



Taylor & Francis  
Taylor & Francis Group



ENGINEERING STRUCTURES AND TECHNOLOGIES

ISSN 2029-882X/eISSN 2029-8838

2016 8(2): 71–78

doi:10.3846/2029882X.2016.1196916

## AN EXACT SOLUTION FOR THE RESOURCE-CONSTRAINED CONSTRUCTION SCHEDULING PROBLEM

<sup>1</sup>Bartłomiej SROKA, <sup>2</sup>Elżbieta RADZISZEWSKA-ZIELINA

*Department of Construction Technology and Organization, Institute of Management in Construction and Transport, Cracow University of Technology, Warszawska street 24, 31-155 Cracow, Poland*

Received 14 December 2015; accepted 16 April 2016

**Abstract.** Reduced time and, by the same token, the cost of the project is a crucial factor in contemporary construction. This article presents a method for the exact optimisation of a resource-constrained scheduling problem. Based on the Critical Path Method, graph theory and linear programming, an algorithm was developed and the FROPT program was written in Matlab to minimise the execution time of the task. By using the newly-created program, sample networks were calculated and the results were compared with results obtained by using the MS Project scheduling program (using approximation algorithm). The execution time obtained by using FROPT were on average 10% shorter than those obtained using MS Project. In selected cases the improvement in execution time reached 25%. A deterministic approach to the problem may shorten planned project times and bring financial benefits. Due to the exponential complexity of the algorithm, it is most useful in solving small or highly coherent networks. The algorithm and program may result in benefits not offered by commercial software for planners of building projects.

**Keywords:** RCPSP, CPM, linear programming, MS Project, graph theory, project scheduling.

### Introduction

There is no doubt that in today's world, time is extremely valuable. This is nowhere more evident than in the construction sector, where every second can be calculated into an equivalent cash amount. Of course, a golden mean must be found between reduction of execution time and the costs which that reduction entails. However, if construction works can be accelerated solely by means of correct organisation of work, then it is always advisable to do so. The efficient and reasonable management of available tools, specialist and machinery reduces the total execution time, and thus the cost of the project. Work becomes more efficient and the investor does not need to bear large financial burdens due to down time caused by poor organisation. It is very important from the perspective of construction practice to optimize the duration of a

construction project with limited resources because of possible cost penalties connected with delays. An experienced engineer is able to make decisions about which activities should be carried out first, and which can be postponed. It should be noted though, that each construction project is unique, and no one is able to analyse in his or her head all the possible scenarios and choose the best from among them. Even with the use of commercially available specialist software tools, it is not always possible to obtain the optimal schedule (Kaklauskas, Zavadskas 2015; Kapliński 2008).

The Resource-Constrained Project Scheduling Problem (RCPSP) network model is an extension of the Critical Path Method (CPM). In addition to execution time, every activity in a network is assigned in advance the resources which are required for its completion. Finding the minimal time for the execution of

Corresponding author:

E. Radziszewska-Zielina E-mail: [eradzisz@izwbit.pk.edu.pl](mailto:eradzisz@izwbit.pk.edu.pl)

Copyright © 2016 Vilnius Gediminas Technical University (VGTU) Press

<http://www.tandfonline.com/TESN>

a project with constrained resources is a well-known and well-studied problem (Marcinkowski 2002). The golden age of research in this area was in the 1960s (Weaver 2006). In modern times, metaheuristic algorithms are most often used (due to the computational complexity of the problems) which include simulated annealing (Boctor 1996), artificial neural networks (Ömer, Ümit 2015), tabu search (Omer 2010), genetic algorithms (Sou-Sen, Tzung-Heng 2010), swarm algorithms (Tran *et al.* 2015), hybrid algorithms (Bettemir, Sonmez 2014; Wang, Fang 2012), and memetic algorithms (Safaei, Shams-Javi 2011). Many variations on the RCPSP theme have also arisen: energy saving scheduling (Miyamoto *et al.* 2014), scheduling with transportation delays (Quilliot, Toussaint 2012), and priority-rule methods (Andreas *et al.* 2013). There are many programs are useful in finding solutions to such problems (MS Project, Procore, Aconex), but no manufacturer identifies the methods or algorithms used in these programs. The majority of these programs also do not supply the optimal solution, at best giving an approximation of one. Therefore, it is important to examine this issue and consider methods to obtain the optimal solution.

