# Quantitative time management methods in project management

*Renata Walczak*

Warsaw University of Technology, Faculty of Civil Engineering, Mechanics and Petrochemistry
Department of Mechanical Systems Engineering and Automation
Address: Al. Jachowicza 2/4, 09–402 Płock, Polska, e-mail: rpwalcza@pw.plock.pl

Summary. The paper presents an algorithm for the calculations of a network of relationships using the critical path method, the correlations for the total, free, conditional and independent slack time, the correlations for all kinds of relationships between the tasks: FS, SS, FF, SF. An example of a network with modifications has been presented.

Key words: project management, quantitative methods, critical path method.

## INTRODUCTION

Quantitative methods of project management date back to the beginning of the XX century. Already before WWII a Pole, Karol Adamiecki, created the foundations of the quantitative time management in projects and rhythmical production [4, 8, 13, 16]. H. Gantt, F. Taylor, Ch. Babbage, the team of the professor P. Blacket, also contributed to the development of the quantitative method. Another step was the creation of the critical path method by Du Pont engineers and the simultaneous production of the Polaris ballistic missile along with the development of the PERT method [15, 18]. These methods could not advance due to a lack of calculation tools. Ever since personal computers advanced sufficiently the quantitative methods should have been sidetracked, yet this is not the case. Despite availability of sophisticated software for project management the software is not properly used. There is no knowledge on the mathematical quantitative methods used in the software [20]. On many occasions huge projects involving over 10 thousand tasks are planned manually and the project management software is merely there for printing purposes. The plans are prepared due to the requirements of public procurement or insurance companies but later they are sent directly to the archives and the project is managed by intuition. The aim of this paper is to acquaint the reader with the basics needed to employ the quantitative methods of managing time in project management and prepare the reader to use the software based on these methods.

## WORK BREAKDOWN STRUCTURE

When organizing projects it is necessary to determine the aim of the task, the basic project checkpoints (milestones), the tasks that will be realized, duration of these tasks, used resources, expenses etc. After determining of the project aim a multilevel work breakdown structure is created (WBS) formed through a decomposition of the subsequent elements aiming at putting all the tasks together that will be realized during the project. In order o identify the individual tasks the structure is encoded [1, 2, 3, 5, 6, 7].

The work breakdown structure constitutes the basis for work planning, estimation of resources and expenses, determination of the scope of works, determination of the development path and the inspections, assignment of responsibility, risk identification and introduction of changes. The benefit resulting from the application of WBS is easy retrieval of information from previous projects with similar tasks, processes, assembly and components, the possibility of determining of the purchasing needs and specifications, the possibility of using joint integrated management system - joint databases, documentation, quotations, budgets, technical drawings, financial accounting systems as well as the possibility of performing of the cost analysis and other data.

The work breakdown structure is used on many managerial levels. Depending on the needs of the recipient it is possible to analyze information on different levels of the structure taking into account their different detail level. The highest level managers need consolidated information available at higher WBS levels while the lower level employees are interested in details characteristic of the lower levels.

The responsibility for the preparation of the work breakdown structure is on the side of the project manager. The WBS may be created through a top-down method (method dictated by the higher levels of authority), when the main scaffolding of the structure is created by the project manager and higher level managers and the detailed work is defined by the line managers. The bottom-up method consists in preparing of the structure from detail to general. This method is usually applied when the structure is created based on the already existing projects, the results to be achieved are known and the structure is composed of known elements. The structure is developed by employees of lower level and people responsible for the work realization and the adjustments and acceptance is done by the project manager. It is worth involving important interested parties such as clients in the WBS creation [17, 19].

An example of work breakdown structure has been presented below.

*Level 1 Level 2 Level 3 Level 4*

***0.*** *Building a house*
- ***0.1.*** *Foundation*
  - ***0.1.1.*** *Digging the foundation*
  - ***0.1.2.*** *Shuttering*
  - ***0.1.3.*** *Pouring concrete*
- ***0.2.*** *Construction of a building*
  - ***0.2.1.*** *Construction of a basement ceiling*
  - ***0.2.2.*** *Building walls*
    - ***0.2.2.1.*** *Building partition walls*
    - ***0.2.2.2.*** *Building load bearing walls*
  - ***0.2.3.*** *Construction of a tie beam*
- ***0.3.*** *Construction of a roof*
  - ***0.3.1.*** *Fixing the ceiling*
  - ***0.3.2.*** *Construction of the roof elements*
  - ***0.3.3.*** *Roofing*
- ***0.4.*** *Installations*
  - ***0.4.1.*** *Water and sewage*
  - ***0.4.2.*** *Central heating*
- ***0.5.*** *Project management*
  - ***0.5.1.*** *Project organization*
  - ***0.5.2.*** *Planning*
  - ***0.5.3.*** *Project supervision*
  - ***0.5.4.*** *Documentation*

