with possibility of using many workgroups by the contractor has been presented. It leads to reduction of project duration. For solving NP-hard optimisation problem, a tabu search algorithm has been applied in the model. The example of model and application of the algorithm are also included in the paper.

Keywords:

construction works scheduling, optimization, job scheduling, tabu search algorithm

INTRODUCTION

Construction practice often involves such cases which include completion of a number of civil structures or their elements which are part of a given project. Such projects can include apartment blocks, residential developments or groups of civil structures, as well as motorways, roads and utilities. These sorts of projects are characterized in repeatability of the works to be performed on individual placements or structures. This characteristic feature requires the forming of specialized working groups (crews) which perform one and only one kind of specific task (a job). In such projects these crews are moved from one site to the other doing only the job assigned to them. These kinds of projects are known as repetitive projects, and in some cases – multiunit projects [6,16].

The study of the scheduling of these projects results in identification of a number of models which can facilitate their precise description. Detailed solutions of the issues are given in the paper. The Polish literary sources focus mostly on the models basing on the idea of time coupling methods (TCM) understood as the time between the completion of individual jobs [10,13], whereas the efforts of foreign researchers are directed at solutions of tangible problems encountered in real repeatable project cases. For projects such as high-rise buildings, apartment blocks, residential developments or other buildings, methods based on the idea of *Line of Balance* (LOB) dominate [11]. The models created for repetitive projects often involve solving various objective functions and constraints. The problems studied most often are optimization issues of project duration minimization, minimization of total slack time for crews and minimization of the project costs, while respecting the binding project completion deadline. The following tools are used to solve these problems: linear programming, dynamic programming [3], artificial neural networks (ANNs) [1], generic algorithms (GA) and hybrid algorithms [12,18]. The techniques applied in repetitive project differ quite significantly from the techniques for the nonrepetitive projects. For the latter, what is successfully used are mostly the network scheduling methods, such as the critical path method (CPM). Project risk assessment is a crucial issue for such projects as well [19].

1. DESCRIPTION OF THE STUDIED MULTIUNIT PROJECT

The characteristic feature of the multiunit project studied here is its deterministic perspective: technical conditions are known, as is the technology and organization. Precise bill of quantities is available. The resources (crews) are also available at any moment for the period when they are needed and the works are done with due efficiency and quality. Also, there are no vital distortions to the works. The sequential relations between individual works are expresses in a sequence (order) invariable for any site. Such relation is usually encountered in sites characterized by uncomplicated working method, e.g. detached house. In case of these kinds of projects, individual tasks will be performed sequentially: earthworks, foundation work, walls with ceilings, roofing ect. In the studied multiunit project, however, both partial overlapping of individual tasks and time stops between the tasks are allowed. This is determined by the reality of construction work itself, where for example, work on one task is permitted to begin before the prior task has been completed. Additionally, in this model, a possibility exists to include additional time periods indispensible for the crew moving between the sites, as dependent on the kind of the crew and the sequence in which projects are to be completed. This is a crucial factor for those projects where construction sites are scattered across many specially distributed areas [7].

Another issue in each project is proper selection of resources for its completion. In the studied model of a multiunit project, contractor crews are assumed to be the resources. Work at the project sites is made possible by more than one group of crews, what significantly speeds up the completion of individual jobs and, as a consequence, decrease of the duration of the whole project. In the project model defined in this manner optimal solution is sought, such which would include the criterion applied. The duration of the whole project is the most often encountered criterion and this particular criterion is to be applied in the case presented.

The multiunit project model described in such a way can be identified as a flow problem with machines in parallel which is investigated as part of scheduling theory. The scheduling theory problems are the subject of studies in many academic and research institutions word-wide, which results from the fact that they model the operation of real manufacturing and industrial production systems. For the vast majority of practical problems, it is basically impossible to construct effective problem-solving algorithms mostly due to NP-hard character of optimization problems which these algorithms would have to tackle. What is possible, is the creation of only such exact algorithms, for which the calculation time grows exponentially with the magnitude of the problem to be analyzed. Therefore even a manifold increase of computer capacity does not translate into a significant improvement of exact algorithm problem-solving speed directly. This results in the search for optimal solutions in the form of efficient metaheuristic approximation algorithms which yield suboptimal solutions. These include evolutionary algorithms, tabu search, simulated annealing, ant colony optimization algorithms, hybrid algorithms etc.

