SILESIAN UNIVERSITY OF TECHNOLOGY PUBLISHING HOUSE

SCIENTIFIC PAPERS OF SILESIAN UNIVERSITY OF TECHNOLOGY 2025
ORGANIZATION AND MANAGEMENT SERIES NO. 234

SPREADSHEET AS THE CPM METHOD SUPPORT SOLUTION

Bartosz SZCZĘŚNIAK

Silesian University of Technology, Faculty of Organisation and Management, Institute of Economics and Informatics; bartosz.szczesniak@polsl.pl, ORCID: 0000-0002-9683-4629

Purpose: The main research purpose underlying this study is to develop and present a spreadsheet-based tool supporting the CPM method in line with the MiRel concept.
Design/methodology/approach: Based on a literature review, a CPM method variant for which the tool in question will be developed has been chosen, and it has been analysed in detail. The data required to be entered to define the analysed project have been determined, and a structure aligned with IDEF1x has been presented with reference to these data. Mechanisms enabling fully automatic execution of the method's consecutive steps, making it possible to calculate adequate time values for events and activities, as well as fully automatic identification of critical paths have been developed for data stored according to the structure proposed.
Findings: A spreadsheet provides a fully sufficient environment to create a tool supporting the CPM method. For the method version based on a graph created in line with the Activity on Arrow approach, such a graph, along with all the required data, can be defined within a relatively simple table structure. Assuming that only built-in spreadsheet functions are to be used, for the said structure, it is possible to develop mechanisms which automatically calculate the earliest and latest occurrence times for events as well as floats for events and activities, and mechanisms which automatically identify all critical paths. All the actions performed require a specific sequence of events and activities to be established. This can be achieved using matrix calculus similarly to the manner in which it is implemented for cost management purposes.
Practical implications: The solution presented in the paper can be successfully applied in practice to support the CPM method. As proposed, the concept of a mechanism enabling all paths to be identified in a directed graph can be employed in any other tool where path identification is necessary.
Originality/value: The concept described in the paper is the author's original solution.
Keywords: spreadsheet, database, data processing, Microsoft Excel, MiRel, project management.
Category of the paper: Conceptual and technical paper.

1. Introduction

In the contemporary world, where every person and every organisation is forced to constantly process enormous amounts of data, spreadsheets continue to form a group of programs which are often and willingly chosen to support processes implemented in that sphere (Barreto, 2015; Cocking; 2017). There are various concepts mentioned in the literature as to how they can be put into practical use in different operating areas of organisations. The solutions described in numerous papers include those dedicated to work time organisation (Zasadzień et al., 2017), quality management (Knight, 2009; Szczęśniak, Molenda, 2013; Szczęśniak, 2020, 2021, 2022, 2023a), project management (Szczęśniak, 2023b, 2024) and problems related to sales, marketing, and finance (Jackson, Staunton, 2004; Próchnicki, 2012; Winston, 2019; Najihah, Hanyza, 2019; Kusztelak, 2020). Despite numerous studies discussing the use of spreadsheets to support organizational processes and methods used in management, the literature does not offer any proposals for applying them to support the CPM method. The concept of a tool designed to assist this method is presented later in the article. In order to make sure that the tools created in spreadsheets are well organised, the data collected in them can be stored in an explicitly defined structure consistent with the relational data model. Such a method of data organisation is proposed under the author's signature MiRel concept (Szczęśniak, 2017, 2018a, 2018b, 2020a). A tool aligned with this concept, aimed at supporting the CPM method, has been presented and discussed further on in this study. The tool described later in the article was created in accordance with this concept.

2. Methodology

Based on literature research, various variants of the CPM method were identified. One of these variants was selected as the variant for which a suitable spreadsheet tool would be developed. This variant was analysed and described in detail to precisely define its input data and all the steps involved. Based on the analysis, a data structure for the developed tool was proposed, consistent with the relational data model. A method for presenting the input data structure in the spreadsheet tool was proposed. Based on the input data structure, solutions were developed that enabled fully automated execution of all identified method steps. In line with the MiRel concept, all solutions were developed using built-in spreadsheet functions and mechanisms. The tool was created using the desktop version of Excel, available as part of the Microsoft 365 suite.

3. Literature review: Critical Path Method

CPM is one of network programming methods, and it has been extensively described in the literature on the subject (Kerzner, 2009; Kukuła et al., 2011; Aja, Chukwu, 2017). In this method, the project subject to analysis is presented using a network diagram, being a directed graph. As mentioned in the literature, there are two approaches to how such a graph is constructed: Activity on Arrow, or a PERT chart, and Activity on Node. It is the former that has been addressed under this study. According to this approach, in a graph being created, vertices represent events, while edges are activities. The activities found in the graph represent any identified part of a given project. Under activities, specific work is performed and resources are consumed. They are characterised by a specific duration. Activities include a group of dummy activities with a zero duration time. They occur in relation to the necessity of following certain rules applicable to the creation of graphs. The events occurring in the graph correspond to the states in which a project may be found while it is being implemented. A specific state is associated with reaching a certain milestone, i.e. the completion of a certain group of activities. The symbols used in the network diagram according to the Activity on Arrow approach have been shown in Figure 1.

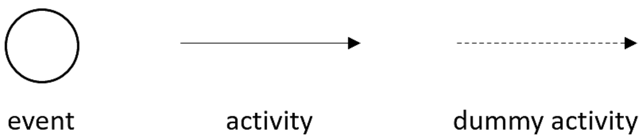


Figure 1. Symbols used in the Activity on Arrow approach.
Source: own work.

The direction of arrows determines the sequence in which the consecutive project milestones, represented by successive events, can be achieved. The activities represented by the arrows coming into a given event must be completed prior to the activities represented by the arrows going out of that event. An example of a very simple graph illustrating a project has been provided in Figure 2.

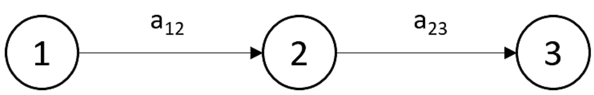


Figure 2. Example of a simple graph aligned with the Activity on Arrow approach.
Source: own work.