The aim of this article is to attempt to find the minimum time for the execution of a project with constrained resources. Based on the CPM, and using linear programming in the Matlab language, an algorithm was developed and the FROPT (FRONT OPTimization) program was written. FROPT is non-commercial software, and was developed for research purposes at the Cracow University of Technology; it may be shared among those interested in the construction planning industry. Example solutions for networks generated in FROPT are presented here. In order to verify these results, they were compared with those obtained by using MS Project.

## 1. Description of the problem

The algorithm described below is an adaptation and extension of the mathematical method of solving the problem of scheduling with constrained resources as described in academic works (Ignasiak 2001, 1975). A network with constrained resources is a generalisation of the CPM. In this case however, each activity has two parameters:  $d_i$  – duration and  $z_i$  – demand of a given activity for a resource. Resource is taken to mean a renewable good which serves to execute an activity, for

example an employee or piece of machinery. There are also certain constraints on the amounts of resources. Constraint is represented by  $z_{\max}$ . A front of activities  $F_j$  in a network is a set of activities in which any two activities of this set may not belong to the same path. By execution of the project, we understand a column vector of non-negative real numbers  $T_j$  clearly assigned to the fronts  $F_j$ . The number  $T_j$  should be interpreted as the duration of consecutive fronts.

In regard to optimisation of project execution time, one must strive to fulfil the condition given in formula 1:

$$\sum_j T_j \rightarrow \min. \quad (1)$$

Among the fronts, a pair of contrary fronts should be identified. The fronts  $F_i = \{Ck, Cl\}$  and  $F_j = \{Cm, Cn\}$  are contrary if the activity  $Ck$  precedes them on the activity path  $Cm$ , while the activity  $Cn$  precedes the activity  $Cl$ . Such a pair may not appear in execution. Transformation of the network into a highly coherent graph eliminates the problem of contrary fronts. Such a graph may be constructed by adding dummy activities with zero duration and demands on resources.

Each dummy activity may possess two lines. In order to find the optimal solution, further calculations should be performed on all the representatives of the highly coherent (acyclic) graphs.

The relation between the fronts and activities are presented mainly in the form of a matrix of incidence  $A_{ij}$ . This is a binary matrix. If at the axis of the  $i$  and  $j$  lines of the column there is a value of 1, then the  $i$  activity belongs to the  $j$  front.

The meaning of the demand of the front for resources should now be defined. This will be the sum of demands for resources of the activities which comprise the front. The calculation method is presented in formula 2:

$$z(F_j) = \sum_i z_i, \text{ for } i, \text{ so that: } C_i \in F_j. \quad (2)$$

Fronts whose values  $z(F_j)$  exceed the permissible resources  $z_{\max}$  may not be taken into consideration.

The durations of specific activities are calculated as the sum of the duration of the fronts which include the activity. This is presented in formula 3:

$$D(C_j) = \sum_j T_j, \text{ for } j, \text{ so that: } C_i \in F_j. \quad (3)$$

The problem discussed may be reduced to the issue of linear programming shown in formula 4:

$$\text{find } T_j, \text{ so that: } \begin{cases} \sum_j T_j \rightarrow \min, \\ \Lambda_i \sum_j A_{ij} T_j = d_i, \\ \Lambda_j \sum_i z_i A_{ij} \leq z_{\max}. \end{cases} \quad (4)$$

where:  $A_{ij}$  – is the matrix of incidence of fronts with activities,  $T_j$  – is the duration of these,  $d_i$  – is the duration of particular activities,  $z_i$  – is the demand of particular activities for resources,  $z_{\max}$  – is the resource constraint.

### 2. Algorithm

The FROPT algorithm finds the shortest possible execution time for a network with constrained resources. The program was entirely implemented in the Matlab R2014a with an additional module for optimisation (Optimization Toolbox). For the program to work correctly, the following assumptions must be met (these are also requirements of the CPM):

- A single initial and final peak (event). The initial event must have only outgoing activities, while the final event must have only incoming activities. All other events have gaps (activities) both incoming and outgoing.
- Nodes must be numbered by successive positive integers in the order that they are completed.
- There may not be cycles in the graph.
- Durations of all activities and their demands for resources must be known.
- Available resources must be at least equal to the maximum demand of individual activities. There is no pre-established constraint, however cases in which available resources exceed the maximum demand of the front do not add anything new.
- If at least two activities lead directly from peak  $i$  to  $j$ , they should be replaced with a graph in which peak  $k$  is introduced. One activity  $i$ - $j$  remains unchanged, while the other is replaced by activities  $i$ - $j$  and  $k$ - $j$ . One of the activities is dummy (zero duration and zero demand for resources), and the other has the parameters of the activity which was replaced.

