International Journal of Artificial Intelligence and Neural Networks– IJAINN
Volume 5 : Issue 1 [ISSN : 2250-3749]
Publication Date: 30 April, 2015

C# method for all Types of Nodes in Fuzzy GERT

[Roland Lachmayer, Mahtab Afsari, Rashid Hassani]

***Abstract*—The advancement of human communities depends on the quality of research. Research projects are usually cost as well as time consuming. Project scheduling is one of the integral parts of project management. Results achievement in short period is considered as an important factor. This paper presents new methodology to solve Fuzzy GERT networks for all kind of nodes. In this method the GERT network is simulated and scheduled by combination of C# and Matlab which leads to solve functions and equations faster than previous manually methods.**

***Keywords*— Research Projects, Fuzzy GERT, α-cuts, Scheduling, C#**

I. Introduction

The history of using fuzzy logic in project control dates back to the 70s decade. Chanas and Kamburowski [18] introduced a new method titled FPERT. This method estimates the duration of activities and project completion time using fuzzy logic. Kaufman and Gupta [19] presented a method based on taking the uncertainty provision for activity timing into account and considering the conditions in which experts extensively disagree. In this method, the fuzzy CPM has been solved once using optimistic data and once using pessimistic data. To find the fuzzy path, fuzzy subtraction is used. Kufmann et al. and Alfered et al. [22]. Margaret Shipley et al. [19] presented an algorithm named BIFPET which computed the expected time using optimistic, most likely and pessimistic times. Chang [5], [6] in his first article introduced the fuzzy GERT method for solving the reliability problem for series systems. In his second article, he presented the capability of repairable systems using fuzzy GERT method. The Chang method has investigated the issue using the Exclusive-OR and cannot be used in research projects which include different types of node. The Itakura and Nishikawa[7] method has also some deficiencies as follows: Time is considered as discrete fuzzy numbers which is impossible in reality. Algorithm is determined based on parts, paths and loops which is very difficult in extended networks. Other deficiencies include:

- In this method, the possibility of occurrence has been considered the same for all the loops which does not happen in reality.
- Activities on the loop have been assumed certain in recursions which is not always possible in reality.
- Only fuzzy has been used in the GERT network. It is better to use a combination of fuzzy and probability.
- The membership degree of activity branch or the occurrence possibility of branch has been expressed as a certain number. In their method, the number of activities outside each node belongs to a fuzzy network [22], [7]. The solution is alike probabilistic GERT method except that min, max functions have been used for fuzzy networks [4]. Karimi Gavareshki et al. presented new Fuzzy GERT method [1], [2]. In this method they used Fuzzy Triangular numbers to show time of activities. They compared 3 methods of Fuzzy GERT, GERT and CPM to show application and suitability of Fuzzy GERT for scheduling research projects [1], [16], [21].

A. *Research project*

Research projects are usually cost and time consuming and do not end in result in short term. Project scheduling is one of the integral parts of project management. Project Management includes management of costs, time, quality, resources and materials procurement. Research projects have some particularities including [1],[2],[9]:

- Uncertainty in defining the activities
- Ambiguous and uncertain results of researches
- Uncertainty in periods of activities
- Non-repetitiveness of project activities
- Loops and recurrences.

Due to such problems, specific techniques are required for scheduling research projects. Certain techniques do not have the capability in modeling uncertain activities and duration of activities and nodes [10]. So, it is obvious that employing these techniques will increase the error percentage in scheduling research projects.

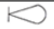
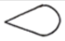
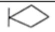
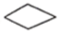
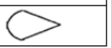
GERT networks are widely used in solving existence and reliability problems [23]. In GERT networks, two parameters are related to activity branches, namely activity period and occurrence probabilities are expressed in terms of probable distribution. it seems that fuzzy GERT networks are the same as probable GERT networks, but fuzzy parameters have been

Roland Lachmayer, Mahtab Afsari
Leibniz University of Hannover
Germany

Rashid Hassani
University of Rostock
Germany

replaced by probable parameters [9], [12].There are 6 type nodes in GERT networks that includes 3 inputs and 2 outputs. These nodes are shown in Table I.