In this example, the project is initially at the stage represented by event 1. At this stage, it is possible to perform activity a12. After this activity has been completed, the stage represented by event 2 is reached, where it is possible to perform activity a23. Once this activity has been performed, the stage to which event 3 corresponds is reached and the entire project is completed.

According to Kukuła's description (Kukuła et al., 2011), while creating a graph, it must additionally meet the following conditions. There must be exactly one starting vertex and one ending vertex in the graph, and it must contain no cycles. The arrows representing activities must not intersect. There can only be one edge between two vertices, meaning that two identified events can be linked by only one activity. Where two events should be linked by several activities, additional events and dummy activities are to be added. Such a situation has been illustrated in Figure 3. In this example, activity b would be the second activity linking events 1 and 2. Consequently, one must add event 3 and dummy activity c in the diagram.

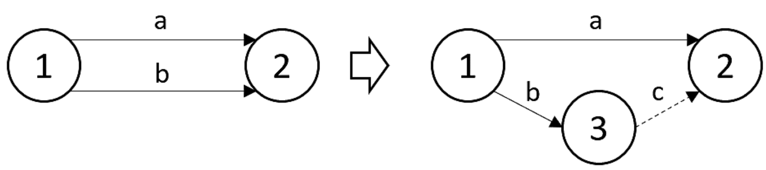


Figure 3. Example of how a dummy activity is used.
Source: own work.

During the analysis, the following quantities are calculated for each event:

- t – the earliest occurrence time of an event,
- T – the latest occurrence time of an event,
- L – float.

Assuming that activity a_{ij} goes out of event *i* and comes into event *j*, and that it is the only activity coming into event *j*, time t for event *j* is calculated according to the following formula:

$$t_j = t_i + d_{i-j} \tag{1}$$

where:
t_j – the earliest occurrence time of event j,
t_i – the earliest occurrence time of event I,
d_{i-j} – duration of activity a_{ij}.

Assuming that several activities come into event j and that set P is a set of events from which these activities go out, time t for event *j* is calculated according to the following formula:

$$t_j = \max_{i \in P}(t_i + d_{i-j}) \tag{2}$$

The sequence in which times t are calculated for individual events is from the starting event to the ending event. For the starting event, one assumes that t = 0.

Assuming that activity a_{ij} goes out of event *i* and comes into event *j*, and that it is the only activity going out of event *j*, time T for event *i* is calculated according to the following formula:

$$T_i = T_j - d_{i-j} \tag{3}$$

where:

- T_i – the latest occurrence time of event *i*.
- T_j – the latest occurrence time of event *j*.
- d_{i-j} – duration of activity a_{ij}.

Assuming that several activities go out of event *i* and that set S is a set of events for which these activities are incoming, time T for event *i* is calculated according to the following formula:

$$t_i = \min_{j \in S}\left(T_j - d_{i-j}\right) \tag{4}$$

The sequence in which times T are calculated for individual events is from the ending event to the starting event. For the ending event, it is assumed that the latest occurrence time of the event equals its earliest occurrence time, i.e. T = t.

Based on the earliest and latest occurrence time of the event, float is calculated for each event. Float is calculated according to the following formula:

$$L = T - t \tag{5}$$

Float is also calculated for individual activities. Assuming that activity a_{ij} goes out of event *i* and comes into event *j*, the float is calculated according to the formula below:

$$F_{i-j} = T_j - t_i - d_{i-j} \tag{6}$$

where:

- F_{i-j} – float for activity a_{ij}.
- T_j – the latest occurrence time of event *j*.
- t_i – the earliest occurrence time of event *i*.
- d_{i-j} – duration of activity a_{ij}

Once all times and floats have been calculated for events and activities, one proceeds with the identification of critical paths, i.e. sequences of activities leading from the starting event to the ending one, such that each activity they feature has a float of 0.

4. Results: spreadsheet tool

A CPM-based analysis can be performed using dedicated tools. However, it can also be conducted with a tool created in a spreadsheet. The tool presented in this article makes it easy to define, modify, and expand any network representing a project subject to analysis, aligned with the Activity on Arrow approach, as well as to automatically execute consecutive CPM steps. In accordance with the MiRel concept, it is assumed that the aforementioned network will be described in the said tool using data entered into corresponding tables, as in a conventional relational database. Based on these data, all the required values for the earliest and latest occurrence times of individual events as well as the floats will be calculated for the events and activities. With these values at hand, critical paths will be automatically identified. The tool described in the paper has been developed using Excel from the Microsoft 365 suite.

4.1. Data structure applied in the tool

The data structure applied to the problem at hand is relatively simple. It has been proposed that it should include entities activity and event. The relationships between the entities and the attributes identified within them have been presented in an IDEF1x-aligned model (Allen, 2006) shown in Figure 4.

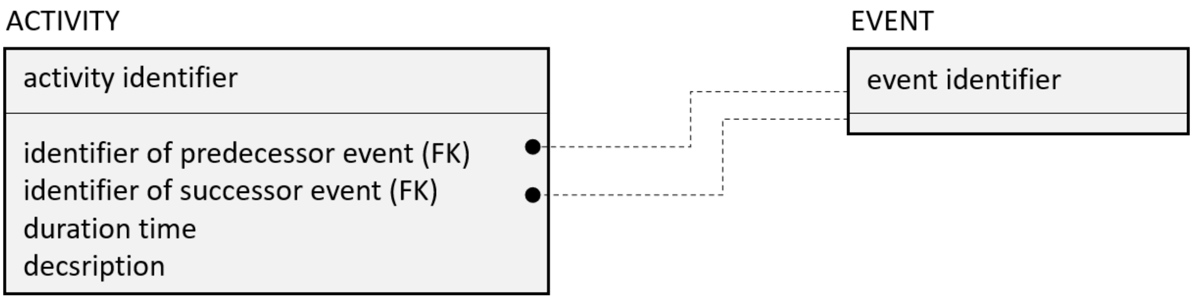


Figure 4. Data structure applied in the tool proposed.

Source: own work.

The above entities are presented in the tool as assumed under the MiRel concept for type I of entity representation. Consequently, the tool contains two tables: table A for the Activity entity, and table E for the Event entity. They have been entered in separate worksheets, named after the corresponding tables. Their layout has been shown in Figure 5.