The complete FROPT algorithm is presented in Fig. 1. After entering starting data the numbering of the nodes and activities should be changed. For a graph like this, a family of highly coherent graphs is generated. Next, check whether the representatives of the family contain a cycle. For graphs which do not

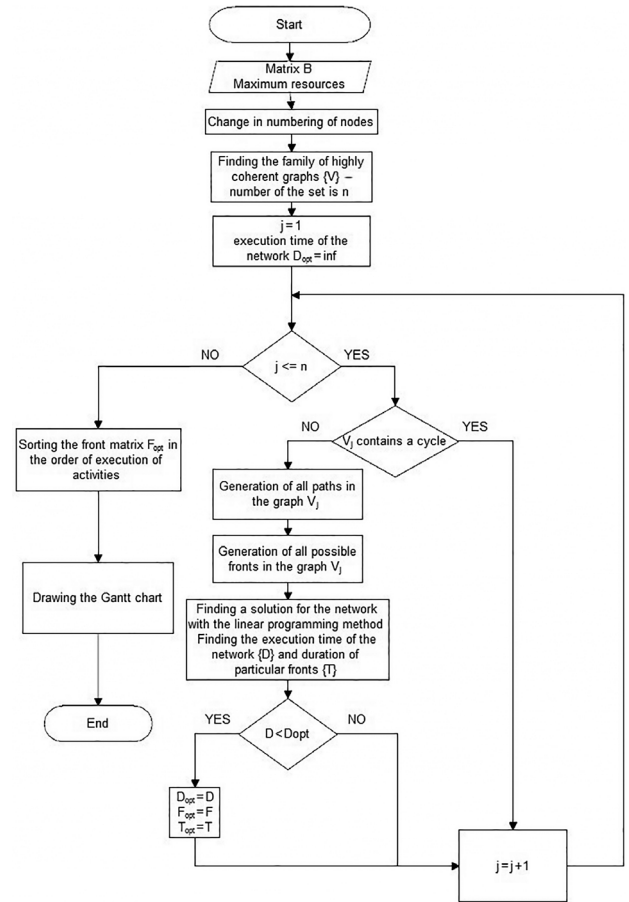


Fig. 1. The FROPT algorithm

meet this condition, a matrix of incidence is generated and analysis is carried out to find the optimal solution using linear programming. For the optimal solution, the fronts are sorted based on the structure of the network. The final stage is the generation of a Gantt chart. This is a deterministic algorithm, so it always finds an optimal solution. The algorithm possesses exponential computational complexity  $O(2^n)$ .

The algorithm for finding fronts in the form of a matrix of incidence with activities for a given network goes as follows (let  $N$  mean the number of activities in a given front):

1. For  $N = 1$  each of the activities may be a front. The matrix  $F$  and supplementary unit matrices  $D$  and  $C$  should be placed. The variable “ $i$ ” means the number of fronts in the matrix  $D$ , and the variable “ $j$ ” in matrix  $C$ .
2.  $N = N + 1$
3. For each number from 1 to “ $i$ ” and from 1 to “ $j$ ”, check whether  $D_i$  and  $C_j$  may be a front (all activities belonging to front  $D_i$  and  $C_j$  do not lay on the same path and meet the demands

for available resources). If such a front can exist (and does not belong to matrix  $F$ ), it should be saved in the supplementary variable  $E$  and added to matrix  $F$ . In this way  $N$ -activity fronts are found

4. If  $N$ -activity fronts are not found, the algorithm finishes its operation, and the result is the matrix of incidence  $F$ . Otherwise, matrix  $C$  should be overwritten with variable  $E$ , variable  $E$  should be zeroed, and proceed to step 2

This algorithm finds the same fronts repeatedly, therefore it is necessary to check whether a given front has not already been added to the matrix  $F$  (if so, this front should not be taken into consideration a second time).

FROPT is a program without a graphic user interface. Data is fed from the level of the Matlab environment console or in the form of an m-file. The correct operation of the program requires defining the 4-column matrix describing the network. Subsequent columns designate: the number of the initial activity peak, the number of the final peak, activity duration, the demand of an activity for resources. Next, the loop function must be used, which accepts as arguments the created matrix and the maximum available demands (see an example of using the function in Fig. 3). The program shows first of all the minimum realization time and the organization of task performance in the form of the Gantt chart as well as additional information (that is less practically important), such as the matrix of realization times for all representatives of highly coherent graphs. FROPT requires basic programming knowledge; however, there is a possibility to extend it by a graphic interface in regard to data feeding or to adapt it as an extension to the existing scheduling programs. The authors of the present paper have access to the source code and documentation of the FROPT program. The program is being further developed.