TABLE I. combining 3 input and 2 output relations

input output	Exclusive-Or	Inclusive-Or	AND
Deterministic			
Probabilistic			

B. *Why we create this tool?*

There is not any tool to draw GERT networks, except a simulator that is produce in 1970. This simulator developed through FORTRAN language [9]. There is not any newer tool for GERT analysis, simulation and draw. So we decide to create a tool to draw GERT diagram as first step. We have tried to produce a user friendly tool that is developed through C#.

As you know today, more researches we need to do since 1970. These research activities are time and cost consuming. Using computer program help us to control and manage research teams better. We feel this requirement and develop software in this direction.

II. Application software

A. *C#*

In June 2000, Microsoft introduced the C# programming language (pronounced “C-sharp”). It was designed to improve upon the C++ language developed in the early 1980s at AT&T Labs [13]. Because it is a totally new language, it has no backward compatibility issues, as C++ had with its predecessor, the C language. C# also shares many features with Java, a programming language released in 1996 by Sun Microsystems. It is impossible to discuss C# without discussing the Microsoft .NET Framework that supports C# and many other programming languages. The .NET Framework is essentially a giant collection of tools for software development. It was designed to allow traditional text-based applications, GUI applications, and Web-based programs to all be built with equal ease. Currently, over 60 languages are supported by the .NET Framework. That means applications written in any of these languages have access to the tools provided within the .NET Framework. Applications can even be written that mix and match modules in various languages. In April 2003, only three years after the first release of C# and .NET, C# and the CLI (Common Language Infrastructure—a significant subset of the .NET tools) were adopted as ISO (International Organization for Standardization) standards. C# continues to grow in popularity [14], [15].

For the first 30 to 35 years of software development, text was virtually the only method of displaying results in human-readable form, and in those early days it was quite common for programs to produce huge stacks of alphanumeric output [15]. These days an alternative form of output— graphics— has become much more widely used. With graphics, we are no longer limited to 100 or so printable characters; instead, programmers are free to construct whatever shapes and images they desire. C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create traditional windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. We used C# in our algorithm to solve fuzzy GERT because it was easy in graphically part and for drawing GERT networks that have complexity in structure, it is better to use C# for solving the GERT networks.

B. *Graphic software*

In my developed software, a user must enter node information such as fuzzy time, loops start and end nodes, paths and multiple inputs node. Location of each node in the computer display is another problem for a user. In this software tried, a user draw nodes and paths and relationships of them easily. After doing changes through user to the GERT network, it is a complete model of a project. It means you can change nodes information, location and relationships. At last you can save this GERT network in a file in your computer. If a model was large with many nodes and paths and loop, you can enter this model in many days. Each time you can enter some parts of model and save it, and then in another time you can enter other parts to complete your model.

To control the setting and clearing of pixels, programmers use a collection of functions that are part of a special software package called a graphics library. Virtually all modern programming languages, including C#, come with an extensive and powerful graphics library for creating different shapes and images. Typically, an “industrial strength” graphics library includes hundreds of functions for everything from drawing simple geometric shapes like lines and circles, to creating and selecting colors, to more complex operations such as displaying scrolling windows, pull-down menus, and buttons. We restrict our discussion to a modest set of functions. Although the set is unrealistically small, it will still give you a good idea of what visual programming is like, and enable you to produce some interesting, nontrivial images on the screen. Graphical drawing in the C# and .NET environment is supported by a framework called GDI. GDI stands for Graphical Device Interface. To be able to draw on the screen, it is necessary to set up a “window” to hold the drawing canvas [14], [15], [11], [3], [17].