For the author, the goal during the study of the multiunit project scheduling issues is paying attention to discrete optimization which may be encountered in projects of this particular kind taking into account the existing limitations. The means and methods of solving these problems can provide knowledge on optimal scheduling of multiunit projects much needed by building contractors (construction companies).

2. MULTIUNIT PROJECT OPTIMISATION MODEL

If total project duration is the criterion analysed in a multiunit project, the optimization model can be presented in the following manner[16].

Parameters

- Consider a site, divided into n placements or a set of sites $Z = \{Z_1, Z_2, Z_3, ..., Z_j, ..., Z_n\}$.
- There are construction crews performing one job of one kind and they make up set $B = \{B_1, B_2, B_3, ..., B_k, ..., B_m\}$.
- In each construction crew $B_k \in B$, there is (are) $m_k \geq 1$ crew(s), which is (are) characterized by the same or different productivity or composition: $B_k = \{B_{k1}, B_{k2}, B_{k3}, ..., B_{ki}, ..., B_{k\,m_k}\}$.
- Each site $Z_j \in Z$ requires that m jobs have to be performed, which make up set $O_j = \{O_{j1}, O_{j2}, O_{j3}, ..., O_{jk}, ..., O_{jm}\}$.
- It is assumed that a job $O_{jk} \in O_j$ can be performed by the crew $B_{ki} \subset B_k$. Duration of the job O_{jk} performed by the crew equals $p_{jki} > 0$. The set of possible durations p_j of jobs from the set O_j is specified by vector $p_j = \{p_{j1}, p_{j2}, p_{j3}, ..., p_{jk}, ..., p_{jm}\}$, where $p_{jk} = \{p_{jk1}, p_{jk2}, p_{jk3}, ..., p_{jki}, ..., p_{j\,k\,m_k}\}$.
- It is assumed the possibility of breaks or interruptions between jobs and simultaneous work of many crews on sites. The duration of a break between

one job k and another job k+1 ($s^F_{jk} > 0$) or the duration of simultaneously performed one job and the next one ($s^F_{jk} < 0$) in a site for set of jobs O_j are given in vector: $s^F_j = \{s^F_{j1}, s^F_{j2}, s^F_{j3}, ..., s^F_{jk}, ..., s^F_{jm}\}$. Although values of dependencies s^F_{jk} can be arbitrary, they have to be expressed as integers. They correspond to F (finish-to-start) time dependencies, as used in the MS Project software.

- Additional times required to mobilize the crews from one site to another, as determined by the type of crew and the site order are defined by the matrix SSk = [sS gh]n x n , where g ∈ [1..n], h ∈ [1..n], k ∈ [1..m].

Restraints:

- Technological sequence of jobs $O_{j,k-1}$ $O_{j,k}$ $O_{j,k+1}$ .is assumed.
- It is assumed that at any given time every working group from set B_k performs only one task/job
- It is assumed that job $O_{jk} \in O_j$ is performed uninterruptedly by crew $B_{ki} \subset B_k$ for the duration $p_{jki} > 0$.

The site order π which is a set of disjoined permutations (hereinafter referred to as "permutations", for the sake of brevity) $\pi = (\pi_1, \pi_2, ..., \pi_k, ..., \pi_m) \in \Pi$, where Π is the set of all the probable permutations, is the decisive variable. The site order for the crews from the crew set B_k is defined by permutation $\pi_k = (\pi_{k1}, \pi_{k2}, ..., \pi_{ki}, ..., \pi_{k m_k})$, where $\pi_k = (\pi_{ki}(1), \pi_{ki}(2), ..., \pi_{ki}(l), ..., \pi_{ki}(n_{ki}))$ defines the site order, as assigned to a crew $B_{ki} \subset B_k$, $B_k \subset B$. It is the decisive variable that clearly defines the allocation of crews performing jobs on individual sites. Thus, using set of disjoined permutations π, the duration of individual works on sites is clearly identified. Having permutation π, the set of job durations p_j out of the set O_j is the following: $p_j = \{p_{j1}, p_{j2}, p_{j3}, ..., p_{jk}, ..., p_{jm}\}$, where p_{jk} is the duration of job O_{jk}.