Table: A
Worksheet: A

	A	B	C	D
1	ID_A	DT	ID_PE	ID_SE
2	c	8	3	4
3	a	4	1	2
4	d	5	2	3
5	b	11	2	4

Table: E
Worksheet: E

	A
1	ID_E
2	4
3	2
4	3
5	1

Figure 5. Entity representation in the tool.

Source: own work.

The tables use abbreviated attribute names: ID_A for activity identifier, DT for duration time, ID_PE for predecessor event identifier, ID_SE for successor event identifier, and ID_E for event identifier. It has been assumed that 30 activities and 30 events can be entered in the tool. Consequently, it is possible to enter data into up to 31 rows. And such a range has also been taken into account in the formulas discussed further on in the paper.

4.2. Determining the sequence of calculations

The actions related to determining individual duration times and floats are performed in a specific sequence. Since it has been assumed that the data describing the project are stored in tables such as those of a conventional relational database, both the activities and events can be entered into tables in any chosen order. In the first step, before one proceeds with the actual calculations, it is necessary to determine the sequence in which they should occur in the calculations. To this end, matrix calculus has been employed in a similar way to how it is used in the sphere of cost management (Nowak, 1994, 2003). Assuming that n activities have been identified, the relationships between them can be represented using the following matrix

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & a_{1n} \\ b_{21} & b_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & p_{n1} & \cdots & a_{nn} \end{bmatrix} \tag{7}$$

where b_{ij} can take the value of 0 or a chosen value of x greater than 1. If $b_{ij} = x$, activity i directly precedes activity j. If $b_{ij} = 0$, activity i does not directly precede activity j. The number b_{ij} can be understood as a hypothetical standard of use of the unit of activity i per a unit of activity j. For each activity, a hypothetical quantitative flow will be determined. And for each activity which does not precede another activity, the quantitative flow of 1 has been envisaged. These flows are represented by the following vector:

$$c = \begin{bmatrix} c_1 \\ c_2 \\ \cdots \\ c_n \end{bmatrix} \tag{8}$$

where c_i equals 1 for the activities which do not precede any other activities and 0 for all the remaining activities. The hypothetical quantitative flows for all the values are represented by the following vector

$$d = \begin{bmatrix} d_1 \\ d_2 \\ \cdots \\ d_n \end{bmatrix} \tag{9}$$

This vector is calculated according to the formula below:

$$d = (I - B)^{-1} \cdot c \tag{10}$$

Calculated by this method for each activity, the hypothetical quantitative flows can be used as a coefficient of sequence because – given that the value x in matrix B is greater than 1 – the hypothetical quantitative flow for the preceding activity is always greater than the flow for the activity being preceded. The coefficient of sequence for an activity is also transferred to the event for which that activity is incoming. Therefore, it is the ending event that bears the lowest value of the coefficient. In this case, the value is 1. The highest value should be assigned to the starting event. Since no activity comes into this event, the latter is assigned a value that is greater by 1 than the largest value in vector d. When calculating the earliest occurrence time for an event, calculations for the events having a higher coefficient of sequence must be performed first. And when the latest occurrence time is being calculated for an event, that order must be reversed.

In the tool described in the paper, the consecutive steps aimed at determining the coefficient of sequence for individual activities have been placed in the auxiliary worksheet OR. One table and two auxiliary columns have been added to this worksheet. Their layout along with the formulas applied have been shown in Figure 6. The range of cells in which each formula is entered has been specified. The formula shown corresponds to the upper left cell of that range. Some of the formulas have the array form. These formulas have been entered in the tool as dynamic array formulas, as proposed by Microsoft in 2018 (Guidelines and examples of array formulas, https://support.microsoft.com). In addition to the range of cells in which these formulas are entered, the range of cells where results may appear according to the assumptions adopted is also stated after the semicolon, next to such formulas.

	A	B	C	D	E	AF	AG	AH
	FA1		FA2	FA3			FA4	FA5
1	AT1	c	a	d	b		AC1	AC2
2	c	0	0	0	0		1	1
3	a	0	0	1,1	1,1		0	2,31
4	d	1,1	0	0	0		0	1,1
5	b	0	0	0	0		1	1

FA1	A2:A31	`=IF(A!A2<>"";A!A2;"")`
FA2	B1; B1:AE1	`=IF(TRANSPOSE(A!A2:A31)<>"";TRANSPOSE(A!A2:A31);"")`
FA3	B2:AE31	`=IF(AND($A2<>"";B$1<>"");IF(INDEX(A!$D$2:$D$31;MATCH(OR!$A2;A!$A$2:$A$31;0);1)=INDEX(A!$C$2:$C$31;MATCH(OR!B$1;A!$A$2:$A$31;0);1);1,1;0);0)`
FA4	AG2:AG31	`=IF(A2<>"";IF(SUM(B2:AE2)=0;1;0);0)`
FA5	AH2; AH2:AH31	`=MMULT(MINVERSE(IF(ROW(A2:A31)=COLUMN($B$1:$AE$1);1;0)-B2:AE31);AG2:AG31)`

Figure 6. Auxiliary table and columns in worksheet OR with the formulas applied.

Source: own work.

Formulas FA1, FA2, and FA3 create a matrix consistent with matrix B; formulas FA1 and FA2 generate headers for consecutive rows and columns, while formula FA3 enters the value of 1.1 in the case that the activity from the row header comes into the event from which the activity from the column header goes out. Formula FA4 found in auxiliary column AC1 generates vector c. Based on the matrix and vector thus generated, formula FA5 generates vector d according to formula 10. The values of this vector are used in the subsequent steps to put the events and activities in the right order.