III. Methodology

We need to develop two softwares. One to draw a GERT network and save network data in a file and another to evaluate nodes, paths, loops then generate outputs and results. We use C# for developing first software. Matlab is used to develop the second part of our software. Generated outputs and results of first software which is C# software are inputs of second software which is Matlab software. In both parts of our software we must consider the following assumptions [16],[21].

- Duration time of activities is represented by triangular Fuzzy numbers.
- Repetitions number of loops is represented by triangular fuzzy numbers.
- Membership degree of activities and loops is a number between 0,1.
- Occurrence possibility of loops for different repetitions is equal.

We used fuzzy relations in our calculations. Followings are some used fuzzy equations and variables [1],[2]:

$$N = (a_1, b_1, c_1), M = (a_2, b_2, c_2)$$

$$N \oplus M = (a_1 + a_2, b_1 + b_2, c_1 + c_2)$$

$$\max(M, N) = \left[\vee (a_1, a_2), \vee (b_1, b_2), \vee (c_1 c_2)\right]$$

$$\min(M, N) = \left[\wedge (a_1, a_2), \wedge (b_1, b_2), \wedge (c_1, c_2)\right]$$

Y_i = ith node

X_{i-j} = activity (i-j)

μ_{i-j} =membership degree of activity i-j

$\tilde{t}_{i-j}$ = fuzzy duration time of activity i-j

L_{ni} = loop n to i

$\mu_{L_{ni}}$ = membership degree of loop

$\tilde{r}_{L_{ni}}$ =fuzzy repetition number

P_i = precedence nodes (activities)

$\tilde{ft}_{i-j}$ = fuzzy ending time of activity i-j

$\mu_{ft_{i-j}}$ = membership degree of ending time

$S\tilde{T}_i$ = initial release time of node

$M\tilde{T}_i$ = average time of node

μ_{MT_i} = membership degree of node

IV. Algorithm of Fuzzy GERT for nodes

1) For start node set

$$M\tilde{T}_{start} = S\tilde{T}_{start} = (0,0,0)$$

2) For each node from start to end compute average time of node (MT_i)

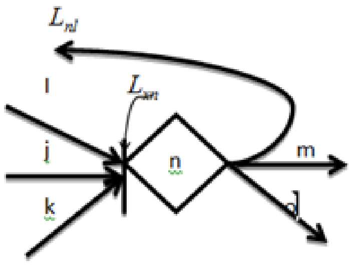


Figure 1. Exclusive-OR node

2.a) Computing ending time and membership degree of precedence activities (for input activities to node):

$$\tilde{ft}_{i-n} = M\tilde{T}_i \oplus \tilde{t}_{i-n} \rightarrow \forall i \in p_n \qquad \mu_{ft_{i-n}} = \min(\mu_{i-n}, \mu_{MT_i}) \forall i \in p_n$$

2b) Computing initial release time of node $\left(S\tilde{T}i\right)$

Initial release time of node is calculated based on kind of input side.

A) Exclusive-OR node

Normalization of membership degree:

$$\mu'_{ft_{i-n}} = \frac{\mu_{ft_{i-n}}}{\sum_{i \in p} \mu_{ft_{i-n}}} \rightarrow \forall i \in p_n$$

$$S\tilde{T}_n = \sum_{i \in p_n} \mu'_{ft_{i-n}} \otimes \tilde{ft}_{i-n}$$

$$\mu_{MT} = \max\{\mu_{ft_{i-n}}\}$$

B) Inclusive- OR node

$$S\tilde{T}_n = \tilde{\max}\{\tilde{ft}_{i-n}\}$$

$$\mu_{MT} = \min\{\mu_{ft_{i-n}}\}$$

C) AND node

$$S\tilde{T}_n = \tilde{\min}\{\mu_{ft_{i-n}}\}$$

$$\mu_{MT} = \min\{\mu_{ft_{i-n}}\}$$

3) Computing average time of node ($M\tilde{T}_i$)

$$M\tilde{T}_n = \left(\sum_{L \in L_n} \mu_L \bullet \left[(\tilde{r}_L \otimes (S\tilde{T}_n \Theta m\tilde{t}_L)) \oplus S\tilde{T}\right]\right) \otimes \left((1 - \sum_{L \in L_n} \mu_L) \otimes S\tilde{T}_n\right)$$

4) Computing project completion time with evaluating network nodes from first node to end node, project network is scheduled and project completion time is obtained that is equal with average time of end node of project network.

$$T_{Proj} = M\tilde{T}_{end}$$

V. Creating and running C# program

Creating and running a C# program is basically a three-step process.