The criterion (objective function) is makespan $C_{max,}$ describing the time when all the jobs in all the sites have been finished. In the model the optimization consists in developing such job schedule, which would minimize the value of objective function while being subjected to constraints specified above.

It is easiest to present the multiunit project discussed as digraph, whose form depends on the assumed set of disjoined permutations π:

$$G(\pi) = (N', E(\pi)) \quad (1).$$

where N' is a set of nodes $E(\pi)$ – set of edges. It has been assumed that $N' = N \cap \{(start), (stop)\}$, where $N = \{1, .., k, ..., m\} \times \{1, ..., j, ..., n\}$ is a set of nodes representing all the jobs for individual sites, whereas (*start*), (*stop*) are fictional nodes representing the initial and final operations. The load of vertex (k, j) equals $p_{k,\pi_{ki}(l)}$, and the weight of nodes (*start*), (*stop*) equal zero. Set structure of the of edges $E(\pi) = E^F \cap E^S(\pi)$ depends on the decisive variable π. Horizontal edges (aka. sequential edges which represent the site order and which are characterized by disjunctive character) from the set $E^S(\pi)$ are located between the vertices $\pi_{ki}(l\text{-}1)$ and $\pi_{ki}(l)$ for *and* = 1, ..., m_k, k = 1, .., m, l = 2, ..., n_{ki}. Technological edges from the set E^F are located between the vertices which signify the jobs k-1 and k for site $\pi_{ki}(j)$. The location of these vertices depends on the position in permutations π_{k-1} and π_k. The edges from the set $E^S(\pi)$ are loaded with values equal $s^S_{k,\pi_{ki}(l-1)\pi_{ki}(l)}$, whereas the edges from the set E^F – with values equal $s^F_{k,\pi_{ki}(j)}$. (Fig. 1). The completion time for each job can be estimated using the recursive formula below

$$C_{k,\pi_{ki}(l)} = \max\{C_{k,\pi_{ki}(l-1)} + s^S_{k,\pi_{ki}(l-1)\pi_{ki}(l)}, C_{k-1,\pi_{ki}(l)} + s^F_{k-1,\pi_{ki}(j)}\} + p_{k,\pi_{ki}(l)} \qquad \textbf{(2),}$$

where:

$$j = 1, ..., n, l = 1, ..., n_{ki}, and = 1, ..., m_k, k = 1, ..., m, \pi_{ki}(0) = 0, C_{k,0} = 0, C_{0,j} = 0.$$

Each job completion time can be fund in the time of order $O(nm)$, similarly as in case of flow – shop problems with parallel machines. For the C_{max} criterion (which is the minimum time for all the jobs to be performed in all the sites, makespan), a set of permutations $\pi^* \in \Pi$ is sought, for which:

$$C_{max}(\pi^*) = \min_{\pi \in \Pi} C_{max}(\pi), \text{ where } C_{max}(\pi) = \max_{j} C_{m,j} \qquad \textbf{(3).}$$

For the studied model, any graph $G(\pi)$ can be treated as critical path. Figure 1 presents one of possible solutions for m = 5 kinds of jobs and n = 6 sites.

