

Fuzzy Model for Optimizing Strategic Decisions using Matlab

Amandeep Kaur¹, Vinay Chopra²

¹M.Tech Student, ²Assistant Professor,
DAV Institute of Engineering. & Technology, Jalandhar

Abstract:- Designing of Fuzzy model for optimizing the strategic decision can be very helpful to protect the failures in the system and software. The systems and software development industry is characterized by a paradigm of project failure. One of the known contributing causes of these project failures is poor requirements engineering and management, poor design strategy which has been repeatedly and widely issued and documented. The above problems could be easily countered by the Fuzzy logic, because fuzzy logic has ability to deal with uncertainty and multi valued logic This paper explains the technique for tracking the progress of the software project being built and the technique for selecting an optimal PERT chart developed by using fuzzy logic and it also explains the important variables that effects the strategic decisions connected with the software project

I. INTRODUCTION

The systems and software development industry is characterized by a paradigm of project failure (Standish 1995).the main causes of these project failures is poor requirements engineering and management, which has been repeatedly and widely discussed and documented for at least 10 years (Hooks 1993; Kasser and Schermerhorn 1994; Jacobs 1999; Carson.

Many of the Software development phases are highly communication-intensive activity that involves, at minimum, analysts, architects, developers, testers, business stakeholders, and end users. Among all the development phases and products, requirements are key ingredient in the process of designing and realizing any systems. Without clearly understanding requirements and their proper management, large projects are likely to fail and could have very high maintenance cost. So, the focus of every development methodology is on requirement engineering phase. Software development involves roughly 50 percent computing and 50 percent communication.

Project manager[3],[4] always need that the project should be completed on time or it should not be a big failure. So, manager the administrator constantly looks forward to those techniques or methods which help him in planning, scheduling, and controlling such activities. The concepts of network planning and critical path analysis have greatly assists him. We have made an attempt of developing a fuzzy expert system. The fuzzy expert system is used to select an optimal PERT chart on the basis of the constraints like cost, manpower, total minimum cost, etc. But in this paper we will use matlab to get the results.

II. FUZZY EXPERT SYSTEM

Fuzzy logic is a form of logic that applies to fuzzy variables. Fuzzy logic is non-monotonic, in the sense that if a new fuzzy fact is added to a database, this fact may contradict conclusions that were previously derived from the database. The concept of Fuzzy Logic (FL) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership. We here discuss the application of fuzzy logic to obtain the optimized PERT chart. In fuzzy logic, the fuzzy set, membership functions etc., are used and the optimized PERT specification is obtained. The inference we use in our fuzzy logic is said to be the Mamdani Inference. Expert systems are meant to solve real problems which normally would require specialized human expert. (Doctor or Mineralogist) Building an expert system requires extracting knowledge from expert (human).

Knowledge Engineer: - Knowledge Engineer can extract knowledge and can build expert system knowledge database.

Rule and Expert System: Expert system is based on IF - THEN rule work. Rule based systems can be

1. Goal driven using back ward chaining.
2. Data driven using forward chaining.

Expert system can use both. The user interacts with the system through UI which may use menus, Natural language or other style of information.[6] Inference engine used to reason with expert knowledge and data specific of problem. Case specific data includes both data provided by the user and partial conclusions, explanation system which allows the program to explain its reasoning to the user. Knowledge database editor helps their expert or knowledge engineer.

A) Steps in Fuzzy Expert System

- 1) Obtain information from one or more experts
- 2) Define the Fuzzy sets
- 3) Defining the Fuzzy rules.

B) Directed steps to use Fuzzy Expert System

- 1) Relate the observations to the Fuzzy sets
- 2) Evaluate case for all Fuzzy rules
- 3) Combine information from the rules.
- 4) Defuzzify the results.

C) Defining Fuzzy Set

Having [13] obtained the suitable information from the experts, the fuzzy set is defined.

- TMC - Total minimum cost.
- MP - Man Power.
- FP - Function points.