The first step is to type the program into a text editor. When you are finished, you save the file, giving it a name with the extension .cs.

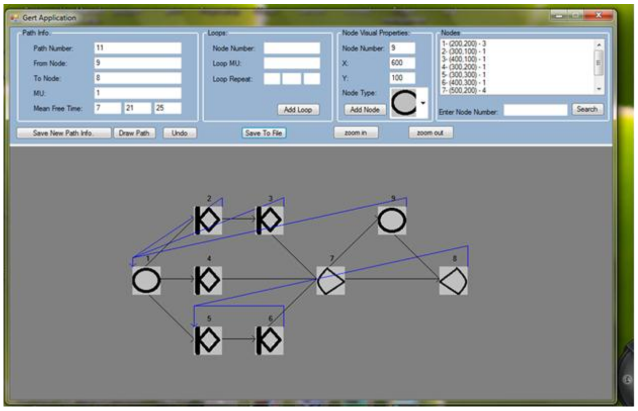
On the second step, the program must be compiled using a C# compiler for your computer, and the resulting object code linked with any C# library object code.

The third step loads and executes the program file; you may have to type operating system commands for the last two steps. The instructions in C#, or indeed in any high-level language, are designed as components for algorithmic problem solving, rather than as one-to-one translations of the underlying machine language instruction set of the computer. Thus they allow the programmer to work at a higher level of abstraction.

We used MS-Visual Studio 2012 integrated development environment (IDE) to type and edit my codes, compile and run my program. It is suitable tool to create and run C# program.

VI. Description of defined methodology

In our software there are many command buttons as you can see in figure 2 which is the screen shot of software.



Path Info.:

Path Number: 11

From Node: 9

To Node: 8

MU: 1

Mean Free Time: 7 21 25

Loops:

Node Number:

Loop MU:

Loop Repeat:

Node Visual Properties:

Node Number: 9

X: 600

Y: 100

Node Type:

Nodes:

1- (200,200) - 3
2- (300,100) - 1
3- (400,100) - 1
4- (300,200) - 1
5- (300,300) - 1
6- (400,300) - 1
7- (500,200) - 4

Enter Node Number:

Figure 2. Example screenshot of this software

We have tried to describe each command button action:

1. “*Add Node*” button action

Before click on “Add Node” button you must enter node number into a textbox, node position on the screen (X and Y) which you can see in the node visual properties section of screenshot, then select node type through combo box.

By pressing or clicking on “*Add Node*” this operation starts. Node information is read and checked so that the information is not wrong and is in the limits. After checking the input information, if there are not mistakes or deficiencies, this information is added to the output array then a message is shown to the user to be informed that the information is inserted. Otherwise, the error message relevant to the deficiency is shown to the user.

2. “*Save New Path Info*” button action

On the left part of the form there is a place for inserting information of each path (See Screen Shot). Information such as number of path, number of the start and finish node, µ, and t1, t2, t3 are given by the user. Then information is checked to be complete and µ be in the limits. If there is any error, an error message suitable with the user's mistake is shown otherwise the path information is saved in an array and a positive correct message is sent to the user.

3. “*Add Loop*” button action

In the central part of the form and page designed by C# (See Screen Shot), there is a place to insert input information of each loop. Information such as destination node number of the loop, µ of the loop and t1, t2, t3 of the loop that is entered by the user. If the information is wrong or is not in the limits, it is checked and an error message relevant to the mistake is sent. Otherwise, the message of saving the information is shown and the information of the loop or loops of each path is saved. Here, you may have several loops for a path and the loops are inserted one by one from the most internal to the most external. This means that information of each loop is inserted separately.

