

The Algorithm Design by using programming language Visual Basic for Application on the example of Critical Path Method (CPM)

Slavomir Vukmirović
Economics Faculty
University of Rijeka, Croatia
vukmirovics@gmail.com,
Marko Čičin-Šain
Primorska Banka, Rijeka, Croatia
Marko.Cicin-Sain@gmail.com
Ivana Mežnarić
Economics Faculty
University of Rijeka, Croatia

Summary

In this paper, we describe usability of programming languages and tools in the development of complex algorithms on the example of critical path algorithm in project network optimisation. We discuss the integration of mathematical and computer algorithms using Visual Basic for Excel. We suggest a methodological frame for development and usability of algorithm, which integrates programming of matrix and recursion formulas, and is implemented in critical path programming.

Key words: Critical Path Method (CPM), algorithm, visualisation, modelling, programming, formula

1. Introduction

The basic paradigm of active process in information technology is programming. In other words, a computer is a universal machine, which can be transformed using an appropriate program into a tool to perform the most various tasks. A computer program consists of a set of steps intended to accomplish an imaginary algorithm (procedure). Each artefact of information and communication technology has been made throughout computer programming. Visual Basic for Excel is an object and visually-oriented programming language, which enables us to make assumptions for integrated modelling and programming, aimed at understanding, visualizing and integrating complex mathematical equations, and creating computer programs.

A mathematical algorithm is a set of mathematical operations that are used to find solution to a problem based on initial data. A computer algorithm is a set of programmed procedures that are used to find solution to a problem based on initial input. In algorithm approach, data and methods are integrated in a whole based on a logical sequence that determines a computing procedure [5]. This paper describes visualization and integration of

mathematical and computer algorithm on the example of critical path method in a project network diagram.

The paper is divided into seven interrelated parts. After the Introduction, in the Part II “Project network modeling by Critical Path Method (CPM)” we describe project network terms and characteristics. The Part III describes “Project network modeling in the spreadsheet interface”. The Part IV “Algorithm designing and visualization” is the description of designing and visualization on the example of an algorithm for calculating the earliest start time. The visualisation of mathematical algorithm in Excel interface allows us to understand and correlate complex, circular and recursion formulas in project network. The Part V “Programing the earliest and latest start time with VBA for Excel introduces interrelations between mathematical equations and computer algorithms. We discuss the program for calculating the earliest start time and connection between programming earlies and latest start time. The Part VI “Critical path modelling and programming” presents the critical path model based on integrated modelling and programming in Excel interface. In the Conclusion, we give a synthesis of our research.

2. Project network modeling by Critical Path Method (CPM)

Critical Path Method (CPM) is a project modelling technique applied for the first time in the chemical processing industries in the USA in 1957 to plan repairs schedule [1]. (IP4, 225). This technique uses a single value for activity duration. The Network Planning Technique is a method based on the graph theory, which is an area of operations research. With today's technology there exist many techniques to compute networks. The primary ones, however, were CPM and PERT. Recently, many advanced techniques and associated software have been developed, with the PRECEDENCE method for electronic computers developed by IBM as the initial one.

The first two aforementioned techniques were developed in the USA forty years ago and are still used today, in their original form. [4].

When processing data across a computer network, calculations are performed in tables. At the end of the process, tabled results for each activity are obtained, based on which a series of operations can be planned or performed without needing to redraw the network. All CPM values such as earliest start times (EST), earliest finish times (EFT), latest start times (LST), and latest finish times (LFT) appear in calendar days. When creating a network diagram in practice, shorter projects use shifts or days as the time unit, while larger projects that can last several months or years use weeks or months. For building reports intended for project investors or buyers (building constructions, shipbuilding etc.), time units are recalculated into calendar data. In such a case that network is made manually, the *Calendar Form* is used. On the basis of the calculated results, we can establish the start and finish time for each activity on the assumption that the project commencement date is defined [4].