As well as defining the linguistic variables, there is need to give each one a range of possible values. Linguistic variables of TMC, MP, FP are high (h), low (l). To represent the fuzzy membership functions, the notation used is WAB where A is the variable (tmc, mp, fp) and B is the variable (h, l). For example, Wtmcl represents the function of total minimum cost low. Membership Functions: The membership functions we use here give the weights for each of the fuzzy sets. The representation of weights may be of the form Wab –where ‘a’ gives the fuzzy set and ‘b’ gives the linguistic variable of the fuzzy set.

D) Defining fuzzy rules

The second step in creating a fuzzy expert system is to define a set of fuzzy rules. These rules [19], unlike those used by traditional expert systems, are expressed in vague English terms and do not define cut-off or thresholds, but rather use subjective terms such as “high” and “low”. This map more naturally to the way an expert would express his or her knowledge and makes the process of converting that knowledge into rules far simpler and less prone to error. Our rules are defined as follows:

Rule1: If TMC is high
 AND MP is high
 OR FP is low
 THEN PCQ is high

Rule2: If TMC is high
 AND MP is low
 OR FP is high
 THEN PCQ is low

Rule3: If TMC is high
 AND MP is high
 AND FP is high
 OR PCQ is high

We now relate the observations to the fuzzy sets. We will examine three sets of data, for the three PERTs. The obtained related values are defuzzified by using defuzzication.

E) Defuzzication

We now need to Defuzzify the outputs to obtain a crisp PCQ to get the required result. The crisp values are obtained by calculating the Centroid for the PCQ using the formula- Centroid (c) = $(\sum Wp(x) * x) / (\sum Wp(x))$

III. FUZZY LOGIC IN PROJECT MANAGEMENT

One problem faced by managers who are using project management models is the elicitation of numerical inputs. Obtaining these with any degree of confidence early in a project is not always feasible. Related to this difficulty is the risk of precisely specified outputs from models leading to over commitment. These problems can be seen as the collective failure of software measurements to represent the inherent uncertainties in managers' knowledge of the development products, resources, and processes. It is proposed that fuzzy logic [9] techniques can help to overcome some of these difficulties by representing the imprecision in inputs and outputs, as well as providing a more expert knowledge based approach to model building. The use of fuzzy logic for project management however should not be the same throughout the development life cycle. Different level of available information and desired precision suggest that it can be used differently depending on the current phase, although a single model can be used for consistency.

One of the important benefits of fuzzy logic for software engineering project management is the flexibility available in terms of the types of input and output variables. Input variables can be expressed as simple fuzzy labels (a large number of entities in the data model), fuzzy numbers (about 250 entities), or using precise values (265 entities). Similarly, the output can be expressed in the same way, as a label (a short development time), fuzzy number (about 400 developer-hours), or precise values (378 developer hours).

Fuzzy Logic could be used in size estimation purpose. Size estimation can be done in advance to make better plans (to better size the job, to divide the job into separable elements), to assist in tracking progress (can judge when job scope changes, can better measure the work), to learn and build estimating skills. Estimation is an uncertain process no one knows how big the product will be.

IV. TECHNIQUES USED IN SOFTWARE PROJECT MANAGEMENT

The two most frequently used techniques in software project management forms of network planning are program evaluation and review technique (PERT) and critical path method (CPM). PERT or CPM has helped management is drastically reducing the project execution time.

V. METHODOLOGY USED

We have used MATLAB technology for computing the results. The Fuzzy Logic Toolbox extends the MATLAB technical computing environment with tools for the design of systems based on fuzzy logic. Graphical user interfaces (GUIs) guide you through the steps of fuzzy inference system[18] design. Functions are provided for many common fuzzy logic methods, including fuzzy clustering and adaptive neuro fuzzy learning.