4.3. Calculating duration times and floats

The successive steps related to the calculation of times for individual events have been performed in worksheet E; this is where three columns and five auxiliary tables have been added for purposes of calculating the earliest occurrence time of an event. Their layout along with the formulas applied have been shown in Figure 7. According to the mechanism proposed, formula FB1 in auxiliary column BC1 first returns the coefficient of sequence for each event in column ID_E, as per the rules discussed above. Based on this coefficient, formula FB2 in auxiliary column BC2 determines the sequence of events in which the relevant calculations will be performed. Events with the highest coefficient value must be placed at the beginning. Formula FB4 in table BT1 identifies all activities for which the event in the current row in column BC2 is a successor event. The first of the events in column BC2 is always the starting event. There are no activities for which this event would be a successor event, and so the first row in each of the auxiliary tables remains empty and all the formulas applied appear from the second row onwards. It has been assumed that each event can have a maximum of four incoming activities, and therefore, all auxiliary tables have four columns. Found in table BT2, formula FB5 determines the duration time of the corresponding activity from table BT1. Formula FB6 applied in table BT3 returns events. Each of these events is a predecessor event for the corresponding activity from table BT1. For these events, formula FB7 in auxiliary table BT4 determines the earliest occurrence time of the event. According to the set sequence of calculations, the form of the formula applied in each row corresponds to the values of the earliest occurrence times found in the rows above the current row. Found in table BT5, formula FB8 determines all the earliest occurrence times resulting from all the activities coming into the given event for the event in the given row of auxiliary column BC2. Based on that determination, formula FB3 in column t calculates the final value of the earliest occurrence time for this event. Formula FB3 ignores the first row of the column because, in this case, the earliest occurrence time for the starting event is 0.

	A	D	E	F	H	I	J	K	M	N	O	P	R	S	T	U	W	X	Y	Z	AB	AC	AD	AE
		FB1	FB2	FB3		FB4				FB5				FB6				FB7				FB8		
1	ID_E	BC1	BC2	t	BT1				BT2				BT3				BT4				BT5			
2	4	1	1	0																				
3	2	2,31	2	4	a				4				1				0				4			
4	3	1,1	3	9	d				5				2				4				9			
5	1	3,31	4	17	c	b			8	11			3	2			9	4			17	15		

FB1	D2:D31	=IF(A2<>"";IF(COUNTIF(A!$D$2:$D$31;A2)>0; INDEX(OR!$AH$2:$AH$31;MATCH(A2;A!$D$2:$D$31;0);1);MAX(OR!$AH$2:$AH$31)+1);0)
FB2	E2; E2:E31	=SORTBY(A2:A31;D2:D31;-1)
FB3	F3:F31	=MAX(AB3:AE3)
FB4	H3:H31; H3:K31	=IF(E3<>0;TRANSPOSE(FILTER(A!$A$2:$A$31;A!$D$2:$D$31=E!E3;""));"")
FB5	M3:P31	=IF(H3<>"";INDEX(A!$B$2:$B$31;MATCH(E!H3;A!$A$2:$A$31;0);1);"")
FB6	R3:U31	=IF(H3<>"";INDEX(A!$C$2:$C$31;MATCH(E!H3;A!$A$2:$A$31;0);1);"")
FB7	W3:W31	=IF(H3<>"";INDEX($F$2:$F2;MATCH(R3;$E$2:$E2;0);1);"")
FB8	AB3:AE31	=IF(H3<>"";M3+W3;"")

Figure 7. Auxiliary columns and tables along with the formulas applied in them, added in worksheet E to calculate the earliest occurrence time of the event.

Source: own work.

For purposes of calculating the latest occurrence time of the event, three columns and five auxiliary tables have been added in worksheet E. Their layout along with the formulas applied in them have been shown in Figure 8.

In the first place, one must determine the sequence of events. To this end, formula FC1 in auxiliary column CC1 returns the coefficient of sequence for each event in column ID_E, as per the rules discussed above. Based on its value, formula FC2 found in auxiliary column CC2 determines the sequence of events according to which calculations will be performed. In this case, events with the lowest coefficient of sequence must come at the beginning. Formula FC5 in table CT1 identifies all activities for which the event in column CC2 is the predecessor event. The first of the events in column CC2 is always the ending event. There are no activities for which this event would be a predecessor event, and therefore, the first row in each of the auxiliary tables remains empty and all the formulas applied appear from the second row onwards.

	A	AG	AH	AI	AJ	AK	AL	AM	AN	AP	AQ	AR	AS	AU	AV	AW	AX	AZ	BA	BB	BC	BE	BF	BG	BH
1	ID_E	FC1 CC1	FC2 CC2	FC3 T	FC4	CT1		FC5		CT2		FC6		CT3		FC7		CT4		FC8		CT5		FC9	
2	4	1	4	17																					
3	2	2,31	3	9		c				8				4				17				9			
4	3	1,1	2	4		d	b			5	11			3	4			9	17			4	6		
5	1	3,31	1	0		a				4				2				4				0			

FC1	AG2:AG31	=IF(A2<>"";IF(COUNTIF(A!$D$2:$D$31;A2)>0;INDEX(OR!$AH$2:$AH$31; MATCH(A2;A!$D$2:$D$31;0);1);MAX(OR!$AH$2:$AH$31)+1);"")
FC2	AH2; AH2:AH31	=SORTBY(A2:A31;AG2:AG31;1)
FC3	AI2	=INDEX($F$2:$F$31;MATCH(AH2;$E$2:$E$31;0);1)
FC4	AI3:AI31	=MIN(BE3:BH3)
FC5	AK3:AK31; AK3:AN31	=IF(AH3<>0;TRANSPOSE(FILTER(A!$A$2:$A$31;A!$C$2:$C$31=E!AH3;""));"")
FC6	AP3:AS31	=IF(AK3<>"";INDEX(A!$B$2:$B$31;MATCH(E!AK3;A!$A$2:$A$31;0);1);"")
FC7	AU3:AX31	=IF(AK3<>"";INDEX(A!$D$2:$D$31;MATCH(E!AK3;A!$A$2:$A$31;0);1);"")
FC8	AZ3:BC31	=IF(AK3<>"";INDEX($AI$2:$AI2;MATCH(AU3;$AH$2:$AH2;0);1);"")
FC9	BE3:BH31	=IF(AK3<>"";AZ3-AP3;"")

Figure 8. Auxiliary columns and tables along with the formulas applied in them, added in worksheet E to calculate the latest occurrence time of the event.

Source: own work.