4. Draw GRET diagram in the form method

By pressing “*Draw Path*” button the GRET starts to draw in the bottom of the form. Showing GRET may start before final saving of the information inserted in the model to ensure the correctness of the input model. To draw GRET, information of each path and loops of them is read one by one. According to the type of start and finish nodes, appropriate shape of and is drawn on the place of showing GRET in the determined coordinate for this node. Then, the connecting line of these two nodes is drawn. After that information of the loops is analyzed and loop lines are drawn. The array of the information of the paths is checked. If there is no path remained, this algorithm is finished and otherwise, we do the same set operations to fulfill the process.

5. “*Save To File*” button action

By pressing “Save To File” button, this operation starts. A file opens to save the information. For the first path, (0, 0, 0, 0) is saved in this file. After arrangement of the information of each path, a path and the first path is read and its information and its loops are saved on the file. If all the loops of the path are saved, it moves to the next path. If there is a path, the process explained above starts and is repeated and if no path is remained, a message is sent to the user that says saving the information is finished and the desired file has been closed successfully.

VII. Some flowcharts of this software

We used C# to draw GERT networks. You can see the following flowcharts that we used to write program in C#.

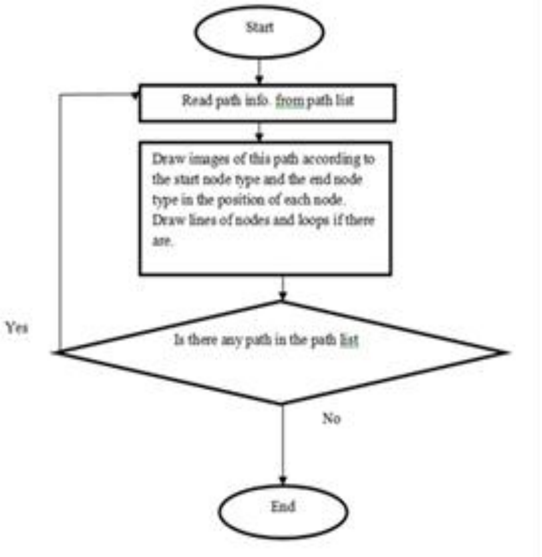

Start
Read path info. from path list
Draw images of this path according to the start node type and the end node type in the position of each node.
Draw lines of nodes and loops if there are.
Is there any path in the path list
Yes
No
End

Figure 3. Draw GERT diagram in the form (splitContainer1_Panel2_Paint)

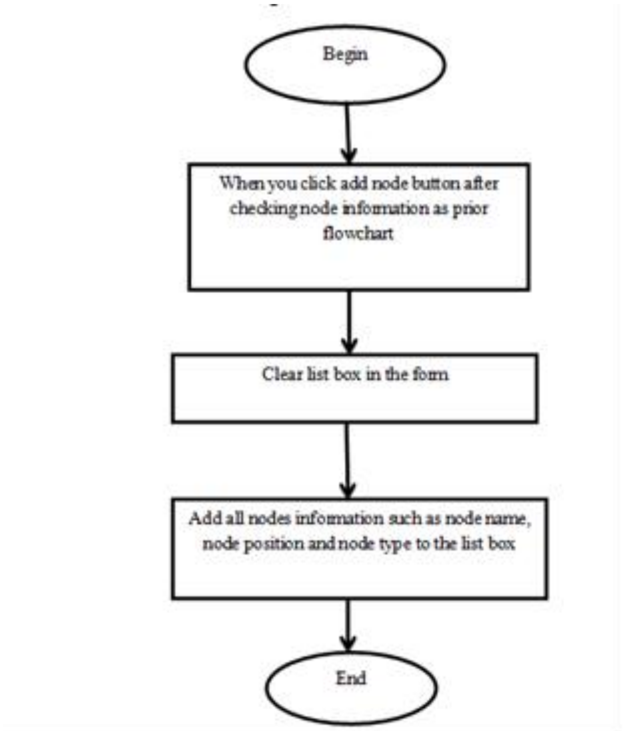

Begin
When you click add node button after checking node information as prior flowchart
Clear list box in the form
Add all nodes information such as node name, node position and node type to the list box
End