## NETWORK METHODS OF PROJECT MODELLING

After the task has been identified a network model of a project is created. The tasks are set one after another in a technological order of performance. The basis of the network is digraphs i.e. ordered sets of vertices (nodes) and arches (edges) with one source and one exit:

$$\mathrm{G}=\{\langle \mathrm{W(G)},\mathrm{L(G)}\rangle\} \tag{1}$$

The network is identified as a set of ordered threes:

$$S = [W, L, \varphi] \tag{2}$$

$$\varphi\colon W{\times}W \rightarrow L \tag{3}$$

where:
$G$ digraph,
$W$ set of graph vertices,
$L$ set of graph branches,
$S$ network,
$\varphi$ relation.

In networks the arches are assigned weights. An ordered pair on the graph is presented with arrows connecting individual nodes oriented from the beginning of the arch to its end. The length of the graph arches does not represent the actual lengths between the edges but only represents the order of their occurrence. In the graphs showing the realization of the projects paths are determined i.e. such sets of vertices in which they occur only once and in which the arches are different. The longest path in the network is referred to as sensitive path, critical path or decisive sequence of tasks. The projects can be presented in the form of graphs where the vertices are assigned events and the arches are assigned tasks (tasks at the edges of the graph) or where the vertices of the graph are assigned tasks and the edges represent their technological order of performance (tasks in the graph nodes). The arches in these graphs are assigned weights denoting, for example, the duration of the task. The events (tasks) are assigned a date. In order to unequivocally determine the graph we only need to enter the set of ordered pairs and the values of the nodes or arches. The neighborhood matrices are determined too (for the edges or the vertices) or the incidence matrices (of the neighborhood of edges and arches). Additionally for identification of the network the weight matrix has to be specified. In this paper only simple network without multiple edges will be considered. One pair of nodes will always be assigned an arch. Cyclic networks (with loops) i.e. edges connecting the same vertices will not be considered [11, 12].

## CRITICAL PATH METHOD

Critical Path Method (CPM) has been discussed in many publications, yet these publications present only the simplified form of calculation. Calculation methods are not presented for other relations than the finish of the predecessor task-beginning of the successor task. Not all the types of slack times (only total and free) are discussed. The principles of Project modeling through the Critical Path Method are thus worth revisiting.

The Critical Path Method is a fully deterministic method. It assumes both the deterministic value of duration of individual tasks and the deterministic construction of the network of relations.

The CPM was developed going on the assumption that there are no limited resources. Such networks can obviously undergo optimizations in terms of resources availability upon construction of the network taking into account the logical sequence of tasks only.

The network calculations are performed in two stages. In the first stage (forward calculations) the earliest dates of the occurrence of all the events are set. In the second stage the latest dates of the occurrence are set [2, 5, 9, 10, 19]. Below the algorithm of the network calculation has been presented.

***Network calculation algorithm***

**Stage I**

1. Setting the start date of the project
2. Setting the earliest start dates of all tasks
3. Setting the earliest finish dates of all tasks
4. Setting the finish date of the project

**Stage II**

1. Setting the latest finish date of the project (most frequently this date is set as equal to the earliest finish date of the project)
2. Setting the latest finish dates of all tasks
3. Setting the latest start dates of all tasks
4. Setting the slack time
5. Setting the critical path

In the calculations the following symbols have been used:

| | |
|---|---|
| *WS* | early start, |
| *WK* | early finish, |
| *PS* | late start, |
| *PK* | late finish, |
| *ZC* | total slack, |
| *ZS* | free slack |
| *ZW* | conditional slack, |
| *ZN* | independent slack, |
| *t* | duration of task, |
| *p, poprz* | predecessor task |
| *n, nast.* | successor task |

Upon performing of the calculations for the assumed durations the optimization of the network is carried out taking into account the availability of resources and other adopted criteria.

In the first stage the start date of the project is set. In non-computer calculations the start date of the project is zero. Computer software assumes the start date of the project as a date given by the operator [2, 5, 9, 10, 19].