The toolbox lets you model complex system behaviors using simple logic rules then implement these rules in a fuzzy inference system. You can use the toolbox as a stand-alone fuzzy inference engine. Alternatively, you can use fuzzy inference blocks in Simulink and simulate the fuzzy systems within a comprehensive model of the entire dynamic system. Like all MATLAB toolboxes, the Fuzzy Logic Toolbox can be customized. You can inspect algorithms, modify source code, and add your own membership functions or defuzzification techniques. Key Features are given as follows.

- Specialized GUIs for building fuzzy inference systems and viewing and analyzing results.
- Membership functions for creating fuzzy inference systems.
- Support for AND, OR, and NOT logic in user-defined rules.
- Standard Mamdani and Sugeno-type fuzzy inference systems.
- Automated membership function shaping through neuroadaptive and fuzzy clustering learning techniques.
- Ability to embed a fuzzy inference system in a Simulink model.
- Ability to generate embeddable C code or stand-alone executable fuzzy inference engines.

The Fuzzy Logic Toolbox provides GUIs to let you perform classical fuzzy system development and pattern recognition. Using the toolbox, you can:

- Develop and analyze fuzzy inference systems
- Develop adaptive neurofuzzy inference systems
- Perform fuzzy clustering

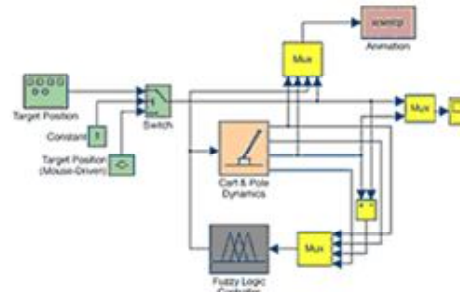


Fig.1. Fuzzy Tool Box

In addition, the toolbox provides a fuzzy controller block that you can use in Simulink to model and simulate a fuzzy logic control system. From Simulink, you can generate C code for use in embedded applications that include fuzzy logic.

Fuzzy inference is a method that interprets the values in the input vector and, based on user-defined rules, assigns values to the output vector. Using the GUI editors and viewers in the Fuzzy Logic Toolbox, you can build the rules set, define the membership functions, and analyze the behavior of a fuzzy inference system (FIS). The following editors and viewers are provided:



Fig.2. Modeling

- FIS Editor - Displays general information about a fuzzy inference system
- Membership Function Editor - Lets you display and edit the membership functions associated with the input and output variables of the FIS
- Rule Editor - Lets you view and edit fuzzy rules using one of three formats: full English-like syntax, concise symbolic notation, or an indexed notation

- In MATLAB[17] two types of Inference Methods used are Mamdani and Sugeno[19] inference method. Mamdani style inference is based on the work by Ebrahim Mamdani in 1975. Mamdani style inference expects the output variables to be fuzzy sets. Sugeno method can be ideal for adaptive techniques, nonlinear systems modeling and interpolation. Defuzzification is the process of producing a quantifiable result in fuzzy logic. Fuzzy logic[8] is a rule-based system written in the form of horn clauses (*i.e.*, if-then rules). These rules are stored in the knowledge base of the system. The input to the fuzzy system is a scalar value that is fuzzified. The set of rules is applied to the fuzzified input. The output of each rule is fuzzy. These fuzzy outputs need to be converted into a scalar output quantity so that the nature of the action to be performed can be determined by the system. The process of converting the fuzzy output is called defuzzification. There are many defuzzification techniques but primarily only three of them are in common use. These defuzzification techniques are Maximum Defuzzification Technique, Centroid Defuzzification Technique and Weighted Average Defuzzification Technique.