The key factor that affects the project network management is finding the critical path, based on which total duration of the project can be established as well as optimum project activity schedule [6]. The network diagrams allow users to generate project activity schedule and identify duration for each activity as well as project chronology. The network diagrams help us to define the activity start time with respect to one or more preceding activities, and the activity finish time with respect to one or more succeeding activities. [9]. The network diagrams clearly indicate the manner in which activities are interrelated in terms of priority. Once the schedule is established, all relationships between the activities are explained through the algorithm. On the other hand, in Gantt charts, activities relationships have to be remembered. In complex projects, Gantt charts can therefore become very impractical. Rescheduling can be done automatically throughout the execution of the algorithm [8].

Mathematical modelling of network diagrams with constant times is based on the following assumptions [2], [4]:

- 1) The succeeding activity normally begins after the preceding activity is finished, or the activity with the longest duration, if there are more activities.
- 2) The earliest finish time of an activity is the sum of its earliest start time and duration.
- 3) The preceding activity finishes before the succeeding activity is started, or the activity with the least duration, if there are more activities.
- 4) The latest start time of an activity is the difference between the latest finish time of that activity and its duration.
- 5) The first activity including all starting activities, which are not preceded by any other activity have the zero start time.
- 6) The last activity, which does not have any succeeding activity has the latest finish value

equalling the value of the earliest finish time and corresponds to the total duration of the project.

- 7) The slack is the difference between the latest and earliest start time, or between the latest and earliest finish time. Activities that at the end of calculation have the zero slack are defined as critical activities. By linking critical activities we are able to establish the critical path.

3. Project network modeling in the spreadsheet interface

The project network modelling will be analyzed on the example of project of commercial building construction. All project activities have been identified in table 1 [3]. The first column contains activity abbreviations,. The second column shows data on mutually dependable activities (predecessors). The third column indicates activity duration. The key data are activity duration and mutual dependence. The predecessors are indicated in the third column, while the successors can be identified by analyzing predecessors with respect to the activity abbreviations in the first column.

Table 1. Project activities

Activity	Predecessors	Duration
A		2
B	A	4
C	B	10
D	C	6
E	C	4
F	E	5
G	D	7
H	E, G	9
I	C	7
J	F, I	8
K	J	4
L	J	5
M	H	2
N	K, L	6

The initial model of the project network in the spreadsheet interface is shown in the Scheme 1. Two fictive activities have been created in addition to the determined activities, specifically the activity *S* that indicates the commencement of the project, and activity *F* that indicates the completion of the project. The value of the starting activity is 0, while the finishing activity time is the maximum duration of the project. The predecessors and successors have been marked by 1 in the corresponding table cells. The table has been divided by diagonal line in two parts. In the left and lower part of the table, the preceding activities have been marked and the earliest start time can be calculated. In the right and upper part of the table, the successors have been marked and the latest start time can be calculated. The earliest start time can be

calculated using a recursive formula as shown in the Scheme 4.

Scheme 1. Project network model in the spreadsheet interface

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1																		0	S	
2				1														2	A	
3		1			1													4	B	
4			1			1	1											10	C	
5				1				1	1									6	D	
6					1				1	1								4	E	
7						1	1											5	F	
8							1			1								7	G	
9								1	1								1	9	H	
10				1														7	I	
11								1			1	1						8	J	
12												1						4	K	
13																		5	L	
14																		2	M	
15																		1	6	N
16																		0	F	
17	S	A	B	C	D	E	F	G	H	I	J	K	L	M	N	F				

4. Algorithm designing and visualization

Algorithm designing and visualization will be presented on the example of calculating the earliest start time (ES). The method of calculating the earliest start time (ES) for each activity is called the Forward Pass through the network. By calculating the earliest start time of an activity, we are able to consider the earliest finish time of that activity. Accordingly, we obtain the commencement time of the project. The first activity will therefore obtain the zero value because it has no predecessor. As it can be seen in the Table 3, the activity B is assigned the preceding activity duration of 12 days. The earliest start time of the activity D is assigned the latest finish time of the preceding activity B