Figure 4. List Box Flowchart

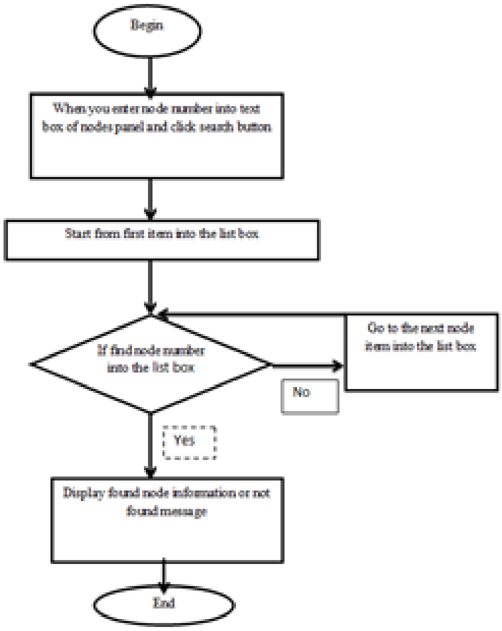

Begin
When you enter node number into text box of nodes panel and click search button
Start from first item into the list box
If find node number into the list box
Go to the next node item into the list box
No
Yes
Display found node information or not found message
End

Figure 5. Search Button

We made image list for adding type nodes. In this image list we put shapes of GERT nodes in it.

For solving the GERT network with C# we should enter some inputs. These inputs are:

- Node numbers
- X, Y pixels in quarter coordinator
- Number of node that loop enter to it
- Membership degree of nodes
- Membership degree of loops
- Triangular number for fuzzy time of activities

After entering inputs and saving them, we will have data as output that will save in Debug that will be numbers

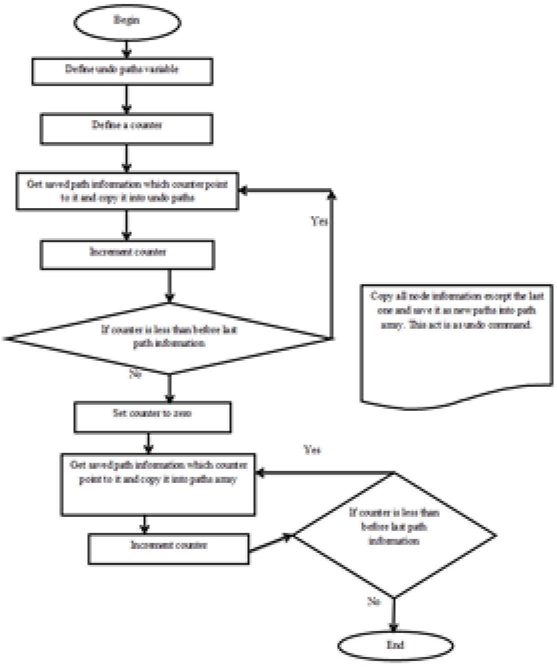

Begin
Define undo paths variable
Define a counter
Get saved path information which counter point to it and copy it into undo paths
Increment counter
If counter is less than before last path information
Yes
No
Copy all node information except the last one and save it as new paths into path array. This act is as undo command.
Set counter to zero
Get saved path information which counter point to it and copy it into paths array
Increment counter
If counter is less than before last path information
Yes
No
End

Figure 6. Undo Button

These numbers have the following meaning:

Path number, from node, type node (from node), to node, type node(to node), times, mu, loop (from node), loop (to node) , repeat of loops, mu of loop, loop (from node) , loop (to node) , repeat of loops , mu of loop.