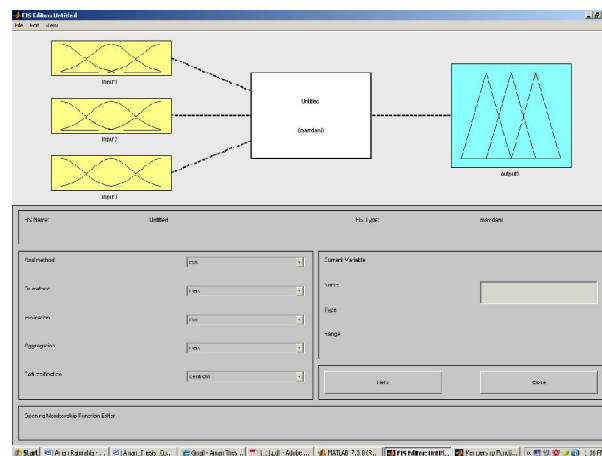


Fig.3. FIS Editor

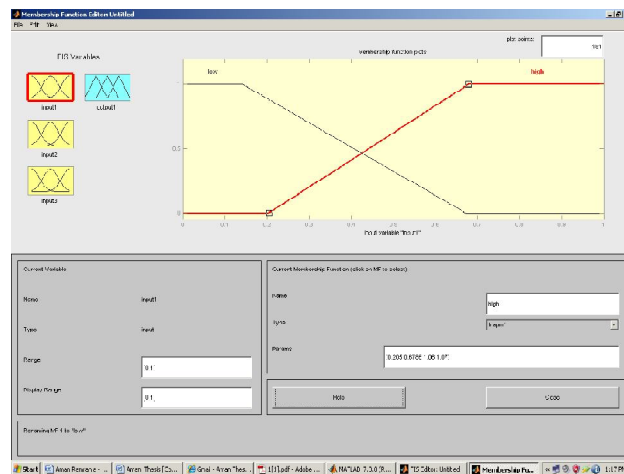


Fig.4. Fuzzy input sets for input 1 “TMC”

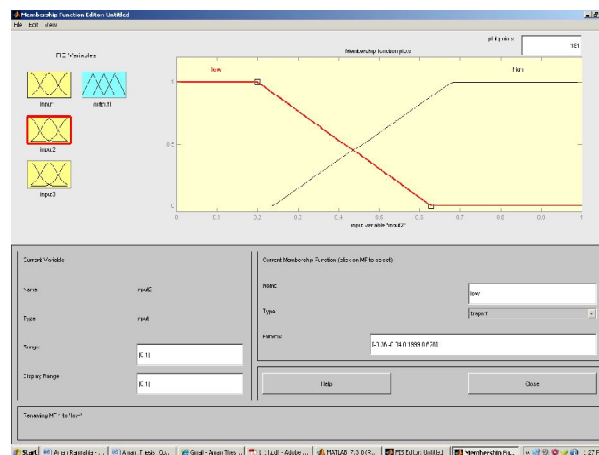


Fig.5. Fuzzy input sets for input 2 “MP”

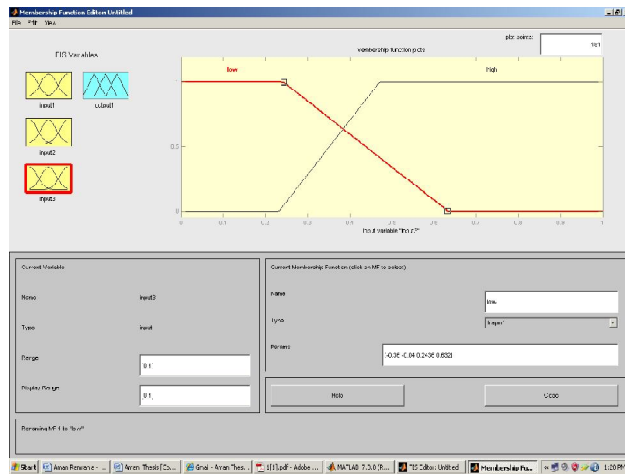


Fig.6. Fuzzy input sets for input 3 “FP”