After assigning the zero start value to the first (fictitious) activity, the earliest start time values of the given activities are calculated successively. The earliest start time of an activity is calculated in such a way that the earliest start time of its preceding activity is added up with the duration time of that preceding activity. The simplest way to calculate an activity's earliest start time is when that activity has a single preceding activity. In such a case, the activity's earliest start time is equal to the sum of its predecessor's earliest start time and duration time. The following is the equation for calculating the earliest start time (ES) [10].

$$ES_j = \max\{ES(i) + D_{ij} \mid i \in P(j)\}$$

The variable ES_j indicates the value of earliest start time for the j -activity, which we want to calculate. It can be seen from the right side of the equation that the earliest start time of the j -activity is calculated as the maximum value for n calculated sums of the earliest start times of the activities that precede the j -activity and duration times of those preceding activities. It means that the variable $ES(i)$ on the right side of the equation indicates the earliest start times of the i -activities, which precede the activity ES_j . The variable D_{ij} indicates the duration time of the activity i 's predecessor, which is

added up with the earliest start time of that activity, where i is the element of the set $P(j)$ that includes all activities, which precede the j -activity.

The algorithm of of the earliest start time calculation will be demonstrated through the example of the activity H as shown in the Scheme 2. It can be seen from the list of activities that the activity H is at the ninth position, including the fictitious starting activity S. Accordingly, ES for H is calculated in the cell I9 that corresponds to the intersection of the ninth row and the ninth column. In the cell A9:H9 in the ninth row the values of the earliest finish times for the activities E and G, preceding the activity H, are calculated. The maximum value in the cell A9:H9 is actually ES for H and is entered in the cell I9.

Scheme 2. The algorithm of of the earliest start time calculation

	A	B	C	D	E	F	G	H	I	Q	R
1										0	S
2		0					D_{ij}			2	A
3			2							4	B
4				6						10	C
5					16					6	D
6						16				4	E
7							20			5	F
8								22		7	G
9	0	2	4	10	6	20	5	<u>29</u>	<u>29</u>	9	H
17	S	A	B	C	D	E	F	G	H		
18										ES_j	

The earliest start times are calculated and positioned in diagonal lines, while the predecessors' duration times are recorded in the column Q. The sums of the earliest start times of the activities that precede the activity H and their duration times are calculated in the ninth row. It can be seen from the Scheme 2 that the activities E and G are preceding the activity H. The sum of E's earliest start time (cell F6, value 16) and its duration time (cell Q6, value 4) has been calculated and recorded in the cell F9 (value 20, bolded). The value of the sum of the earliest start time and duration time for the activity G has been calculated in the same way, and entered in the cell H9. By comparing the calculated values for E and G, it can be concluded that the succeeding activity G has a bigger value or 295 (bolded and underlined), which corresponds to the earliest start time of the activity I, as recorded in the cell I9.

5. Programming the earliest and latest start time with VBA for Excel

In the beginning of the program, we define program loops. We are able to navigate across the table using loops, j -loop to navigate between rows, and i -loop to navigate between columns. Maximum value is calculated in the external loop (j). Earliest start times are recorded in the nodes of j -activities. Sums of earliest start times of the

i-activities that precede the activity ES_j as well as duration time of those activities are calculated in the inner loop (*i*). Sum values are recorded in cells that belong to a row where the earliest start time for “*j*” activity is calculated where “*i*” is the element of the set $P(j)$, which includes all activities that precede the *j*-activity. Program for calculation the earliest and latest start time with VBA for Excel is shown in scheme 3.