### 3. Solution for an example network

#### 3.1. Example 1

The network shown in Figure 2 is assumed. The network possesses 11 nodes and 16 activities. The available resources have been set with a value of 10. For a problem formulated in this way, an execution time of 18 units was obtained. To transform this into a highly coherent graph, as many as 21 dummy activities must be added between the peaks. The number of network

to be calculated is  $2^{21} = 2\,097\,152$ , in order to find the optimal solution. Attention should also be paid to the assignment of the resources at the end of the project. There may be some disproportion of resource usage. Without interfering in the structure of the network, we can switch the execution date of the last two fronts. This will cause available resources to be released at the end of the project. We can also carry out activity 15 earlier (together with activity 16). This will mean that the usage of resources over time will be more balanced. This network was also calculated using Microsoft Project.. The execution time was 22, which is considerably worse than the time obtained by FROPT. The difference between these results is 18%. Figure 3 presents the code of the program, the schedule obtained with the use of FROPT and the schedule obtained with the use of MSP for data from network 1.

A comparison was also made between the execution times for different values of available resources. The results are presented in Figure 4 in the form of a chart and tables. The smallest number of resources was 9 (the highest demand for resources of a single activity), and the largest number was 29 (the highest demand of a front). The chart shows a maximum of 16 resources, as the results obtained for a larger number were the same as those obtained by using the CPM. It can be seen that in the majority of cases, the execution time can be reduced by from 7% to 18% in comparison with those returned by MS Project. In one case only were the results obtained by both programs the same.

The achievement by the FROPT program of results which are 18% better than those of MS Project is very satisfying in terms of the execution time of a project. Even when the execution time of a construction project is only shortened by 7% or 9%, there are considerable savings in terms of lowering costs (for 11 resources, a time of 15.5 was obtained).