It has been assumed that each event can have a maximum of four outgoing activities, and therefore, all auxiliary tables have four columns each. Found in table CT2, formula FC6 determines the duration of the corresponding activity from table CT1. Formula FC7 applied in table CT3 returns events. Each of these events is a successor event for the corresponding activity from table CT1. For these events, formula FC8 in auxiliary table CT4 determines the latest occurrence time of the event. According to the set sequence of calculations, the form of the formula applied in each row corresponds to the values of the latest occurrence times found in the rows above the current row. Found in table CT5, formula FC9 determines all the latest occurrence times resulting from all the activities going out of the given event for the event in the given row of auxiliary column CC2. Based on that determination, formula FC4 calculates the final value of the latest occurrence time for this event in column T. Formula FC3 ignores the first row of the column because, in this case, the latest occurrence time for the ending event is equal to the earliest occurrence time of that event, and it is determined according to formula FC4.

Once the values of the earliest and latest occurrence time of the event have been calculated, it is possible to calculate the float values for the events and activities. For that purpose, three auxiliary columns have been added in worksheet E and worksheet A, respectively. These columns, along with the formulas applied in them, have been shown in Figure 9.

Worksheet: E

	A	BJ	BK	BL
		FD1	FD2	FD3
1	ID_E	DC1	DC2	L
2	4	17	17	0
3	2	4	4	0
4	3	9	9	0
5	1	0	0	0

Worksheet: A

	A	B	C	D	F	G	H
					FE1	FE2	FE3
1	ID_A	DT	ID_PE	ID_SE	EC1	EC2	F
2	c	8	3	4	17	9	0
3	a	4	1	2	4	0	0
4	d	5	2	3	9	4	0
5	b	11	2	4	17	4	2

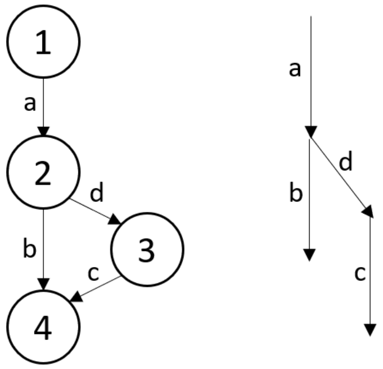
FD1	BJ2:BJ31	`=IF(A2<>"";INDEX($AI$2:$AI$31;MATCH(A2;$AH$2:$AH$31;0);1);"")`
FD2	BK2:BK31	`=IF(A2<>"";INDEX($F$2:$F$31;MATCH(A2;$E$2:$E$31;0);1);"")`
FD3	BL2:BL31	`=IF(A2<>"";BJ2-BK2;"")`
FE1	F2:F31	`=IF(A2<>"";INDEX(E!$AI$2:$AI$31;MATCH(A!D2;E!$AH$2:$AH$31;0);1);"")`
FE2	G2:G31	`=IF(A2<>"";INDEX(E!$F$2:$F$31;MATCH(A!C2;E!$E$2:$E$31;0);1);"")`
FE3	H2:H31	`=IF(A2<>"";F2-G2-B2;"")`

Figure 9. Auxiliary columns added in worksheets E and A to calculate the float of events and activities. Source: own work.

For purposes of the calculations performed for the events, in column DC1, formula FD1 determines the latest occurrence time of the event from column ID_E. Found in column DC2, formula FD2 determines the earliest occurrence time of that event. Based on both the values thus established, formula FD3 calculates the float applicable to the given case. For purposes of the calculations performed for the activities, in column EC1, formula FE1 determines the latest occurrence time of the successor event for the activity in column ID_A. Found in column EC2, formula FE2 determines the earliest occurrence time of the predecessor event for that activity. With reference to the values thus established and the activity duration, formula FE3 calculates the float applicable to this activity.

4.4. Identifying critical paths

The process in which critical paths are identified has been divided into two steps. In the first step, all possible sequences of activities forming paths from the starting event to the ending event are identified. In the second step, those which are not critical paths are eliminated from the set of all paths, and additionally, dummy activities are removed from the critical paths. The solution proposed for the first step is based on an assumption that all the identified paths will appear in the target table, where each column of the table represents a sequence of activities forming one of the paths identified. This solution draws from the observation that, in the target table, with an identical sequence of activities occurring earlier in a given path, a given activity in a given row appears in subsequent columns as many times as the total number of the necessary occurrences of all its successors counts. The incoming activities of the ending event occur only once. The very idea behind how this mechanism works for a simple network, defined in Figure 5, has been illustrated in Figure 10. Activities b and c come into the ending event, which means that they occur once in a given row of the table with the same sequence of preceding activities. For activity d, the only successor is activity c, so activity d also occurs only once. Activity a has two successors. Each of them occurs once, and consequently, activity a occurs twice.

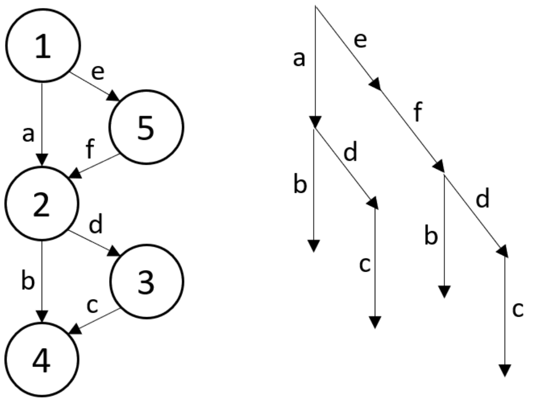


a	a
b	d
	c

Figure 10. Graphical form of the network defined in Figure 5; all potential paths represented as a tree and as a table obtained using the tool proposed.

Source: own work.

Figure 11 illustrates a slightly more extended case. Activities b and c come into the ending event, as previously, meaning that they occur once in a given row of the table with the same sequence of preceding activities. Activity c occurs once with the <ad> sequence of preceding activities and once with the <efd> sequence of preceding activities, while activity b – once with the <a> sequence and once with the <ef> sequence. The only successor of activity d is activity c, and therefore, activity d also occurs once with the <a> sequence and once with the <ef> sequence. The successors of activities a and f are activities b and d, both also occurring once. Consequently, activity a occurs twice with an empty preceding sequence, while activity f occurs twice with the <e> preceding sequence. The only successor of activity e is activity f, so activity f occurs twice with an empty preceding sequence.



a	a	e	e
b	d	f	f
	c	b	d
			c

Figure 11. Identification of all potential paths – example 2.

Source: own work.