The program for computing earliest start time will be demonstrated on the example of the activity H earliest start time, as shown in the Scheme 4. The algorithm has been explained in detail in the chapter 2. The Scheme 2 (lower part of the spreadsheet) is the comparison of the mathematical equation and VBA formula for calculating earliest start time, with corresponding variables in the formulas linked by arrows.

The right side of the equation represents the earliest start time of the *j*-activity or $Cells(j+1,i)$, calculated as the maximum value for *n* calculated sums of earliest start times of the activities that precede the *j*-activity or $Cells(i,i)$, and duration times of those predecessors or $Range(“Duration”).Cells(i).Value$. In VBA, value of each *i*-cell in the *j*-row ($Cells(j+1,i)$) that includes zeros and ones, in which process ones mean predecessors, is multiplied with the value of earliest start time of the corresponding *i*-node ($Cells(i,i)$) for the preceding activity.

It can be seen from the Scheme 4 that the recursion formula is set up by $Cells(i,i)$ to include previously calculated earliest start times of *i*-activities that precede *j*-activities. After calculating and entering the sums of all earliest start times in *i*-cells of *j*-row, the maximum value of earliest start time is recorded in the last “*i*” cell in the diagonal line of the triangular matrix, after leaving the loop *i* throughout the procedure $Cells(j+1,i) = Max$ (line 12).

After performing the forward pass, based on which we are able to obtain the value of earliest finish time for each activity in the network, the backward pass computing is next. The backward pass allows us to calculate the value of latest start time (LS) for each activity. At the beginning of the backward pass computation process, the latest start time of the last activity in the network diagram is equal to the earliest start time of that activity. It means that the last activity in the network diagram is assigned the equation $LS = ES$.

In the second part of the program, as shown in the Scheme 1 (Latest Start), latest start times of project activities are calculated using the backward pass method. In the beginning of this part of the program, the reference (initial) address coordinates are defined, using the variables *m* and *n*, to calculate the latest start times of project activities. The initial (reference) value in the backward pass computing process is the value of latest start time of the last (fictive) activity, which is equal to the value of earliest start time of that activity (line 16 in the program), as calculated in the first part of the program. The initial, minimum value (variable *Min*) is also equal to the value of earliest start time of the last activity (line 18 in the program). The earliest start time of the last, fictitious

activity is calculated in the first part of the program (Earliest Start), thereby the equations assigned to the program lines 16 and 18 are the links between the first and second part of the program.

Scheme 3. VBA program for calculation the earliest and latest start time

	1	Private Sub EarliestAndLatestStart_Click()
Earliest Start (ES)	2	For j = 1 To 15
	3	Max = 0
	4	For i = 1 To j
	5	$Cells(j + 1, i) = Cells(j + 1, i) * Cells(i, i) +$ $+ Range(“Duration”).Cells(i).Value$
	6	If $Cells(j + 1, i) > Max$ Then
	7	Max = $Cells(j + 1, i)$
	8	Else
	9	Max = Max
	10	End If
	11	Next i
	12	$Cells(j + 1, i) = Max$
	13	Next j
	Latest Start (LS)	14
15		n = 18
16		$Cells(m, n - 1) = Cells(m, n - 2)$
17		For i = 1 To 13
18		Min = $Cells(m, n - 2)$
19		For j = 1 To i
20		$Cells(m - i, n - j) = Cells(m - i, n - j) * Cells(m + 1 - j, n - j) -$ $- Range(“Duration”).Cells(m - i).Value$
21		If $Cells(m - i, n - j) > 1$ And $Cells(m - i, n - j) < Min$ Then
22		Min = $Cells(m - i, n - j)$
23		Else
24		Min = Min
25		End If
26		Next j
27		$Cells(m - i, n - j) = Min$
28		Next i
29		End Sub