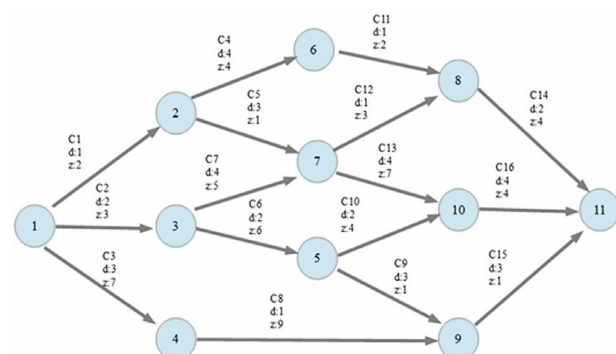


Fig. 2. Network 1

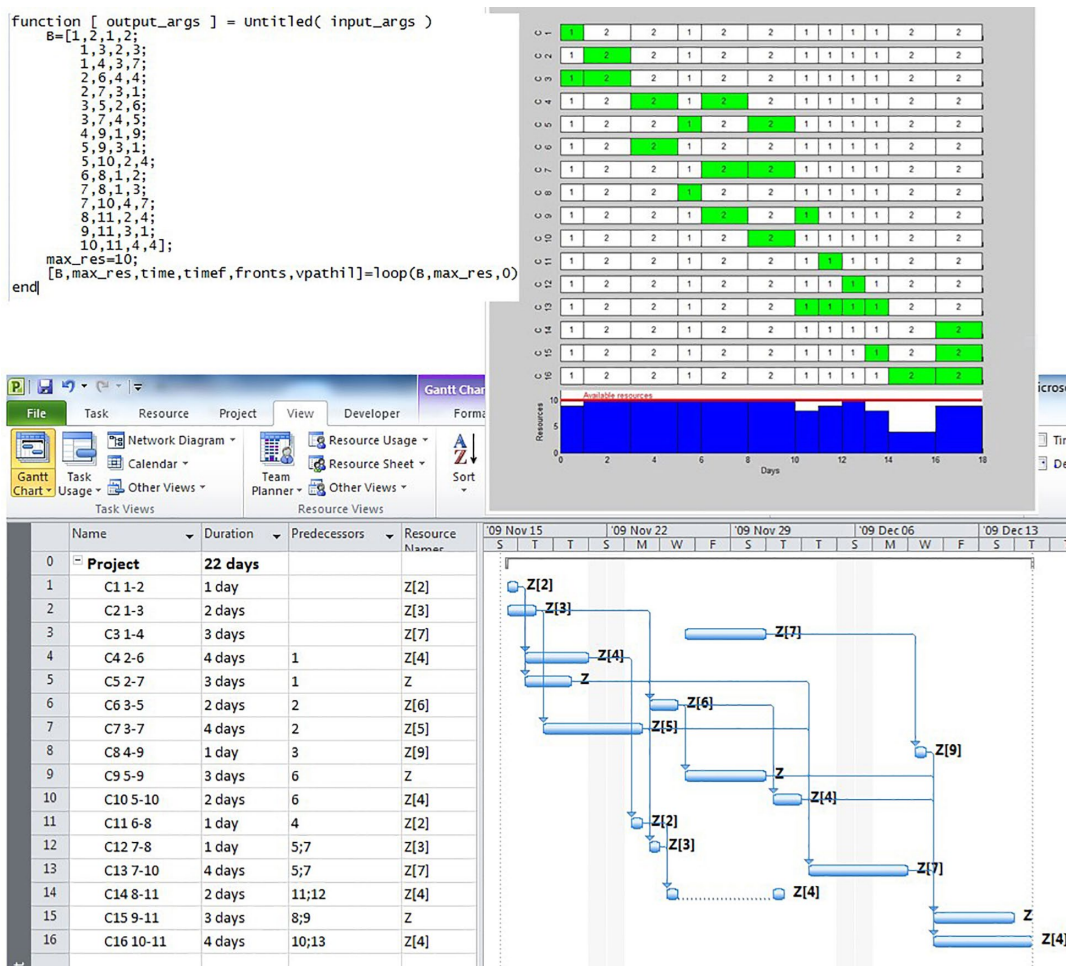


Fig. 3. A solution for Network 1 using FROPT and MSP

Available resources	9	10	11	12	13	14	15	16
Duration using FROPT	20	18	15.5	15	14	14	14	14
Duration using MSP	22	22	17	15	15	15	15	14
Difference	9%	18%	9%	0%	7%	7%	7%	0%

Chart showing the duration dependent on available resources

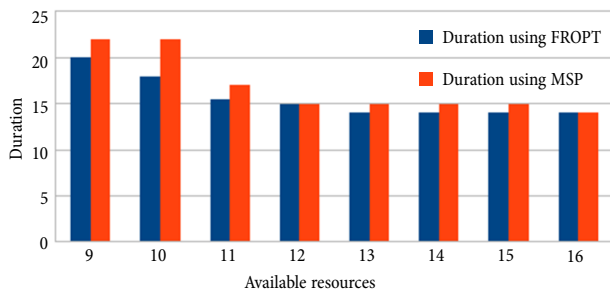


Fig. 4. A comparison of the results obtained using MSP and FROPT for various amounts of available resources for network 1

An analysis was also carried out of the optimal execution times of the network for particular representatives of highly coherent graphs (for the resource constraint value of 10). As was previously mentioned,

in order to find the minimum time, more than two million networks were calculated. It appears that only 1026 of them did not contain cycles and could be processed further. This is a mere 0.05% of all the possible representatives of highly coherent graphs. This means that the greatest effort is required by the task of finding graphs without cycles.

### 3.2. Example 2

An interesting feature of the problem is when non-integers are returned for a network in which the durations of the activities are all integers. It appears that for small networks, such a result may be obtained. Figure 5 shows just such a network. There are three parallel activities leading from peak 1 to peak 2. Each of them requires 1 resources and has a duration of 1 time unit. However, in accordance with the assumptions described above, two additional dummy activities should be added (with zero duration and zero demand for resources). For available resources with a value of 2, the solution is not obvious.

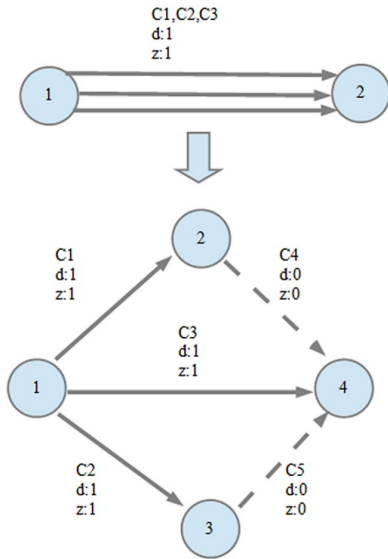


Fig. 5. Network 2

The solution for the network is quite surprising, as for all of the integer durations and demands for resources, the result is a calculable non-integer (Fig. 6). This results from the fact that the structure of the network and the assumptions allow the execution of an activity to be paused and later resumed when resources become available. The network presented is the smallest one in which such a property was observed.