Critical paths are identified in the tool in question in an additional worksheet named CP. For purposes of identification of all the potential paths, five columns and nine auxiliary tables have been added in this worksheet. Their layout along with the formulas applied in them have been shown in Figures 12 and 13.

	A	C	D	E	F	G	H	I	K	L	M	N	P
1	FF1 FC1	FF2 FC2	FF3 FC3	FF4 FT1				FF5 FC4	FF6 FT2				FF7 FC5
2	c	c						0					1
3	b	b	4					0					1
4	d	d	3	c				1	1				1
5	a	a	2	d	b			2	1	1			2
6		st	1	a				1	2				2

FF1	A2; A2:A31	=SORTBY(OR!$A$2:$A$31;IF(OR!$AH$2:$AH$31>0;OR!$AH$2:$AH$31;"");1)
FF2	C2:C31	=IF(A2<>"";A2;IF(A1<>"";"st";""))
FF3	D3:D31	=@IF(C3<>"";IF(C3="st";FILTER(E!$A$2:$A$31;COUNTIF(A!$D$2:$D$31; E!$A$2:$A$31)=0;"");INDEX(A!$D$2:$D$31;MATCH(CP!C3;A!$A$2:$A$31;0);1));"")
FF4	E3; E3:H31	=IF(C3<>"";TRANSPOSE(FILTER(A!$A$2:$A$31;A!$C$2:$C$31=CP!D3;""));"")
FF5	I2:I31	=IF(C2<>"";COLUMNS(E2:H2)-COUNTBLANK(E2:H2);0)
FF6	K3:N31	=IF(E3<>"";INDEX($P$2:$P2;MATCH(E3;$C$2:$C2;0));"")
FF7	P3; P3P31	=IF(C3<>"";IF(SUM(K3:N3)>0;SUM(K3:N3);1);"")

Figure 12. Auxiliary columns and tables in worksheet CP – part 1.

Source: own work.

Formula FF1 to be found in auxiliary column FC1 returns activities arranged in ascending order based on the coefficient of sequence previously calculated. For these activities, formula FF2 adds an additional technical activity *st* at the end of the list. All the identified paths will start with this activity. It will be deleted form the paths at the stage of the final critical path identification. In the next step, formula FF3 found in column FC3 returns a successor event for every activity from column FC2. This formula ignores the first row where one is certain to find the ending event's incoming activity. Technical activity *st* has been added as the one to be found before all activities, and therefore, a starting event is assigned to this activity as the successor event. It is identified as the table E event, which does not appear in column ID_SE of table A. In the next step, formula FF4 found in table FT1 identifies the activities for which the event in column FC3 is the predecessor event. Such activities would surely not be identified in the first row, which is why it is ignored. Therefore, each row of auxiliary table FT1 contains successors of the corresponding activity from column FC2. Formula FF5 found in column FC4 returns the number of successors for every activity. For each corresponding activity from table FT1, formula FF6, found in auxiliary table FT2, determines how many times the former appears in target table FT3 in a specific row, in consecutive columns, with an identical sequence of activities appearing previously in the given path. This is determined on the basis of the values calculated in advance, found in the previous rows of auxiliary column FC5. Found in auxiliary column FC5, formula FF7 sums up the values retrieved from the current row of auxiliary table FT2 to determine how many times the corresponding activity from auxiliary column FC2 appears in consecutive columns, in a specific row of target table FT3. This formula ignores the first row where the value of 1 will surely always be found.

	R	S	T	AW	AX	AY	CB	CC	CD	DG	DH	DI	EL	EM	EN	FQ	FR	FS	GV	GW	GX
1	FF8 FT3	FF9		FF10 FT4			FF11 FT5			FF12 FT6			FF13 FT7			FT8	FF14		FF15 FT9		
2	st	st		st-	st-		st-	st-		1	2		2	2					1	1	0
3	a	a		a-	a-		st-a-	st-a-		1	2		2	2		1	1	1	2	2	0
4	d	b		d-	b-		st-a-d-	st-a-b-		1	1		1	1		1	2	1	1	0	0
5	c			c-			st-a-d-c-			1			1			1	1	1	0	0	0
6																1	1	2	0	0	0

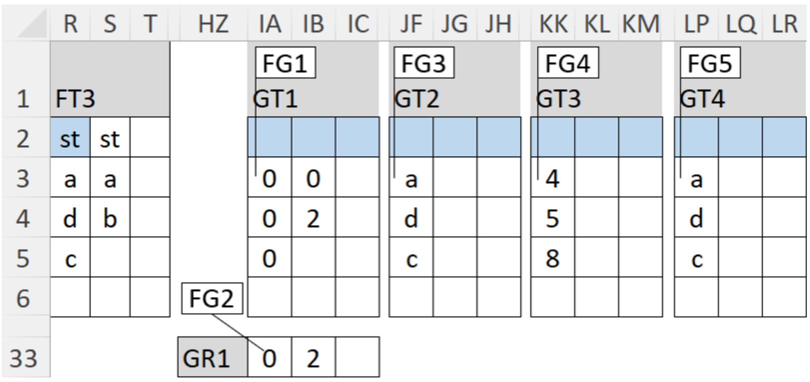
FF8	R3:R31	=IF(GV2>0;INDEX($E$2:$H$31;MATCH(R2;$C$2:$C$31;0);FQ3);"")
FF9	S2:AU2	=IF(AND(R2<>"";DG2<EL2);$R$2;"")
FF10	AW2:BZ31	=IF(R2<>"";CONCAT(R2;"-");"")
FF11	CB2:DE31	=IF(AW2<>"";CONCAT(AW$2:AW2);"")
FF12	DG2:EJ31	=IF(CB2<>"";COUNTIF($CB2:CB2;CB2);"")
FF13	EL2:FO31	=IF(R2<>"";INDEX($P$2:$P$31;MATCH(R2;$C$2:$C$31;0);1);"")
FF14	FQ3:GT31	=IF(DG3<EL3;FQ3;IF(CB2=CC2;FQ3+1;1))
FF15	GV2:HY31	=IF(R2<>"";INDEX($I$2:$I$31;MATCH(R2;$C$2:$C$31;0);1);0)

Figure 13. Auxiliary columns and tables in worksheet CP – part 2.