Another step in the calculation of the first stage is setting the earliest dates of all events. For each task the earliest start date needs to be set. Then the earliest finish date *WK* of this task is set, which is equal to the sum of the earliest start date *WS* and the duration of the task *t*. If, in the analyzed event, several tasks are finished we need to calculate the finish date of each of them and as the date of occurrence of the event we need to assume the greatest number because the successor task may start when the longest of the predecessor task is finished:

$$WK=max\ (WS + t). \tag{4}$$

Upon the completion of the calculations for all the tasks we obtain information on the earliest finish date of the project.

The second stage of the calculations begins with setting the latest admissible finish date of the project [2, 5, 9, 10, 19]. Most frequently, the earliest finish date of the project is assumed (the same as in the first stage of the calculations). This could also be a different date – a date set by a directive. The next step is setting the latest dates for the occurrence of all events. For each task first we need to set its latest finish date *PK*. Then the latest start dates *PS* of the tasks are set. The latest start date of a task *PS* equals the difference between the latest finish date of the task and the duration of the task *t*. If in an event several tasks are started we calculate the latest start dates of these tasks and as the latest date of the event occurrence we assume the least number. This ensures completion of the project in the assumed time:

$$PS = min\ (PK - t). \tag{5}$$

Upon completing the calculations for the tasks we can determine different types of slack times. The total slack time ZC is a minimum determined based on the calculations for the start and finish dates of the tasks. The total slack time can be determined as a difference between the latest finish date of a task and the earliest finish date of the same task. The total slack time can also be determined as a difference between the latest start date of a task and the earliest start date of the same task. The total slack time value is assumed as the least value of the two calculations:

$$ZC = min\ \{(PK - WK), (PS - PS)\}. \tag{6}$$

The total slack time is common for all tasks in one sequence. If it is used by the earlier tasks the later tasks will lose this slack time. It comprises all other slack times. If it is zero then the other slack times are also zero. The task under analysis is performed as early as possible. The subsequent task is performed as late as possible. The total slack time is also a difference between the latest start of the successor task (if there are several successor tasks we should choose the least value) and the earliest finish of the actual task.

Free slack time ZS is a difference between the earliest start date of the successor task (if subsequently there are several successor tasks we should choose the least value) and the earliest finish date of the actual task. The predecessor task is performed as early as possible. The actual task is performed as early as possible. The successor task is performed as early as possible:

$$ZS = min\ (WS_{nast}) - WK. \tag{7}$$

This slack time can be used and the successor tasks will not lose their slack time, yet the predecessor task must be performed in the earliest possible time.

Conditional slack time ZW is a difference between the minimum value of the latest start dates of the successor tasks and the sum of the maximum value of the latest finish dates of all predecessor tasks and the duration of the task for which the conditional slack time is calculated. The predecessor task is performed as late as possible. The actual task is performed right after the predecessor. The successor task is performed as late as possible:

$$ZW = min\ (PS_{nast}) – \{max\ (PK_{poprz}) + t\}. \qquad (8)$$

The slack time allows performing the predecessor tasks at any time. The use of this slack time forces the performance of the successor task at the latest time i.e. all slack times of the successor tasks in this sequence will be used. The conditional slack time comprises independent slack time.

Independent slack ZN is a difference between the minimum value of the earliest start dates of all successor tasks and the sum of the maximum value of the latest finish dates of all predecessor tasks and the duration of the task for which the conditional slack time is calculated. The predecessor task is performed as late as possible. The actual task is performed right after the predecessor. The successor task is performed as early as possible:

$$ZN = min\ (WS_{nast}\ ) – \{max\ (\ PK_{poprz}\ ) + t\}.$$

This slack time provides information in what time the task can be performed without influencing the performance of the successor and predecessor tasks. They can be performed in times determined in the calculations and the independent slack time will remain intact. This slack time can assume negative values, which means that it is impossible to perform the actual task as late as possible and the successor task as early as possible. The negative value indicates how many units of time the successor task needs to shift to be after the actual task.