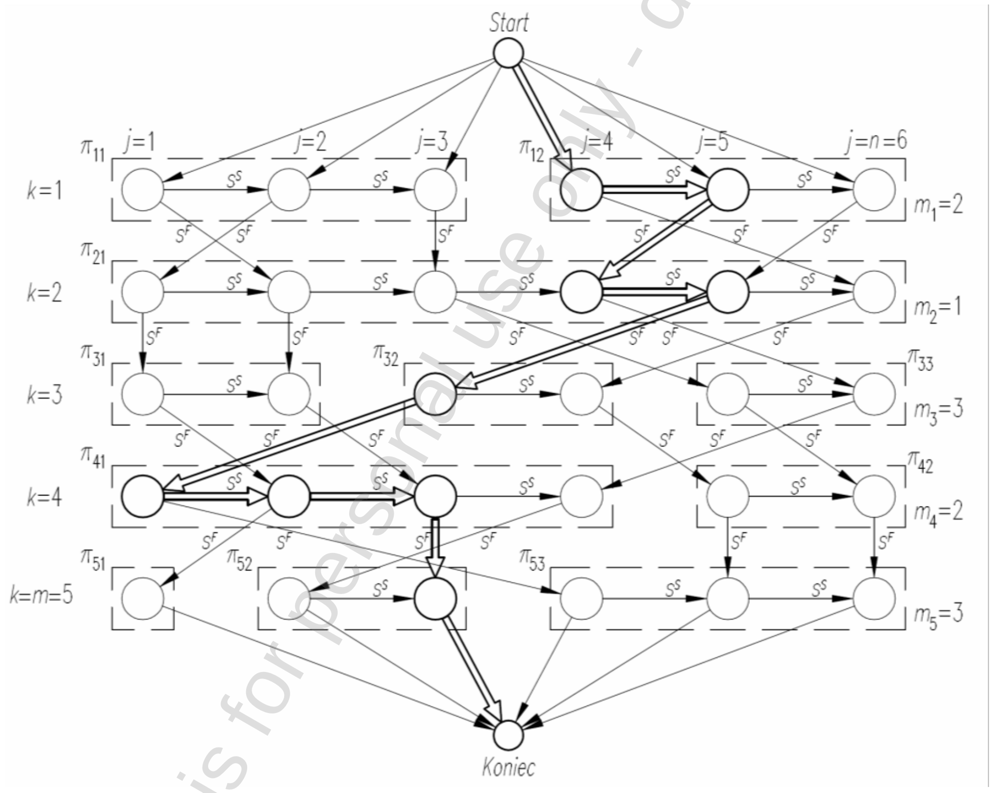

Fig. 1. Exemplary graph $G(\pi)$ for the studied multiunit project (m = 5, n = 6). Critical path highlighted.

Source: Own work

The model studied can be identified as one of scheduling theory problems – m-machines flow-shop scheduling problem with the following restraints: transportation time (added or subtracted) and setup times [20]. During the course of his studies on scheduling theory, the author has not encountered task with multiunit project defined in such manner. Practical application of this model is included in a calculation example in the following chapter.

3. CALCULATION EXAMPLE

As commissioned by the investor, the contractor is to carry out a Project in the form of n = 12 residential buildings spaced far away from one another. Each building requires m = 11 jobs to be performed in a proper sequence, as determined by the technology applied. The contractor has at their disposal only their own construction crews, each of which is specialized to perform one kind of job and works with the same constant productivity. It is assumed that the contractor has the following number of crews (characterised by the same productivity and no. of persons in each) (m = 11): k = 1 (earthworks) – 3 groups, k = 2 (foundations) – 4 groups, k = 3 (walls, floors) – 4 groups, k = 4 (rafter framing) – 4 groups, k = 5 (utilities) – 3 groups, k = 6 (woodwork) – 2 groups, k = 7 (plaster works, flooring, attic) – 4 groups, k = 8 (fencing, driveways) – 3 groups, k = 9 (tiling, painting) – 3 groups, k = 10 (flooring) – 3 groups, k = 11 (sanitary fixtures) – 2 groups. Basing on the labour intensity in individual sites, and the contractor's crew composition, as well as its productivity, job completion times in sites listed in Table 1 have been given. There are temporal relationships s^F_{jk} both between jobs and in the process/technological line. These relationships have been determined basing on the existing technological constraints. They correspond to FS (finish-start) relationships in the MS Project program and are presented in Table 2.