Source: own work.

Auxiliary table FT3 is the target table where all the potential paths are generated. Each of the paths identified there starts with technical activity *st*, which means that *st* will be the only value appearing in the first row. There must definitely be at least one path, and therefore, the value *st* is permanently entered in the first left cell of the first row. Formula FF9 found in the consecutive cells of this row uses the values taken from tables FT6 and FT7 to enter the value *st* in the consecutive cells as many times as there are paths identified. Formula FF8 found in the remaining cells of the table returns the corresponding activities using the pre-determined values retrieved from auxiliary tables FT1, FT8, and FT9. The activity being inserted is selected from the row in table FT1 which contains successors for the activity which is to be found in the cell above in table FT3. The number of the column from which the activity being inserted is taken is determined in table FT8. For each activity in table FT3, formulas FF10 and FF11, found in tables FT4 and FT5, create a code consisting of hyphenated activities that form the path in which one can find the given occurrence of the activity. The code contains activities from *st* to the current activity. Formula FF12 found in table FT6 counts how many times the corresponding activity from table FT3 has appeared with an identical sequence of activities occurring earlier in the given path. Found in table FT7, formula FF13 determines how many times the given activity should appear with an identical sequence of activities occurring earlier in the given path. Based on the values retrieved from auxiliary tables FT5, FT6, and FT7, formula FF14 found in table FT8 determines the number of the column from which the activity to be inserted into the corresponding cell of table FT3 should be retrieved. In this table, the formula in question appears neither in the first row nor in the first column. In the first identified path in table FT3, all activities are always retrieved from the first column, and therefore, the values of 1 are permanently entered in the first column of table FT8. In table FT3, formula FF8 enters consecutive activities only if the activity in the cell above has successors. In order to verify whether this condition has been met, formula FF15 in table FT9 determines the number of successors for the corresponding activities from table FT3.

It is necessary to isolate the critical paths from all the paths identified. As part of this step, four tables and one auxiliary row have been added in worksheet CP. Their layout along with the formulas applied have been shown in Figure 14.



FG1	IA3:JD31	=IF(CP!R3<>"";INDEX(A!$H$2:$H$31;MATCH(CP!R3;A!$A$2:$A$31;0);1);"")
FG2	IA33:JD33	=IF(IA3<>"";SUM(IA3:IA31);"")
FG3	JF3:JF31; JF3:KI31	=FILTER(R3:AU3;$IA$33:$JD$33=0;"")
FG4	KK3:LN31	=IF(JF3<>"";INDEX(A!$B$2:$B$31;MATCH(CP!JF3;A!$A$2:$A$31;0);1);"")
FG5	LP3:MS3; LP3:MS31	=IF(JF3<>"";FILTER(JF3:JF31;KK3:KK31>0;"");"")

Figure 14. Auxiliary columns and tables in worksheet CP – part 3.

Source: own work.

In all auxiliary tables, the formulas have been entered from the second row onwards. In this way, the technical activity has been removed from the paths identified. Found in table GT1, formula FG1 returns the float for each corresponding activity from table FT3. These floats are summed up for each path using formula FG2 in auxiliary row GR1. Based on the values retrieved from this row, formula FG3 transfers to table GT2 only those paths whose sum of floats equals 0, i.e. the critical paths. At this stage, these paths may contain dummy activities. In order to remove them, in auxiliary table GT3, formula FG4 returns the duration for each corresponding activity from table GT2. Formula FG5 transfers to table GT4 only those activities whose duration value is greater than 0. By that means, the final list of all the identified critical paths is generated in table GT4.

5. Discussion

The tool proposed by the author is aligned with the MiRel concept, according to which all data should be collected within a structure consistent with the relational data model, while data processing should be performed automatically using mechanisms created using built-in spreadsheet functions.

Assuming that a project subject to analysis is described using a directed graph consistent with the Activity on Arrow approach, such a graph can be defined within a relatively simple table structure, as proposed in the paper. One is completely unrestricted in terms of the sequence in which activities and events are entered in tables. Similarly unconstrained is the sequence of their occurrence in the graph being defined. This makes it easy to create a new graph as well as to expand and modify an existing one. Based on a graph defined in tables, the values envisaged for the CPM method, including the earliest and latest occurrence times for events as well as floats for both events and activities, can be calculated fully automatically. It is also possible to fully automate the identification of the critical paths existing in the project analysed. Both the calculations being performed and the critical path identification require a specific sequence of events and activities. This sequence can be established with reference to the proposed coefficient of sequence. It can be calculated using matrix calculus in a manner similar to how it is applied in the sphere of cost management.

The critical path identification process is broken down into two steps. In the first one, all paths from the starting event to the ending event are identified and then extended by the technical activity *st* placed at the beginning. In step two, the critical paths are selected from among all paths, and the technical activity at the beginning as well as all the dummy activities are removed from these paths. The first step has proved to be particularly difficult to perform. However, the author has successfully developed a mechanism capable of identifying all paths existing in the graph in a fully automatic manner. As assumed under the MiRel method, all the data processing actions the tool entails are performed by mechanisms which use built-in spreadsheet functions. The set of the functions applied is relatively limited, including those available in Excel, such as: IF, TRANSPOSE, AND, INDEX, MATCH, SUM, MMULT, MINVERSE, ROW, COLUMN, COUNTIF, MAX, OR, SORTBY, FILTER, MIN, COLUMNS, COUNTBLANK, CONCAT.

6. Conclusions

The solution described in this article clearly demonstrates that a spreadsheet provides a fully sufficient environment to create a tool supporting the CPM method. The solution presented in the paper can be successfully applied in practice to support the CPM method. As proposed, the concept of a mechanism enabling all paths to be identified in a directed graph can be employed in any other tool where path identification is necessary. The fact that one is not forced to use any code written in any programming language supports the argument that solutions such as the one proposed by the author can be successfully developed and extended by persons who are familiar with spreadsheets but have no command of programming.

The presented solution is limited by the fact that it was developed using the functions and mechanisms available in the desktop version of MS Excel, and its proper operation can only be guaranteed in this environment. Confirming its applicability to other popular spreadsheets, both desktop and online, requires further in-depth research.