Scheme 4: Programming of Earliest Start Time for Activity H

	A	B	C	D	E	F	G	H	I	Q	R	S	
1			Range(“Duration”).Cells(i).Value							0	S		
2		0								2	A		
3			2							4	B		
4				6						10	C		
5					16					6	D		
6						16				4	E		
7							20			5	F		
8								22		7	G		
9	0	2	4	10	6	20	5	29	29	9	H		
17	S	A	B	C	D	E	F	G	H				
18			$ES_j = Max\{ES(i) + D_{ij} \mid i \in P(j)\}$										
19													
20			$Cells(j+1,i) = Cells(j+1,i) * Cells(i,i) + Range(“Duration”).Cells(i).Value$										

After calculating the latest start time of the finishing activity, the values of latest start time for other activities are calculated successively. The latest start of an activity is calculated in such a way that the latest start time (LS) of the succeeding activity is subtracted by its duration time.

The following is the equation for calculating the latest start time (LS_i) [10].

$$LS_i = \min\{LS(j) - D_{ij} \mid j \in S(i)\}$$

The variable LS_i indicates the *i*-activity's latest start time value. It can be seen from the right side of the equation that the latest start time of the *i*-activity is calculated as the minimum value for *n* calculated differences between the latest start times of the activities, which succeed the *i*-activity and duration times of those succeeding activities. It means that the variable LS(*j*) on the right side of the equation indicates the latest start times of the *j*-activities, which succeed the activity LS_i. The variable D_{ij} indicates the duration time of the *j*-activity's successor, which is subtracted from the latest start time of that activity, where *j* is the element of the set S(*i*) that includes all activities that succeed the activity *i*. In latest start time programming, *i*-loop is the external loop and is used to navigate between rows, while *j*-loop is situated inside *i*-loop and enables us to navigate between columns.

6. Critical path modelling and programming

A set of network activities where the finishing milestone of an activity corresponds to the starting milestone of its successor is called the path. A series of interrelated activities, which spread from the starting to the finishing milestone in the network diagram or project and have the longest time in totality, are called the critical activities [2].

Based on the critical path calculation, we are able to calculate the total duration of the project and optimum project activity schedule.[7]. The project networks allow us to schedule project activities and identify activity duration and mutual dependence. In such a way, we can define the time when certain activity is to start with respect to one or more preceding activities, and the time when the activity is to finish with respect to the start time of one or more succeeding activities. [3]. Rescheduling can be done automatically throughout the execution of the algorithm.

In visual programming of the critical path, equations involved in mathematical models of project networks can be reprogrammed in Excel interface, and can be used for all problem types that are related to calculating critical path and minimum duration of the project. The total duration of the project changes as duration times of particular activities in the project change, where non-critical activities can become critical [1]. Visual programming of the critical path in Excel interface allows us to calculate automatically the critical path immediately upon inserting activities changes. Throughout the visual

programming of the critical path we obtain a graphical algorithm, which allows us to understand and link complex circular and recursive equations in a project network.

Model of calculating earliest and latest start times with VBA is shown in the scheme 5. This is the model of project network results obtained throughout the VBA program in Excel interface. It can be seen from the scheme that the series of results have been calculated and recorded in diagonal lines. Looking from left to right, the series of results in the first diagonal correspond to the earliest start times (ES), while the results in the second diagonal relate to the latest start times (LS).

Scheme 5. Model of calculating earliest and latest start times with VBA

Based on the calculation of the earliest and latest start time (Scheme 5), we are able to calculate slack times and define the critical path. Slack is a difference between