Table. 1. Job completion times by construction crews as expressed in working days

Task No. and kind k =	Sites j =											
	1	2	3	4	5	6	7	8	9	10	11	12
1.(earthworks)	3	3	3	3	3	3	2	3	2	3	3	3
2.(foundations)	12	10	9	11	12	15	14	15	14	14	13	10
3.(walls, floors)	22	25	23	26	23	27	27	19	23	25	22	25
4.(rafter framing)	14	15	16	17	16	13	16	11	12	16	18	15
5.(utilities)	10	8	13	10	11	9	9	12	13	9	12	8
6.(woodwork)	3	3	4	4	2	3	4	3	4	3	2	2
7.(plaster works, flooring, attic)	20	22	19	18	21	25	19	26	26	20	25	23
8.(fencing, driveways)	15	15	18	19	18	11	16	15	10	20	20	11
9.(tiling, painting)	8	10	8	7	8	7	10	7	7	8	7	8
10.(flooring)	8	11	7	8	10	6	8	5	7	7	9	5

Task No. and kind k =	Sites j =											
	1	2	3	4	5	6	7	8	9	10	11	12
11.(sanitary fix-tures)	5	5	5	5	4	4	3	6	5	5	7	6

Source: Own elaboration

The data concerning the time needed for the crew to be mobilised sites (as dependent on the kind of construction crew and site order) are recorded in the form of 11 matrices S^S_k (k = 1...11): $S^S_1 = S^S_2 = S^S_3 = S^S_7 = S^S_8 = [s^S{}_{gh}]_{n \times n} = 1$, $S^S_4 = S^S_5 = S^S_6 = S^S_9 = S^S_{10} = S^S_{11} = [s^S{}_{gh}]_{n \times n} = 0$.

Considering the above structure and the characteristics of resources (crews) used, the contractor seeks optimal scheduling, such which takes into account makespan criterion C_{max}.

Table. 2. Temporal relationships $s^F{}_{jk}$ between jobs k and k+1 [working days]

Jobs k =	Sites j =											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	5	5	5	5	5	5	5	5	5	5	5	5
3	0	0	0	0	0	0	0	0	0	0	0	0
4	-4	-5	-5	-5	-5	-4	-5	-3	-4	-5	-5	-5
5	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
7	0	0	0	0	0	0	0	0	0	0	0	0
8	-15	-15	-18	-19	-18	-11	-16	-15	-10	-20	-20	-11
9	10	10	10	10	10	10	10	10	10	10	10	10
10	-2	-3	-2	-2	-3	-2	-2	-1	-2	-2	-2	-1

Source: Own elaboration

4. SOLVING THE PROBLEM OF OPTIMISING A CALCULATION EXAMPLE USING A TABO SEARCH ALGORITHM

It is due to a high NP-hardness of the optimization problem in the given example [2] that a suboptimal solution will be sought using one of metaheuristics − a tabu search (TS) algorithm. The example given is quite small in size, thus one can assume that the solution found will approximate the optimal solution. Because of the high quality/accuracy of the solutions developed, tabu search algorithm is nowadays successfully applied in finding optimization issues in construction project planning [16].

Tabu search was originally proposed by F. Glover in his two part series [4, 5] devoted to the tabu search methodology. This method mimics the natural problem-solving process, as performed by human beings. The basic version of TS algorithm is initiated with a one potential solution to a problem. Then, a search space is found for this solution and the search space is defined as a set of feasible solutions which can be considered created following the moves in the given solution, that is transformations changing the given solution in accordance with certain rules. In this search space, a solution is sought such which would have the lowest objective function. This solution is the basis for the next iteration. The algorithm yields the best solution out of the search trajectory.

The essence of TS is the use of search history which precludes the trajectory from being stuck in the local optima and which enables its introduction into the more promising search areas. Short time memory (also called a tabu list) is most often used for this purpose. For a certain, short period of time this list stores the attributes of the most recently visited solutions, moves leading to these solutions or attributes of the moves of the recently visited solutions. It is usually done via introduction of the static tabu list, which results in removal of the oldest element at the moment when the most recent one is added to the list. What results from the attributes of this list, is that some future moves may not bode well and therefore will be treated as forbidden (tabu). This tabu can be annulled if the so-called aspiration-level function classifies the given move as beneficial. This is an additional function whose value for a given solution may cause either the moves leading to the solution or the attributes of the solution not be put onto the tabu list. The condition for ending the operation of the algorithm can be the following: time limit, maximal iteration number, reaching the satisfactory objective function or optimal value. Below you can find general outline of the TS method as used in flow shop problem-solving in the scheduling theory.