Acknowledgements

The research presented in the article was supported by statutory work BK-257/ROZ1/2025 (13/010/BK_25/0087).

References

1. Aja, R.O., Chukwu, W.I.E. (2017). Application of Project Evaluation, Review Technique and Critical Path Method (PERT-CPM) Model in Monitoring Building Construction. *World Applied Sciences Journal*, *35(11)*, pp.2401-2413, doi: 10.5829/idosi.wasj.2017.2401.2413
2. Allen, S. (2006). *Modelowanie danych*. Gliwice: Helion.
3. Barreto, H. (2015). Why excel? *The Journal of Economic Education, 46(3)*, pp. 300-309.
4. Cocking, S. (2017). Seven Reasons why Excel is still used by half a billion people worldwide. *Irish Tech News*. Retrieved from: https://irishtechnews.ie/seven-reasons-why-excel-is-still-used-by-half-a-billion-people-worldwide/, 23.06.2025.
5. *Guidelines and examples of array formulas*. Retrieved from: https://support.microsoft.com /en-us/office/guidelines-and-examples-of-array-formulas-7d94a64e-3ff3-4686-9372-ecfd5caa57c7, 10.06.2025.

6. Jackson, M., Staunton, M. (2004). *Zaawansowane modele finansowe z wykorzystaniem Excela i VBA*. Gliwice: Helion.
7. Kerzner, H. (2009), *Project Management. A System Approach to Plannin, Scheduling and Controlling. Tenth Edition*. Hoboken, New Jersey: John Wiley & Sons, Inc.
8. Knight, G. (2006). *Analyzing Business Data with Excel*. Sebastopol: O'Reilly Media.
9. Kukuła, K., Jędrzejczyk, Z., Skrzypek, J., Walkosz, A. (2011). *Badania operacyjne w przykładach i zadaniach*. Warszawa: PWN.
10. Kusztelak, P. (2020). *Microsoft Excel w pracy finansisty. Analiza i modelowanie danych finansowych*. Warszawa: PWE.
11. Najihah, J., Hanyza, A. (2019). Usage of Microsoft Excel Spreadsheet as Accounting Tools in SME Company. *INWASCON Technology Magazine, Vol. I*, pp. 23-25, doi:10.26480/itechmag.01.2019.23.25.
12. Nowak, E. (1994). *Decyzyjne Rachunki Kosztów.* Warszawa: PWN.
13. Nowak, E. (2003). *Zaawansowana Rachunkowość Zarządcza*. Warszawa: PWE.
14. Próchnicki, W. (2012). *Zastosowanie Excela w pracy analityka finansowego, specjalisty ds. controllingu i analityka sprzedaży*. Gliwice: Helion.
15. Szczęśniak, B. (2017). Microtools Based on the Relational Data Model - representation of entities in a spreadsheet. *17th International Multidisciplinary Scientific GeoConference SGEM 2017, 17(21).* Albena, Bulgaria, pp. 447-454.
16. Szczęśniak, B. (2018a). Microtools Based On The Relational Data Model - transition from type II to type I of entity representation. *18th International Multidisciplinary Scientific GeoConference SGEM 2018, 18(2.1)*. Albena, Bulgaria, pp. 553-560.
17. Szczęśniak, B. (2018b). Microtools based on the relational data model - conformity between foreign and primary keys. *XV International Conference Multidisciplinary Aspects of Production Engineering (MAPE), 1(1)*. Zawiercie, Poland, pp. 491-497.
18. Szczęśniak, B. (2020). A spreadsheet to support the Suzuki ABCD method. *Scientific Papers Of Silesian University Of Technology, Organization And Management Series, Iss. 142*, pp. 205-216.
19. Szczęśniak, B. (2020a). Microtools based on the relational data model for improvement of information processes - multiplicity of relationships, and mandatory nature of relationships and attributes. In: K.S. Soliman (Ed.), *Education Excellence and Innovation Management: a 2025 Vision to Sustain Economic Development during Global Challenges. Proceedings of the 35th International Business Information Management Association Conference (IBIMA)* (pp. 10537-10544). Seville, Spain.
20. Szczęśniak, B. (2021). MiRel concept-conforming tool for supporting service quality measurement by the SERVQUAL method. In: K.S. Soliman (Ed.), *Innovation management and information technology impact on global economy in the era of pandemic. Proceedings of the 37th International Business Information Management Association Conference (IBIMA)* (pp. 12207-12218). Cordoba, Spain.

21. Szczęśniak, B. (2022). Online forms and spreadsheet as means to support quality measurement by the SERVQUAL method. *Scientific Papers of Silesian University of Technology - Organization and Management Series, Iss. 161*. https://doi.org/10.29119/1641-3466.2022.161.19
22. Szczęśniak, B. (2023a). Online forms and spreadsheet as means to support the ABCD method. *Scientific Papers of Silesian University of Technology - Organization and Management Series, Iss. 169*. https://doi.org/10.29119/1641-3466.2023.169.40
23. Szczęśniak, B. (2023b). MiRel concept-aligned tool for analysing employee participation in projects. *Scientific Papers of Silesian University of Technology - Organization and Management Series, Iss. 185*. https://doi.org/10.29119/1641-3466.2023.185.25
24. Szczęśniak, B. (2024). MiRel concept-aligned tool for analysing employee participation in projects – data validity in the tool. *Scientific Papers of Silesian University of Technology - Organization and Management Series, Iss. 197*. https://doi.org/10.29119/1641-3466.2024.197.30
25. Szczęśniak, B., Molenda, M. (2013). Spreadsheet application supporting the x-r control chart. In: *Conference Proceedings of the 22th Conference Modern Mathematical Methods in Engineering (3mi)*, (pp. 128-134). Horni Lomna, Czech Republic.
26. Winston, W.L. (2019). *Analiza marketingowa. Praktyczne techniki z wykorzystaniem analizy danych i narzędzi Excela*. Gliwice: Helion.
27. Zasadzień, M., Szczesniak, B., Skotnicka-Zasadzień, B. (2017). Implementation of maintenance employees' work time scheduling. In: P. Limpaphayom, G. Huang (Eds), *Proceedings of the Second International Conference on Economic and Business Management*, (pp. 226-231). Shanghai: Atlantis Press.