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	S	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	ES	0	S	
2	0	0	0	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	2	A
3	0	2	2	2	2	-4	-4	-4	-4	-4	14	-4	-4	-4	-4	-4	-4	-4	4	B
4	0	2	6	6	6	10	6	-10	-10	-10	8	-10	-10	-10	-10	-10	-10	-10	10	C
5	0	2	4	16	16	20	-6	-6	20	-6	-6	-6	-6	-6	-6	-6	-6	-6	6	D
6	0	2	4	16	6	16	16	16	-4	29	-4	-4	-4	-4	-4	-4	-4	-4	4	E
7	0	2	4	10	6	20	20	20	-5	-5	-5	20	-5	-5	-5	-5	-5	-5	5	F
8	0	2	4	10	22	4	5	22	26	26	-7	-7	-7	-7	-7	-7	-7	-7	7	G
9	0	2	4	10	6	20	5	29	29	33	-9	-9	-9	-9	-9	-9	-9	-9	9	H
10	0	2	4	16	6	4	5	7	9	16	18	18	-7	-7	-7	-7	-7	-7	7	I
11	0	2	4	10	6	4	25	7	9	23	25	25	26	25	-8	-8	-8	-8	8	J
12	0	2	4	10	6	4	5	7	9	7	33	33	34	-4	-4	34	-4	4	4	K
13	0	2	4	10	6	4	5	7	9	7	33	4	33	33	-5	33	-5	5	5	L
14	0	2	4	10	6	4	5	7	38	7	8	4	5	38	42	-2	42	2	2	M
15	0	2	4	10	6	4	5	7	9	7	8	37	38	2	38	38	38	6	6	N
16	0	2	4	10	6	4	5	7	9	7	8	4	5	40	44	44	44	44	44	F
17	LS	S	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P			

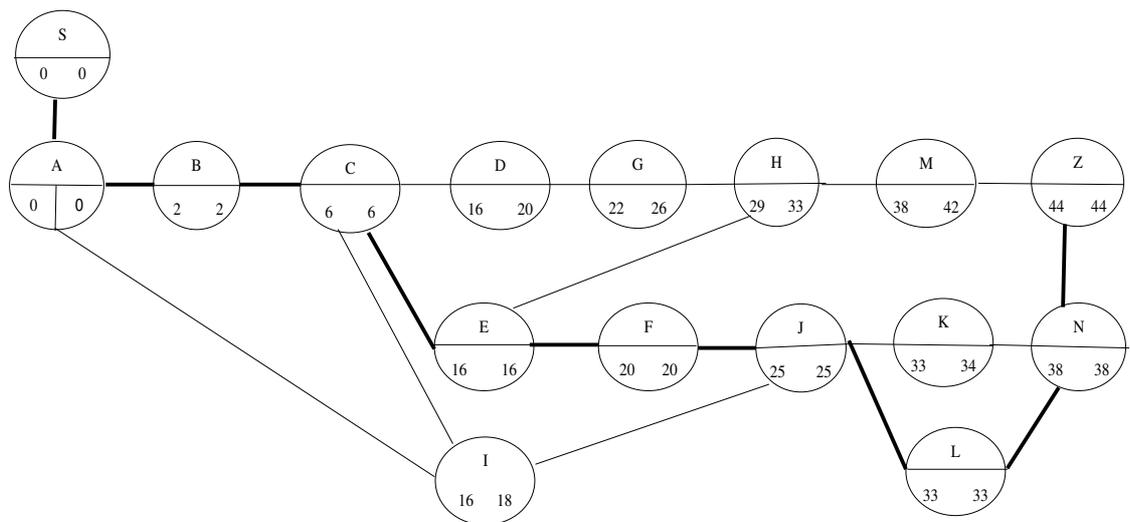
the latest and earliest start time. The critical path includes activities that have equal earliest and latest start times, or activities that are assigned the equation ES = LS [2]. In the given example, slack values have been calculated in the Scheme 6. The third diagonal indicates differences between the earliest and the latest start times for each activity. Activities with zero slack are called critical activities, and are always found on one or more critical paths. This (or these) critical path(s) determine minimum project duration since all of them must be completed as part of the project, and all noncritical activities can be completed concurrently with the critical activities (CPM). The solution of project network model by CPM is shown in scheme 6.

In the given example, those are activities ABCEJLN. The critical path diagram is shown in the Scheme 7. The critical path activities are doubly linked. The slack has a very practical significance in network planning. The slack indicates when, under what conditions and for how many time units can a start or finish time be postponed.