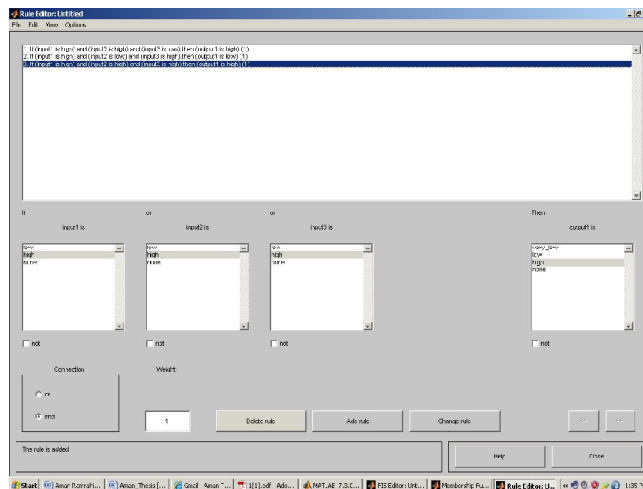


Fig.7. Fuzzy Logic Rules

The rule viewer of Fuzzy simulation in MATLAB for one case as a sample. Other cases have been also executed and the output values are tabulated as shown in the table 1, below.

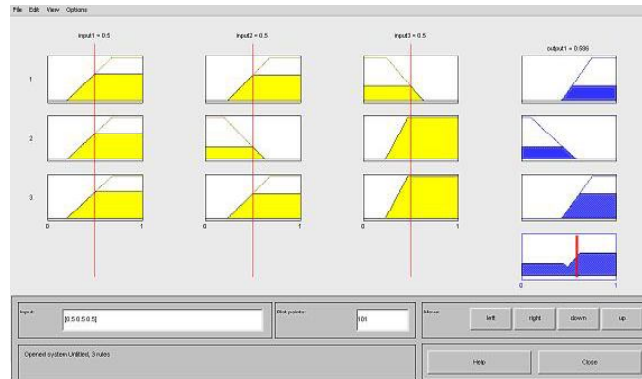


Fig.8. Rule Viewer for one sample case

The issues considered while selecting a PERT as Total minimum cost taken for a project to complete, manpower required to complete the project, function points required to complete it. We have taken these factors as input values from which we can get a single output in terms of PCQ (Pert Chart Quotient).

The table carrying the inputs and output values for different combinations of the inputs has been given as shown below.

S. No.	Inputs			Output
	TMC	MP	FP	PCQ
1	0.2	0.3	0.4	0.53
2	0.4	0.3	0.1	0.72
3	0.3	0.2	0.2	0.48
4	0.1	0.4	0.3	0.43
5	0.5	0.5	0.5	0.58
6	0.6	0.4	0.7	0.45
7	0.6	0.6	0.8	0.74
8	0.7	0.8	0.5	0.69
9	0.8	0.7	0.9	0.78
10	1	1	1	0.78

Table 1: Inputs and Outputs

We would like to choose a PERT Chart that affords a minimum Total Minimum Cost (TMC). Allocate a weight WTC for this cost function. Let Weight age allocated for manpower is WMP and its associated cost MP. Another factor is functional point (FP) and allocated weight will be WFP. We define a factor for comparison of PERT charts and call it PERT Chart Quotient (PCQ)

$$PCQ = (TMC * Wtc + MP * Wmp + FP * Wfp) / (Wtc + Wmp + Wfp)$$

VI. CONCLUSIONS

The main goal of this paper is to get an optimal PERT chart using fuzzy logic. The issues considered while selecting a PERT as total minimum cost taken for a project to complete, manpower required to complete the project, function points required to complete it. All the three constraints are crashed on time basis. The constraints are crashed by considering the crashed time only. Now the optimal PERT is to be selected using fuzzy logic. We developed a fuzzy expert system which is used to select an optimal PERT chart. Optimality is described here within the rules of fuzzy logic. The soft computing techniques are used to develop the Software management tools. Thus an attempt has been made to develop a tool to select an optimal PERT of a project. This can be used as a tool by the project managers to select an optimal PERT chart in a project. The PERT can be crashed not only by considering the issues like cost, manpower and function points, but also with the issues like the vitality, quality, delivery date etc., basing on the interest of the project manager and also the choice of the customer. It is concluded that the object-oriented template built can be used as a software tool to select an optimal PERT chart.