Let $\pi \in \Pi$ be any permutation, *LT* tabu list, *c* established objective function, and π^* the best solution found so far (at the beginning permutation π is accepted as π^*).

Step 1. Set neighbourhood N_π of permutation π such which would not contain the elements forbidden by tabu list *LT*;

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

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

Step 3. If $c(\delta) < c(\pi^*)$ then $\pi^* \leftarrow \delta$

Enter the attributes δ on the list *LT*;

$\pi \leftarrow \delta$;

Step 4. If a Stopping Condition is true then STOP, else go to Step 1.

Tabu search method has varying degrees of freedom: freedom of the kinds of moves and defining of the neighbourhood, of the form in which the tabu mechanism is operating (e.g. tabu list length, types of attributes) and search strategy. Currently, TS algorithms are among the most efficient tools used in theory of scheduling. The form of algorithm used has been selected as in case of a flow shop problem with machines in parallel basing on the papers published [14, 15]. In particular, it was studied in the paper [16]. The key control parameters of TS algorithm employed were the following: tabu list length equal 1/3 *mn*; no additional limitations of the magnitude of the neighbourhood set N_{π} were being imposed and maximal number of iterations allowed was 2000. Software implementation of the TS algorithm for the studied model has been conducted in *Mathematica* environment.

In order to find an optimal solution, the calculations using TS algorithm have been conducted three times. Project makespan from example No. 1 (best C_{max} value) is 179 working days, and it was achieved in the 2638 iteration of the work of the algorithm.

Legend for Figure 2:

ES	t_{ij}	EF
	j/i	
LS	TF	LF

i – kind of job

j – site numer

t_{ij} – job duration on site *j*,

ES – earliest start for a job,

EF – earliest finish for a job,

LS – latest finish for a job,

LF – latest finish for a job,

TF – total float

The best project duration time was obtained for the following disjoined permutation:

π={{{1,4,6,3,2,10,8,9},{12,11},{5,7}},
{{5,3,11},{12,6,2},{4,9,7,10},{1,8}},
{{12,11,6},{5,2,7},{4,9,10},{1,3,8}},
{{4,2,10},{1,9},{12,3,8,7},{5,11,6}},
{{12,9,7},{1,4,11,6},{5,3,2,8,10}},
{{1,5,3,7,8,10},{12,4,11,2,9,6}},
{{12,2,8},{4,3,6},{5,11,10},{1,9,7}},
{{5,12,3,6,7},{4,11,8},{1,2,9,10}},
{{12,11,6,8},{1,5,2,10},{4,3,9,7}},
{{1,12,11,2,8},{9,7},{4,3,5,6,10}},
{{1,12,9,4,10,7},{11,2,5,6,3,8}}}.

Figure 2 presents the schedule from project given as an example in the form of directed graph $G(\pi)$ and for the suboptimal solution found.