An analysis of the above network was also carried out using MS Project. The Microsoft program returned an execution time for the schedule shown above of 2 units (Fig. 6). This means that the FROPT result is 25% better.

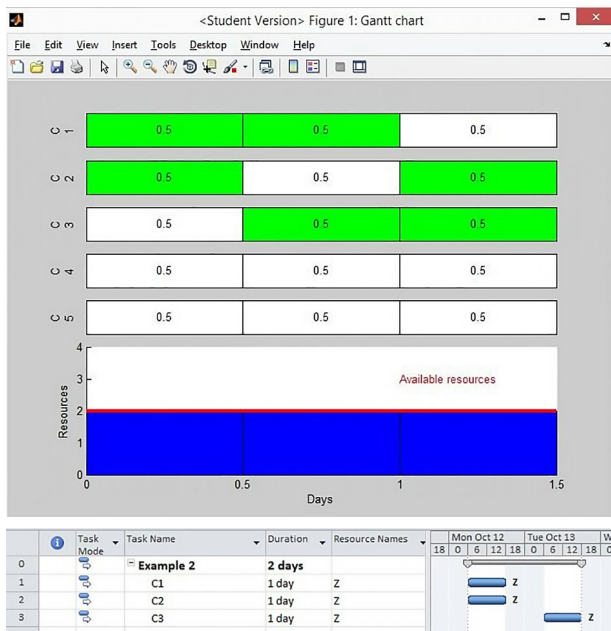


Fig. 6. A solution for Network 2 using FROPT and MSP

### 3.3. Example 3

Figure 7 in MSP presents in the form of a table the scope of work, duration times and the required number of workers to make a reinforced concrete foundation with a foundation wall using prefabricated concrete blocks during the realisation of two identical buildings. Tasks on plots are performed in the technological order while tasks between the plots are independent. A given number of people is assigned for each task. The contractor has a team consisting of maximum 9 workers. Figure 7 also shows the schedule obtained in MSP. Figure 8 presents the schedule obtained with the use of FROPT (dummy activities have been removed from the diagram). The realisation time with the use of MSP is 19 workdays while with the use of FROPT 17.5 days. The result obtained in FROPT is therefore by almost 9% better than in MSP. This proves the usefulness of the method and the program for scheduling projects with constrained resources.

### Final conclusions

The presented problem is important in construction. The application of the FROPT program can reduce the duration of a project by means of correct organisation of work. Completing a project in a shorter time allows the building contractor to avoid penalties.

The program we developed serves to find an exact solution to resource-constrained project scheduling problems (RCPSP) by optimising execution time. FROPT is non-commercial software, and was developed for research purposes at the Cracow University of Technology; it may be shared among those interested in the construction planning industry.

Based on our research, the following conclusions can be made:

- The algorithm used in the FROPT program is effective. It finds optimal solutions in finite time. It is a deterministic algorithm, however it is characterised by exponential complexity, which restricts its potential application to small networks or ones of a specific structure.
- Due to the computational complexity of the problem, a deterministic approach is not effective for large networks. For some networks, the calculation time could amount to several years, which is of course unacceptable. A solution for this problem may be the application of distributed computing or the application of heuristic algorithms (for example genetic algorithms or the Monte Carlo method).