VIII. Example of screenshot of this software

To better explanation, we put an example screen shot of this software as shown in figure 2. As you can see, you must enter nodes information first. Then enter each path information and loops of this path. Then after entering each path you can draw the GERT network to confirm or undo this path. If your GERT network is large, you can search for a node. Also you can zoom in or zoom out to visit your model better. Finally you can save your model information into a file. This file is comma separated file, which you can save each path and its loop information in a line. The information is separated from each other via comma. You can save large model in many times. This output file acts as input file for node evaluation and program analysis for Matlab application.

IX. Conclusion

Fuzzy GERT networks are the same as probable GERT networks in which fuzzy parameters have been replaced for probable parameters. These networks provide several features such as high capabilities to model research projects using logical nodes, fuzzy branches and cycles as well as simplifying the calculations and expressing uncertainty in the projects. Therefore, fuzzy GERT networks are suitable means for research project scheduling. So far, three different methods have been presented for fuzzy GERT networks.

In reality, estimating a single number for a possibility is difficult. The results from the three methods (fuzzy GERT, fuzzy CPM and certain CPM) indicate that the method of fuzzy GERT and Fuzzy CPM is closer to reality. In conclusion, it is recommended that in higher levels, the fuzzy GERT networks to be used because it considers loops and uncertainties that may be unimportant for project managers.

In this article, we presented new method of fuzzy GERT networks by using C# software to solve functions and equations faster than manually method. We have seen how the using of a high-level language overcomes many of the disadvantages of assembly language programming, creating a more comfortable and useful environment for the programmer. In a high-level language, the programmer need not manage the storage or movement of data values in memory. The programmer can think about the problem at a higher level, can use program instructions that are both more powerful and more natural language–like, and can write a program that is much more portable among various hardware platforms.

As a future work, we suggest the program that can solve all kinds of nodes in Matlab and C# .we suggest also methods that can show fuzzy inference systems in Matlab for all types of nodes in GERT networks. C# is not the only high-level language. You might be interested in looking at the other online modules for languages similar to C# (Java, Python, C++). Still other languages take quite a different approach to problem solving. We used C# to draw Graphical part of GERT networks because it is easier than other languages in Graphical solutions

References

[1] Karimi. G and M. H,” New Fuzzy GERT Method for research project scheduling’ IEEE International Engineering Management Conference, pp. 820-824, ISBN: 0-7803-8519-5, Singapore, 2004.

[2] Karimi. G, M.H,”New Technique in Fuzzy GERT for research project scheduling”, M.S. thesis, Dept. Electron. Eng, Iran Science and Technology Univ., Iran, 2001.

[3] Rashid Hassani, Amirreza Fazely, Riaz Choudhury and Peter Luksch, “Analysis of Sparse Matrix-Vector Multiplication Using Iterative Method in CUDA”, IEEE NAS 2013, IEEE computer society, ISBN: 978-0-7695-5034-3/13, pp. 262-266, DOI: 10.1109/NAS.2013.41, July 2013, Xiamen, China.

[4] Juite. A, ”Fuzzy project scheduling approach to minimize schedule risk for product development”, Journal of Fuzzy Sets and Systems, Special issue: Optimization and decision support systems. 2002.

[5] Chang, C. H, ”Fuzzy repairable reliability based on Fuzzy GERT Microelectronic reliability”,36: NO.10 , pp. 1557-1563, 1996.

[6] Chang, C. H, ”Fuzzy consecutive K-Out-Of-N Microelectronic reliability”, NO.12, pp. 1909-1922, 1994.

[7] Itakura. H and Nishikawa. Y, “Fuzzy Network technique for technical forecasting”, Fuzzy set and systems, pp. 99-131, 1984.

[8] Pritsker. A, “GERT:Graphical Evaluation and Review Technique”, Rand Corporation , NASA RM, pp. 44-73, 1996.