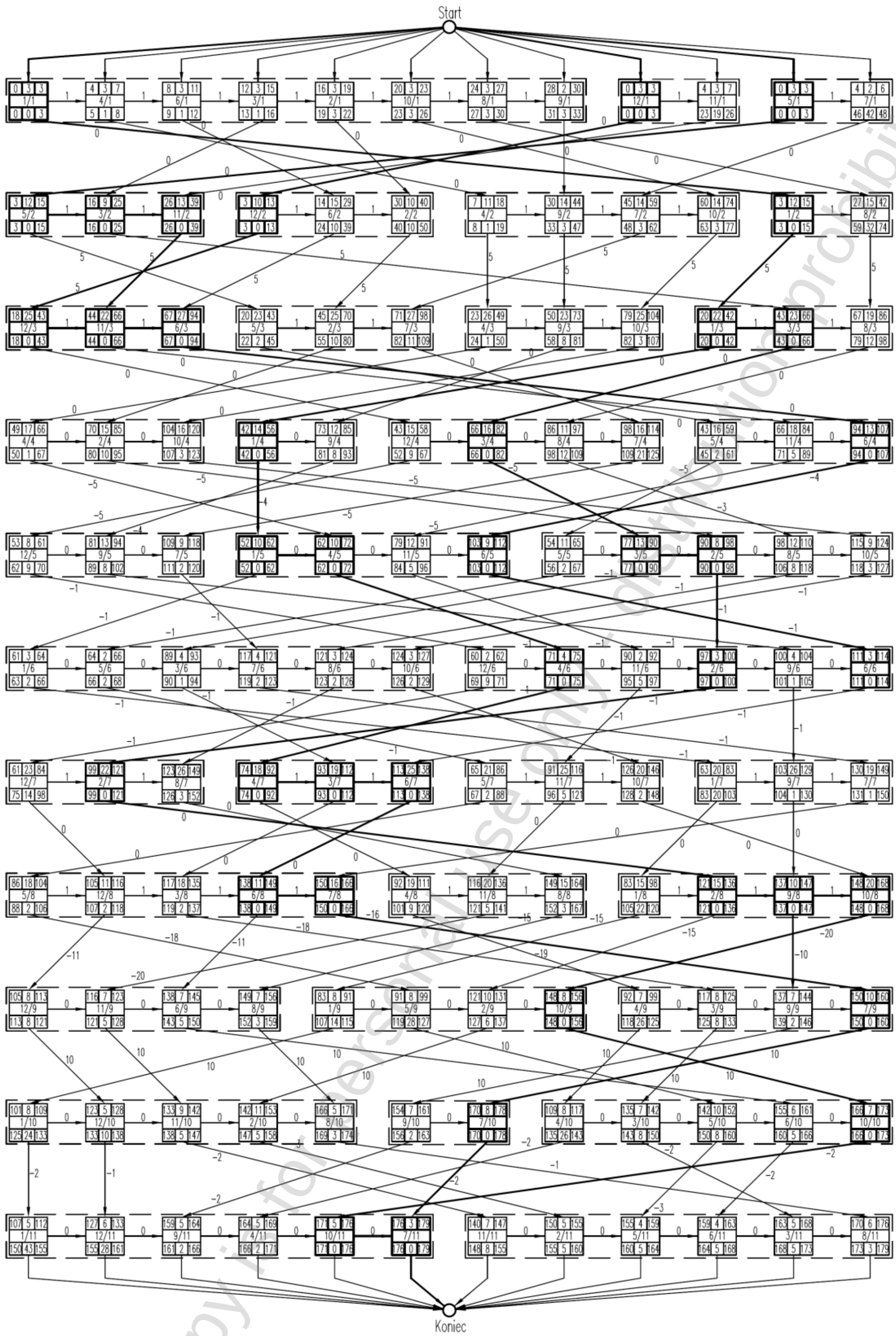

Fig. 2. Project schedule from example 1 in the form of a directed graph $G(\pi)$ for the solution found. Critical path is highlighted (m = 11 kinds of jobs, n = 12 sites)

Source: Own elaboration

CONCLUSION

Scheduling theory provides its users with strong tools which can be used to solve optimization issues to be encountered in construction projects. One of those tools is tabu algorithm which has been presented in the model multiunit project described in the paper. The model defined is characterized by the presence of a strong NP-hard discrete optimization problem which cannot be solved using the available exact algorithms. One of the papers cited [16] sought the optimal solution using other approximation algorithms (genetic and simulated annealing ones) for the example studied. In both cases, however, the results obtained had been worse than the results produced by TS algorithm by ca. 6%. Further studies by the author will be directed towards discrete optimization problems, the continuation of the above-analysed issues with reference to objective functions other than the function of project completion time/makespan of the project.

