IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727,
PP 45-50
www.iosrjournals.org

Soft Computing Techniques for Software Effort Estimation Comparative Study Using Cocomo Ii Model

Mr. Azade Sanjay Y[1,] Dr. Lomte S. S.[2,] Mr. Aqueel Ahmed A. Jalil[3]
[1]Department Of C. S. & I. T. Dr. B.A. M. U. Aurangabad
[2]Principal Dr. Seema Quadri Institute Of Eng. & Tec.Aurangabad
[3]Assist. Professor Radhai Mahavidyalaya Aurangabad

Abstract:- *Software effort estimation is the key success factor for the success of software because the complex and multidependent character of software makes the software effort estimation a challenging task. An estimation which provides the clear enough view of the software and provides a decision making view is referred as a good estimation. An estimate is considered as a deterministic terms i.e. as a single value but due to an inherent estimation uncertainty range estimates are alternatively proposed. Traditionally software effort estimation is associated with planning the amount of resources required to completing project activities but now a days software effort estimation is required to support number of decision making tasks and while doing so one has to select the appropriate effort estimation approach. Software project management decisions often fail because effort estimation methods do not provide sufficient insight into resource related sources of project risk. The COCOMO is an algorithmic software effort estimation model. It uses basic regression formula. The parameters derived for this model are derived from historical projects data and current project characteristics. This model is used for estimating effort, cost and schedule. Reference to this model calls it COCOMO 81. In 1997 COCOMO II was developed and finally published in 2000. It is a successor of COCOMO 81 and is batter for estimation modern software projects.*

I. Introduction

Software effort estimation is an integral part of software development, for the success of software. As the software grows in size and complexity the software effort estimation task is getting complex there for to deal with the complexity since last few years many researchers all over the world tries to develop new modeling techniques which could deal with the changing complexity and increased size of software. In this paper we have presented few of new techniques which are recently presented for software effort estimation. We have presented the main outcomes of few research papers particularly related with soft computing and COCOMO II model, it illustrates the advantages and disadvantages and similarities and dissimilarities between soft computing techniques used for effort estimation**.**

Soft computing is a series of techniques and methods which deal with real practical situations in the same way as humans deal with them, on the basis of intelligence, common sense, consideration of analogies, approaches, etc. in other words, soft computing is a family of problem-resolution methods headed by approximate reasoning and functional and optimization approximation methods, including search methods. Soft computing is therefore the theoretical basis for the area of intelligent systems and it is evident that the difference between the area of artificial intelligence and that of intelligent systems is that the first is based on hard computing and the second on soft computing. Soft Computing is still growing and developing.

Soft Computing is a new multidisciplinary field, to construct a new generation of Artificial Intelligence. The main goal of Soft Computing is to develop intelligent machines and to solve nonlinear and mathematically unmolded system problems (Zadeh 1994) and (Zadeh 2001). The applications of Soft Computing have proved two main advantages. *First*, it made solving nonlinear problems, in which mathematical models are not available, possible. *Second*, it introduced the human knowledge such as cognition, recognition, understanding, learning, and others into the fields of computing.

Soft computing particularly encompass fuzzy logic, neural network, AI, data mining etc**.** in this paper we have used only fuzzy logic , because fuzzy logic is having a capability to deal with the problems of uncertainty and it provides methods for modeling and reasoning under uncertainty. It is easy to map input and output using fuzzy rules. In addition to this using fuzzy logic data can be represented in the form of linguistic values instead of crisp values. The aim of fuzzy logic is to exploit tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve close resemblance with human-like decision making.Software effort estimation methods Historical analogy estimation, expert judgment, model based and rule of thumb [3]Effort estimation based on statistical regression analysis, regression-based methods share, to a large extent, general characteristics with other data-driven, model-based estimation methods [1].

II.Related work

Sandeep kad and vinay chopra designed Fuzzy COCOMO and revels The Fuzzy Logic COCOMO model which overcomes the uncertainty in inputs in COCOMO and improves the accuracy of software effort estimation. In [4] the author proves that that when soft computing is applied tomachine learning technique is batter for predation of effort, they uses the fuzzy logic with three inputs with triangular membership functions and observed that the fuzzy model provide fair results for effort estimation. The main limitation of the analytical structure of FLC is that as the number of inputs increases the rule base also increases but for every input all rules of FLC are to be verified although only few rules would be fired, it is time consuming and the rules should be minimized. [5] States that data mining and soft computing techniques with linguistic terms and trapezoidal membership function and associative classification for generating rules improves the accuracy of estimation. For optimization of rules GA is used. [6] the model is designed based on fuzzy logic and two parts of COCOMO,fuzzified nominal effort prediction and effort adjustment factor and says that the performance of FIS can be increased by increasing number of membership functions. The key notable thing in this paper is Gaussian membership function is used for cost estimation and observed that the results are very near to actual effort. [7] Reusability is very important it allows the selection of components and helps to improve the efficiency of the system and decreases the effort and cost of the system, the used the sensitivity analysis the modes and shows that one is batter, the main motto of this research is to estimate the reusability of components using soft computing. [8]Novel ANN uses features of ANN and maintains the merits of COCOMO ANN calibrates attributes of COCOMO using past projects data and observed the improvement in accuracy of COCOMO II as compare to original COCOMO II model

III. Cocomo II Model

The COCOMO method represents a data-driven, model-based, parametric estimation method that implements a fixed-model approach. In other words, COCOMO provides a fixed estimation model that has been built on multi organizational project data using statistical regression, which represents a data-driven, parametric method.