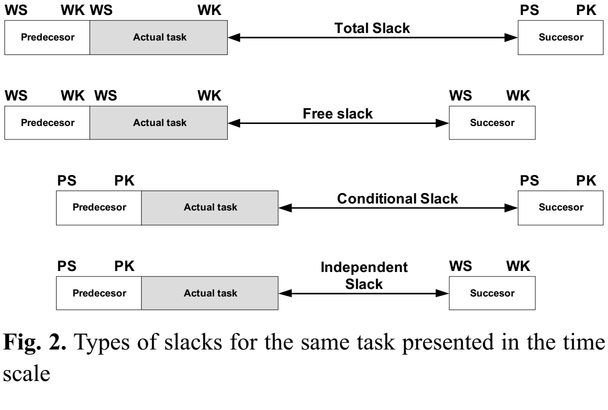

**Fig. 2.** Types of slacks for the same task presented in the time scale

The longest sequence of tasks in duration in the whole project is the critical path. This is at the same time a sequence that does not have slack times or has the least slack times of all sequences of tasks. The tasks on the critical path are subject to particular supervision on the side of the project manager. If any of the tasks on this path extends in time then the whole project will extend in time by the same amount of time. If the tasks on this path are shrunk in time the whole project may be finished earlier (but not always) by the time the critical path was shrunk. Upon shrinking of the critical path other paths may appear that may be longer than that being shrunk. In such a case the critical path becomes the longest path in the whole project. In a project there may be several critical paths. Then, completion of a project is dependent on the tasks on each of these paths.

In a project the tasks can be correlated in different ways. The basic and default relationship is the finish-start (FS) relationship. With this relationship when the predecessor task is finished we can start the successor task. The start-start (SS) relationship allows placing the start of the successor task after the start of the predecessor task. In the finish-finish relationship after finishing of the predecessor task finishing of the successor task takes place. The most controversial is the start-finish (SF) relationship. Here, after the start of the technologically predecessor task the finish of the technologically successor task is placed. Obviously, the first in time will be the task that finishes first, yet this finish depends on the start of the task that is placed as the successor in time. An example could be the change of the shifts of the security personnel. The employee can only end his shift if the next employee comes in to replace him.

Below complete relations have been presented for the determination of dates and slack times in the network.

Each successor task can be shifted against the predecessor by a relative or absolute period of time. Superimposing of rigid time schedules of the task realization is also possible.

### DEPENDENCES OF THE CALCULATIONS FOR DIFFERENT RELATIONS IN THE NETWORK

- Stage I of the calculations

**Calculating the earliest start for the successor task**

For the relationship FS: $WS_n = WK_p + \Delta.$ (9)

For the relationship SS: $WS_n = WS_p + \Delta.$ (10)

For the relationship FF: $WS_n = (WK_p + \Delta) - t_n.$ (11)

For the relationship SF: $WS_n = (WS_p + \Delta) - t_n.$ (12)

Of all the $WS_n$ value select the greatest value

**Calculating the early finish of the successor task**

$$WK_n = WS_n + t_n. \qquad (13)$$

- II stage of the calculations

**Calculating the latest finish of the predecessor task**

For the relationship FS: $PK_p = PS - \Delta.$ (14)

For the relationship SS: $PK_p = (PS_n - \Delta) + t_p.$ (15)

For the relationship FF: $PK_p = PK_n - \Delta.$ (16)

For the relationship SF: $PK_p = PK_n - \Delta + t_p.$ (17)

Of all the $PK_p$ values select the least value

**Calculating the latest start of the predecessor task**

$$PS_p = PK_p - t_p. \qquad (18)$$

**Total slack time**

$$ZC = \min \{ (PS_p - WS_p), (PK_p - WK_p) \}. \qquad (19)$$

**Free slack time**

For the relationship FS: $ZS = (WSn - WKp) - \Delta$. (20)

For the relationship SS: $ZS = (WSn - WSp) - \Delta$. (21)

For the relationship FF: $ZS = (WKn - WKp) - \Delta$. (22)

For the relationship SF: $ZS = (WKn - WSp) - D$. (23)

Of all the calculated free slack times select the least value

Example of the network of relations calculated with critical path method has been presented in figure 1.

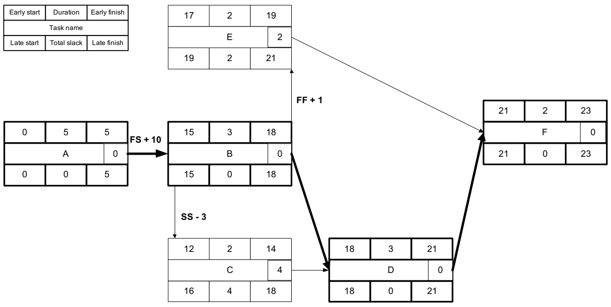

**Fig. 1.** Example of the network of relations drawn with the method activities on nodes

The calculations have been performed with the critical path method.