[9] Pasit Lorterapong and Osma Moslehi, “Project Network analysis using Fuzzy set Theory”, Journal of Construction Engineering and Management, pp. 308-318, DOI: 10.1061/ASCE0733-9364122:4308, 1996.

[10] Willy. H and Roel. L,”project scheduling under uncertainty: survey and research potentials” European journal of operational research, pp. 289-306, 2005.

[11] Rashid Hassani, Amirreza Fazely and Peter Luksch, “High Performance Concurrent Multi-Path Communication for MPI”, The 19th European MPI conference (EuroPVM/MPI 2012), Springer LNCS 7490, pp. 285-286, DOI: 10.1007/978-3-642-33518-1_34, Sep 2012, Vienna, Austria.

[12] Dwiri. T and Kablan. M, ”Using fuzzy decision making for the evaluation of the project management internal efficiency”, Journal of Decision Support Systems, pp. 712-726, 2006.

[13] “Introduction to the C# Language and the .NET Framework”, available at “http://msdn.microsoft.com/en-us/library/z1zx9t92.aspx”, accessed on 10/07/2014.

[14] Online module to accompany Invitation to Computer Science, 6th Edition, ISBN-10:1133190820, ISBN-13:9781133190820, Cengage Learning, 2013.

[15] Lachmayer. R and Afsari. M, “Algorithm of Converting Fuzzy GERT into CPM for Research Project Scheduling With Case Study”, 8th International Conference of Project Management, 2012, Tehran, Iran.

[16] Lachmayer. R and Afsari. M, “Fuzzy GERT method for Scheduling Research Projects”, 9th International Conference of Industrial Engineering, 2013, Tehran, Iran.

[17] Rashid Hassani, Md. Aiatullah and Peter Luksch, “Improving HPC Application Performance in Public Cloud”, FIE 2014, IERI Procedia by ELSEVIER, ISSN: 2212-6678, DOI: 10.1016/j.ieri.2014.09.072, July 2014, Beijing, China.

[18] Margaret. F, and Khursheed Omer, “A fuzzy logic approach for determining expected values”, Journal of the operational research society, 1996, 47: 562-569.

[19] Kaufmann. A and Gupta. M, “Fuzzy mathematical models in engineering and management science”, ISBN-10: 0444705015, 1988.

[20] Rashid Hassani, Ganesh Chavan and Peter Luksch, “Optimization of Communication in MPI-Based Clusters”, CyberC 2014, IEEE computer society, Oct 2014, Shanghai, China.

[21] Chanas. S, Zielinski. P, “Critical path analysis in the network with fuzzy activity times”, Journal of Fuzzy Sets and Systems. 2001, 122: 195-204.

[22] Didier. D, Fargier. H. and Fortemps. P, Fuzzy scheduling: “Modeling flexible constraints”, European Journal of Operational Research, 147: 231-252, 2003.

[23] Hong. Z, C. M.Tam and Heng. Li, “modeling uncertain activity duration by fuzzy number and discrete-event simulation”, European journal of operational research, No: 164, pp. 715-729, 2005.

About Author (s):

Prof. Dr.-Ing. Roland Lachmayer is head of the institute of Product Development at Leibniz University of Hannover.

Mahtab Afsari has been graduated Master degree of Industrial Engineering from Isfahan University of Technology, Iran. She is PhD researcher in the Product Development Institute at Leibniz University of Technology from 2012 until now. She is working on GERT Networks in Project Management and she is trying to make user friendly software for GERT Networks as a computer package to encourage project managers to use GERT Networks as the best of networks that has appropriate benefits more that CPM and PERT Networks.

Rashid Hassani received his BE degree in Information Technology from VTU, India. He graduated with M.Sc. degree in Computer System Engineering emphasized in Embedded and Cooperating Systems from Halmstad University, Sweden. He is currently a research assistant and academic teacher in the department of computer science at Rostock University, Germany. His research interests include parallel and high performance computing, Cloud computing and networking protocols.