Scheme 6. The solution of project network model by CPM

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1		ES	LS															0	S	
2	A	0	0	→	0													2	A	
3		B	2	2	→	0												4	B	
4			C	6	6	→	0											10	C	
5				D	16	20	→	4										6	D	
6					E	16	16	→	0									4	E	
7						F	20	20	→	0								5	F	
8							G	22	26	→	4							7	G	
9								H	29	33	→	4						9	H	
10									I	16	18	→	2					7	I	
11										J	25	25	→	0				8	J	
12											K	33	34	→	1			4	K	
13												L	33	33	→	0		5	L	
14													K	38	42	→	4	2	M	
15														N	38	38	→	0	6	N
16															F	44	44		F	
17	S	A	B	C	D	E	F	G	H	I	J	K	L	M	N	F				

Scheme 7. Critical path diagram



7. Conclusion

In this paper, we describe object modelling and programming in algorithm visualization on the example of critical path method in project network. Object Modelling and Programming in VBA for Excel helps us to create the conditions for integrated modelling and programming. We discuss the integration of modelling and programming on the example of algorithm creation and visualization to calculate the critical path in a project network, in which process the following methodological frame can be suggested:

1. Understanding and visualizing mathematical equations/algorithms (visualizing and correlating recursion, circular and matrix formulas in project network)
2. Integrating mathematical equations and computer algorithms (for instance, correlating the mathematical equation with the computer program formula for calculating the earliest start time)
3. Creating and developing computer algorithms, implementing and integrating mathematical equations in a computer program

Visualization, iterativity and synergy in a combined utilization of human and software resources are the main characteristics of integrated modelling and programming in Excel interface. Synergised utilisation and a simultaneous view of the model development process using computer tools, and program development process using VBA can encourage logical and inventive thinking in understanding and resolving problem situations.

Integrated modelling and programming based on object-oriented technology, and program visualization in Excel interface can be used as a tool to understand and find an effective solution to a problem, as well as a mean to create and interpret ideas. The program development process enables us to conceive essential elements of a problem, which were not clearly identifiable, allowing us not only to resolve the existing problem, but also to redefine the problem and simultaneously upgrade the program as the problem develops.

References:

- [1]. Barković, D., Uvod u operacijski management, Eonomski fakultet, Osijek, 1999.
- [2]. Dobrenić S., Operacijska istraživanja, Fakultet organizacije i informatike Varaždin, Sveučilište u Zagrebu, Varaždin, 1977.
- [3]. Hillier, F.S., Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets, McGraw-Hill Higher Education; 3rd edition (June 2007)
<http://highered.mcgraw-hill.com/sites/dl/free/0073129038/449984/Chapter16.pdf>
- [4]. Kovačiček, F. (et all), Inženjerski priručnik IP4, Proizvodno strojarstvo, Školska knjiga, Zagreb, 1998.
- [5]. Magdić, D., Programski alati za matematičko modeliranje,
http://zpi.ptfos.hr/modeli/Presentations/3_Modeli_osnove2.ppt
- [6]. Mathur, K., Solow, D., Management Science, The Art of Decision Making, Prentice Hall, Englewood Cliffs, New Jersey, 1994.
- [7]. Pašagić, H., Matematičko modeliranje i teorija grafova, Zagreb, Fakultet prometnih znanosti, Zagreb, 1998
- [8]. Schroeder, R.G., Upravljanje proizvodnjom, MATE, Zagreb, 1999.
- [9]. Smith, D., E., Quantitative Business Analysis, Wiley/Hamilton Publication, University of California, Los Angeles, USA, 1977.
- [10]. Zhao, T., Tseng, C.L., A note on activity floats in activity-on-arrow networks, Journal of tge Operation Research Society (2003) 54, p. 1296-1299, USA, 2003,
<http://www.palgrave-journals.com/jors>