REFERENCES

1. Adeli H., Karim A., *Scheduling/cost optimisation and neural dynamic model for construction*, [in:] "Journal of Construction Engineering and Management", 1997, 123(4), 450-458.
2. Błażewicz J., Złożoność obliczeniowa problemów kombinatorycznych, WNT, Warszawa, 1988.
3. El-Rayes K., Moselhi O., *Optimising resource utilization for repetitive construction projects*, [in:] "Journal of Construction Engineering and Management", 2001,127(1), pp. 1-18.
4. Glover F., *Tabu Search, Part I*, [in:] "ORSA Journal of Computing", 1989, No. 1, pp. I90-206.
5. Glover F., *Tabu Search, Part II*, [in:] "ORSA Journal of Computing", 1990, No. 2, pp. 4-32.
6. Harris R.B., Ioannou P.G., *Scheduling projects with repeating activities*, [in:] "Journal of Construction Engineering and Management", 1998, 124(4), pp. 269-278.
7. Hegazy T., Elhakeem A., Elbeltagi E., *Distributed Scheduling Model for Infrastructure Networks*, [in:] "Journal of Construction Engineering and Management", 2004, 130(2), pp. 160-167.
8. Hegazy T., Wassef N., *Cost optimisation in projects with repetitive nonserial activities*, [in:] "Journal of Construction Engineering and Management", 2001, 127(3), pp. 183-191.
9. Hejducki Z., Podolski M., *Harmonogramowanie przedsięwzięć budowlanych z zastosowaniem algorytmów metaheurystycznych*, [w:] „Zeszyty Naukowe Wyższej Szkoły Oficerskiej Wojsk Lądowych", No. 4(166), Wrocław 2012, pp. 68-79.
10. Hejducki Z., Rogalska M., *Time couplings methods,* Oficyna Wydawnicza PWr., Wrocław, 2011.

11. Lumdsen P., *The line of balance method*, Pergamon Press, London, 1968.

12. Moselhi O., Hassanein A., *Optimised scheduling of linear projects*, [in:] "Journal of Construction Engineering and Management", 2003, 129(6), pp. 667-673.

13. Mrozowicz J., *Metody organizacji procesów budowlanych uwzględniające sprzężenia czasowe*. DWE, Wrocław, 1997.

14. Nowicki E., *Metoda tabu w problemach szeregowania zadań produkcyjnych*, Prace Naukowe Instytutu Cybernetyki Technicznej PWr., Wyd. Politechniki Wrocławskiej, Wrocław 1999.

15. Nowicki E., Smutnicki C., *The flow shop with parallel machines: A tabu search approach*, [in:] "European Journal of Operational Research", 1998, No. 106, pp. 226-253.

16. Podolski M., *Analiza nowych zastosowań teorii szeregowania zadań w organizacji robót budowlanych*. Praca doktorska, Raporty Inst. Bud. PWroc. 2008, Ser. PRE No. 5/08, http://www.dbc.wroc.pl/Content/2515/Podolski_Analiza_PhD.pdf .

17. Podolski M., *Zastosowanie metaheurystycznych algorytmów szeregowania zadań w harmonogramowaniu wieloobiektowych przedsięwzięć budowlanych. Technologia and zarządzanie w budownictwie*, [in:] "Prace Naukowe Instytutu Budownictwa Politechniki Wrocławskiej", Seria: Studia and Materiały No. 18, pp. 147-156.

18. Rogalska M., Bożejko W., Hejducki Z., *Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling,* [in:] "Automation in Construction", No. 18, 24-31, 2008.

19. Skorupka D., *Modelling of risk in the building projects*, [in:] "Operations Research and Decision", (2006), pp. 133-143.

20. Smutnicki C., *Algorytmy szeregowania*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2002.

BIOGRAPHICAL NOTE

Michał PODOLSKI, Ph. D, Eng. – adjunct researcher at Wroclaw University of Technology's Faculty of Civil Engineering. He received his Master of Science and Engineer title at Wroclaw University of Technology's Faculty of Civil Engineering and his doctorate in technical sciences in 2008. His scientific interests include: job scheduling issues in construction projects, which includes optimisation of project scheduling, application of artificial intelligence (A.I.) and application of job sequencing theory in the construction sector.

HOW TO CITE THIS PAPER

Podolski M., (2015). Scheduling of multiunit projects using tabu search algorithm. Zeszyty Naukowe Wyższa Szkoła Oficerska Wojsk Lądowych im. gen. Tadeusza Kościuszki Journal of Science of the gen. Tadeusz Kosciuszko Military Academy of Land Forces, 47 (1), pp.110-122. http://dx.doi.org/10.5604/17318157.1158558