## CONCLUSIONS

There are many methods of quantitative modeling of projects. Aside from the critical path method such methods were developed as the PERT method that takes into account the probabilistic characteristics of the task duration, the fully probabilistic GERT method, the method of critical chain taking into account only time and delay management or the earned value method whose purpose is to manage the realization of the project. A variety of methods optimizing the project have also been developed.

This paper aimed at consolidating the knowledge related to the basics in the network calculations with the use of the critical path method. It is the computer tools for project management that most frequently use this method. The exploration and full comprehension of the course of the calculations will enable a successful understanding of computer software for project management.

## REFERENCES

1. A guide to The Project Management Body of Knowledge- PMBOK Guide. Third Edion. 2000. MT&DC.
2. **Badiru A. B. 1996:** Project management in manufacturing and technology operations. John Wiley & Sons Inc. New York.
3. **Cleland D.I. 1999:** Project management. Strategic design and implementation. McGraw-Hill, New York.
4. **Dibnis G. 2010:** The problems of administration for industrial enterprices. TEKA Kom. Mot. i Energ. Roln. – OL PAN, v. 10D, p. 101-106.
5. **Kerzner H. 2000:** Applied project management. Best practices on implementation. John Wiley & Sons Inc. New York.
6. **Kimmons R.L., Loweree J.H. 1989:** Project management. A reference for professionals. Marcel Dekker Inc., New York.
7. **Knutson J. 2001:** Project management for business professionals: a comprehensive guide. John Wiley and Sons, New York.
8. **Korobetsky Y., Sokolova Y., Doroshko V. 2010:** Imitation abstract models of production systems. TEKA Kom. Mot. i Energ. Roln. – OL PAN, v. 10A, p. 285-293.
9. **Lawrence J.A.Jr., Pasternack B.A. 1998:** Applied management science. A computer-integrated approach for decision making. John Wiley & Sons Inc. New York.
10. **Lientz B.P., Rea K.P. 1998:** Project management for the 21-st century. Academic Press, San Diego.
11. **Mantel S.J.Jr., Meredith J.R., Shafer S.M., Sutton M.M. 2001:** Project management in practice. John Wiley & Sons Inc. New York.
12. **Nicholas J.M. 2001:** Project management for business and technology. Principles and practice. Prentice Hall, Upper Saddle River, New Jersey.
13. **Riazantseva N., Tyulyenyev H. 2010:** Efficiency of decision making in the state institutions. TEKA Kom. Mot. i Energ. Roln. – OL PAN, v. 10D, p. 230-235.
14. **Samuelson F.W., Marks S.G. 1998:** Ekonomia menedżerska. PWE, Warszawa.
15. **Shapovalova A., Shapovalova S. 2010:** TEKA Kom. Mot. i Energ. Roln. – OL PAN, v. 10D, p. 249-256.
16. **Sharipova O. 2010:** Management of diversified entrepreneurial structures activity harmonization. TEKA Kom. Mot. i Energ. Roln. – OL PAN, v. 10D, p. 257-262.
17. **Shtub A., Bard J.F., Globerson S. 1994:** Project management, engineering, technology, and implementation. Prentice Hall, Englewood Cliffs, New Jersey.
18. **Slobodyanyuk M., Nechaev G. 2010:** TEKA Kom. Mot. i Energ. Roln. – OL PAN, v. 10B, p. 162-170.
19. **Stuckenbruck L.C. 1981:** The implementation of project management: the professional's handbook. Project Management Institute. Addison-Wesley Longman.
20. **Zhytna I., Yefremenko O. 2010:** Competitiveness of an enterprise and methods of its estimation.. TEKA Kom. Mot. i Energ. Roln. – OL PAN, v. 10D, p. 325-330.

## METODY ILOŚCIOWE ORGANIZACJI CZASU W ZARZĄDZANIU PROJEKTAMI

Streszczenie. W pracy przedstawiono algorytm obliczeń sieci relacji używając metody ścieżki krytycznej, korelacji dla sumarycznego, wolnego, warunkowego i niezależnego czasu luźnego, korelacji dla wszystkich rodzajów relacji między zadaniami: FS, SS, FF, SF. Zaprezentowano przykład sieci z modyfikacjami.

Słowa kluczowe: zarządzanie projektami, metody ilościowe, metoda ścieżki krytycznej.