VII. REFERENCES

- [1] M. Aksit and F. Marcelloni, Deferring Elimination of Design Alternatives in Object- Oriented Methods, in: Concurrency Practice and Experience, 2000.
- [2] M. D. Hakel, "How Often is Often", American Psychologist, pp. 533-534, 1968.
- [3] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, "Object-Oriented Modeling and Design", Prentice-Hall, 1991.
- [4] F. Marcelloni and M. Aksit, "Reducing Quantization Error and Contextual Bias problems in Software Development Processes by Applying Fuzzy Logic", Proceedings of the 18th International Conference (NAFIPS'99), pp. 268-272, New York, June 1999.

- [5] F. Marcelloni and M. Aksit, Improving Object- Oriented Methods by Using Fuzzy Logic, in: ACM Applied Computing Review, 2000.
- [6] B. Tekinerdogan and M. Aksit, “Modeling Heuristic Rules of Methods”, University of Twente, Department of Computer Science, 1998.
- [7] L.A. Zadeh, “Outline of a New Approach to the Analysis of Complex Systems and Decision Processes”, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-3, No.1, pp. 28-44, January, 1973.
- [8] Angelov, P. and Buswell, R. “Identification of evolving fuzzy rule-based models” IEEE Transactions on Fuzzy Systems, Vol.10, Issue 5, pp.667-677, Oct.. 2002
- [9] Foulloy, L. and Galichet, S., “Fuzzy control with fuzzy inputs” IEEE Transactions on Fuzzy Systems, Vol.11, Issue 4, pp.437-449, August 2003
- [10] Baranyi, P. Koczy, L.T. and Gedeon, T.D., “A Generalized Concept for Fuzzy Rule Interpolation” IEEE Transactions on Fuzzy Systems, Vol.12, Issue 6, pp. 820-837, Dec. 2004
- [11] Park, Y., Tahk, M.-J. and Bang, H., “Design and Analysis of Optimal Controller for Fuzzy Systems With Input Constraint” IEEE Transactions on Fuzzy Systems, Vol.12, Issue 6, pp. 766-779, Dec. 2004
- [12] Li, Y. and Li, S., “A Fuzzy Sets Theoretic Approach to Approximate Spatial Reasoning”, IEEE Transactions on Fuzzy Systems, Vol.12, Issue 6, pp. 745-754, Dec. 2004
- [13] Gazdik, “Fault detector and prevention by fuzzy sets,” IEEE Trans. Rel., vol. R-34, pp. 382–388, Oct. 1985.
- [14] “Engine fault analysis part II: Parameter estimation approach,” IEEE Trans. Ind. Electron., vol. 32, pp. 301–307, Nov. 1985.
- [15] S. R. Chu, R. Shoureshi, and M. Tenorio, “Neural networks for system identification,” IEEE Contr. Syst. Mag., vol. 10, pp. 31–35, Apr. 1990.
- [16] I.Alguindigue, A. L. Buczak, and R. E. Ulrig, “Monitoring and diagnosing element bearing faults using neural networks,” IEEE Trans. Ind. Electron., vol. 40, pp. 209–217, Apr. 1993.
- [17] M. Y. Chow, R. N. Sharpe, and J. C. Hung, “On the application and design of artificial neural networks for motor fault detection—Part 1,” IEEE Trans. Ind. Electron., vol. 40, pp. 181–188, Apr. 1993.

- [18] J. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 665–685, May/June 1993.
- [19] C. Li and C.-H. Wu, "Automatically generated rules and membership functions for a neural fuzzy-based fault classifier," in *Proc. Midwest Symp. Circuits and Systems*, 1994, vol. 2, pp. 1377–1380.
- [20] R. N. Sharpe, M. Y. Chow, S. Briggs, and L. Windigland, "Methodology using fuzzy logic to optimize feed forward artificial neural network configurations," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 760–767, May 1994.