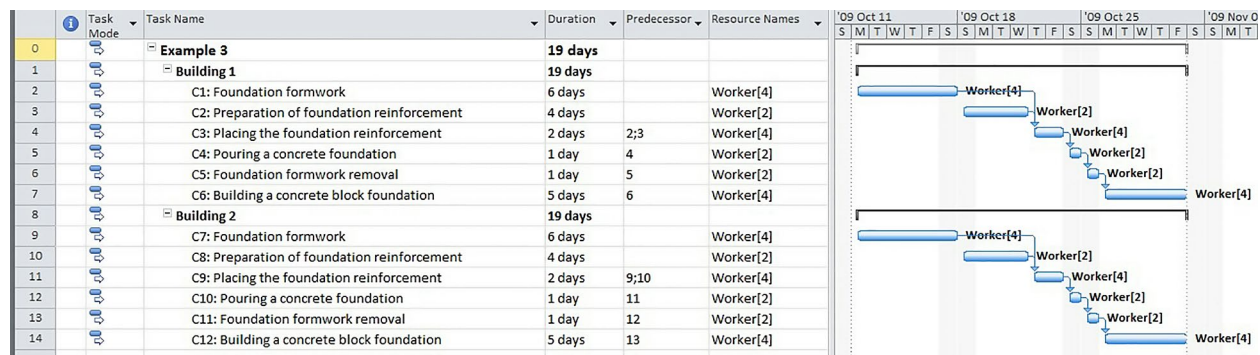


Fig. 7. A solution for example 3 using MSP

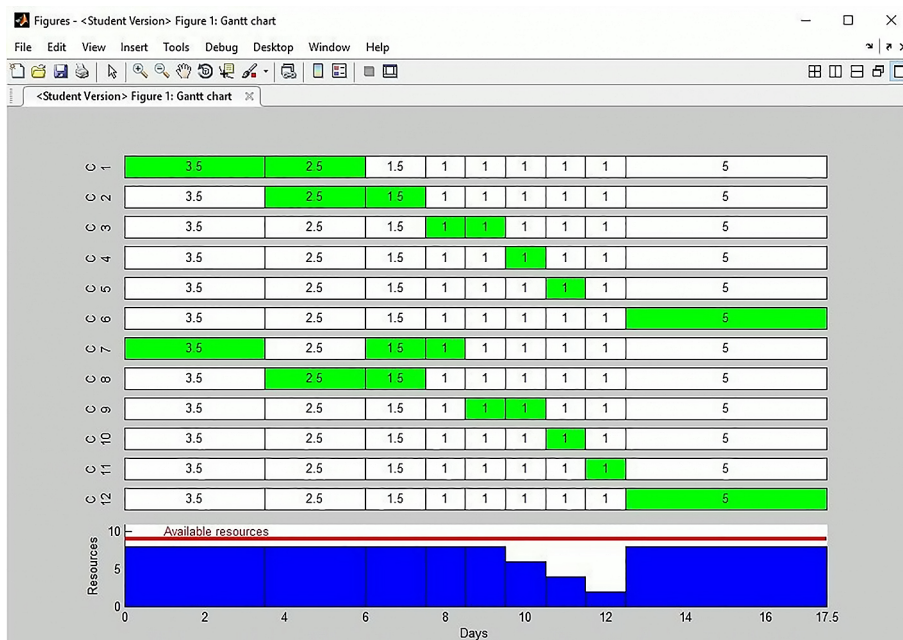


Fig. 8. A solution for example 3 using FROPT

- For large networks with a coherent structure (where few dummy activities are needed to transform the network into a highly coherent graph) the execution time of the program will be acceptable to the user. Removal of activities from such a graph (those activities which can be removed due to conditions which must be fulfilled by the CPM network) paradoxically results in an increase in the number of representatives of highly coherent graphs which must be calculated, increasing the execution time of the program.
- The use of the FROPT can provide as much as 25% reductions in execution time in comparison to the results obtained using MS Project. This was the best result obtained for a given network (Example 2). For a fairly regular network (Example 1), reductions of as much as 18% were obtained. On average, approximately 7–10% improvements were noted. The time-saving depended on the structure of the network, its

size, and on available resources. A shorter execution time can directly influence savings. Optimisation is carried out strictly in terms of organisation of work; other factors are not considered. It can be assumed then, that using the same available resources, the time required for works can be shortened, and thus made cheaper.

- The FROPT program does not calculate optimal resource distribution. The amount of resources used during the works should be, as far as possible, at a consistent level. If the distribution of resources is unsatisfactory, corrections may be entered into the schedule manually, taking into account time reserves afforded by activities which lie outside of the critical chain.
- For a sufficiently large amount of available resources, both programs return the same execution times (equivalent to solutions offered by the Critical Path

Method). It should be noted though, that in order to obtain results equal to those delivered by CPM, it is sufficient to use a mere fraction of the available resources. This means that shorter execution times can be achieved with far fewer resources than had been anticipated.

- As Example 2 showed, for networks with durations of activities expressed in integers, it was possible to obtain non-integer execution times. This is caused by the distribution of resources and the possibility to interrupt certain activities. This concerns networks with fronts which include at least three activities.
- As shown in example 3, the presented method and the FROPT program are useful for scheduling real projects with constrained resources.

Summing up, the following areas for further research can be defined. To develop exact methods for scheduling with constrained resources, the problem of the generation of acyclic networks should be considered, which will significantly improve the performance time of programs. A further development of metaheuristic methods and the introduction of more efficient deterministic algorithms may improve our understanding of the problem. In the case of the development of quantum computers, deterministic methods should be developed, as computational complexity will cease to be relevant and exact solutions will be achievable within acceptable times.

## References

- Andreas, S.; Thibaut, F.; Peter, J. S.; Mark, G. W. 2013. Solving RCPSp/max by lazy clause generation, *Journal of Scheduling* 16(3): 273–289. <http://dx.doi.org/10.1007/s10951-012-0285-x>
- Bettemir, Ö.; Sonmez, R. 2014. Hybrid genetic algorithm with simulated annealing for resource-constrained project scheduling, *Journal of Management in Engineering* 31(5).
- Boctor, F. F. 1996. Resource-constrained project scheduling by simulated annealing, *International Journal of Production Research* 34(8): 2335–2351. <http://dx.doi.org/10.1080/00207549608905028>
- Ignasiak, E. 2001. *Badania operacyjne*. Warszawa: Polskie Wydawnictwo Ekonomiczne (in Polish).
- Ignasiak, E. 1975. *Programowanie sieciowe*. Warszawa: Państwowe Wydawnictwo Ekonomiczne (in Polish).
- Kaklauskas, A.; Zavadskas, E. K. (Eds.). 2015. *Multiple criteria Analysis of the life cycle of the built environment*. Vilnius: Technika. 448 pp.
- Kapliński, O. 2008. Planning instruments in construction management, *Technological and Economic Development of Economy* XIV(4): 449–451. <http://dx.doi.org/10.3846/1392-8619.2008.14.449-451>
- Marcinkowski, R. 2002. *Metody rozdziału zasobów realizatora w działalności inżynieryjno-budowlanej*. Warszawa: Wojskowa Akademia Techniczna. (in Polish)
- Miyamoto, T.; Mori, K.; Kitamura, S.; Izui, Y. 2014. A study of resource constraint project scheduling problem for energy saving, in *International Conference on System Science and Engineering ICSSE*, 11–13 July 2014, Shanghai, China, 23–26. <http://dx.doi.org/10.1109/icsse.2014.6887897>
- Ömer, Ö.; Ümit, G. 2015. A neural network for resource constrained project scheduling programming, *Journal of Civil Engineering and Management* 21(2): 193–200. <http://dx.doi.org/10.3846/13923730.2013.802723>
- Omer, A. 2010. Tabu search and an exact algorithm for the solutions of resource-constrained project scheduling problems, *International Journal of Computational Intelligence Systems* 4(2): 255–267.
- Quilliot, A.; Toussaint, H. 2012. Resource constrained project scheduling with transportation delays, *Information Control Problems in Manufacturing* 14(1): 1481–1486. <http://dx.doi.org/10.3182/20120523-3-ro-2023.00061>
- Safaei, A.; Shams-Javi, M. 2011. A hybrid method of memetic algorithm and artificial immune system for solving RCPSp, *International Review on Modelling and Simulations* 4(5): 2644–2650.
- Sou-Sen, L.; Tzung-Heng, H. 2010. A genetic algorithm-based optimal resource-constrained scheduling simulation model, *Construction Management and Economics* 20(2): 131–141.
- Tran, D.; Cheng, M.; Cao, M. 2015. Solving resource-constrained project scheduling problems using hybrid artificial bee colony with differential evolution, *Journal of Computing in Civil Engineering* 30(4). [http://dx.doi.org/10.1061/\(asce\)cp.1943-5487.0000544](http://dx.doi.org/10.1061/(asce)cp.1943-5487.0000544)
- Wang, L.; Fang, C. 2012. A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem, *Expert Systems with Applications* 39(3): 2451–2460.
- Weaver, P. 2006. A brief history of scheduling – back to the future, *myPrimavera Conference*, 4–6 April 2006, Canberra, Australia.

**Elżbieta RADZISZEWSKA-ZIELINA** is the Head of the Department of Construction Technology and Organization at Tadeusz Kościuszko Cracow University of Technology. She has over 100 publications in the field of materials and technologies applied in construction industry, the management of a construction enterprise (especially in relation to partnering relations in construction, systems supporting the decision making in construction, marketing in a construction enterprise, construction market research); public procurement, organisation of construction work tenders, construction enterprise management, construction project planning and organisation.

**Bartłomiej SROKA** is a lecturer in the Department of Construction Technology and Organization at Tadeusz Kościuszko Cracow University of Technology. His scientific interests include decision theory, network programming and the planning of multi-object investments.