Boehm (1981) developed the first COCOMO model using a multiple regression analysis. The most recent COCOMO II results from calibrating the original model, which Boehm et al. (2000) conducted using measurement data and expert judgment. For this purpose, they used a hybrid approach of model parameters learned from measurement data using statistical regression and provided directly by human experts, which have been integrated using Bayes' Theorem. Major additional capabilities of the COCOMO II model compared to the original model include the following (Boehm et al. 2000)

- Size measurement is custom tailorable involving lines of source code (LOC), Function Points, or Object Points metrics.
- The model accounts for reuse and reengineering.
- Five exponential factors are used for modeling diseconomies of scale.
- Several additions, deletions, and updates with respect to effort drivers and their impact on effort as compared to the previous COCOMO model.

Cocomo II defines three sub models: application composition, early-design model and post architecture model. These models offer different effort drivers, effort equations, and different rules for measuring software size, depending on the project phase in which estimation takes place.

The early-design model is intended for use in the early stages of a software project when very little is known about the nature of the project and the software products to be delivered. Such aspects as the size of the product to be developed, the target platform, the personnel to be involved in the project or the detailed specifics of the processes to be employed are typically not known in very early phases of the project. In order to cope with sparse information, the early-design model requires less data than the post-architecture model, yet provides less accurate estimates. The model is used to make rough estimates of a project's effort before the entire software architecture is determined; that is, while alternative software architectures and concepts of operation are still explored.

The post-architecture model represents a detailed estimation model, and it is intended to be used after an overall software architecture has been developed; that is, during actual development and maintenance of a software product. In order to produce more detailed estimates, the post-architecture model requires appropriately more input information, which should, however, be available by the time of the intended model's use.As a regression model, COCOMO II is used mainly for prediction purposes. In order to predict the effort of a new project, one measures or estimates project parameters required as input to COCOMO and use these data for computing effort according to the COCOMO II equation. It has 17 Effort multipliers and 5 scale factors following table summarizes the COCOMO II drivers and effort multipliers.

Table 1. Post Architecture Cost drivers andScale Factor

Effort Multipliers of COCOMO II Name	Abbreviation	Grouping of cost drives depending on impact
Reliability	**RELY**	PG Pessimistic group The range values of these multipliers are directly proportional to effort
Database size	DATA	
Product complexity	CPLX	
Reusability	RUSE	
Documentation needs	DOCU	
Execution time	TIME	
Storage	STOR	
Platform volatility	PVOL	
Analyst capability	**ACAP**	OG Optimistic group The range values of these multipliers are inversely proportional to effort
Application experience	APEX	
Platform experience	PLEX	
Language and tool experience	LTEX	
Personal continuity	PCON	
Software tools used	TOOL	
Programmer capability	PCAP	
Multi-Site Development	SITE	
Schedule	SCED	

Scale Factor	Abbreviation
precedentedness	**PREC**
Development flexibility	FLEX
Risk resolution	RESL
Team cohesion	TEAM
Process Maturity	**PMAT**

In Cocomo IImodel Effort is measured in person-months and is a function of development baseline productivity, software size, and several effort drivers. Size represents volume of software product and is measured in terms of thousands of lines of source code (KLOC).

$$\text{Effort} = A \text{ x Size}^{E} \text{ x } \mathop{\pi}\limits_{i=1}^{n} EMi\_\_\_\_\_\_\_\_\_\_\_\_ \qquad [1]$$

Where

- Effort represents total project effort
- A represents development productivity and is initially set up to A = 2.94
- EM represents effort drivers
- E represents effect of scale and is computed as follows:

$$E = B + 0.01 \text{ x} \sum_{J=1}^{5} SFj\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \qquad [2]$$

Where

- B represents a constant and is initially set up to B = 0.91
- SF represents scaling factors

The development time is calculated as equation 3

$$TDEV = C \text{ x } E^{f}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_[3]$$

Where c = 3.67

The coefficient F is as

$F = D + 0.2 \text{ x } 0.01 \text{ x } \sum SF_j$ or $F = D + 0.2 \text{ x } (E-B)$ where D = 0.28 The new value of C is 3.67 and F is determined as $F = 0.28 + 0.002 \sum SFj$ according to [9] if we consider all the values of factors and multipliers are considered as nominal the effort is calculated as Effort = 2.94 x Size 1.1 and duration TDEV = 3.67 x (Effort) 3.18

The COCOMO II Post Architecture model is given as

$$PM = A \text{ x [Size] } 1.01 + \sum_{i=1}^{5} SFi \text{ x } \mathop{\pi}\limits_{i=1}^{17} Emi\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \qquad [4]$$

Where PM is effort expressed in person month. A is Multiplicative constant Size is project size expressed in KLOC EM are effort multipliers 1,2, ….17 and SF_i are 5 exponent scale factors which has exponential effect of increasing or decreasing amount of effort. The 17 EM's are grouped in four categories as product attributes, platform attributes, personal attributes and computer attributes.

IV. Fuzzy logic

Fuzzy logic is used to solve the complex problems, which is based on fuzzy set theory introduced by Prof. Zadhe in [9] using fuzzy logic linguistic constructs can be used i.e. fuzzy logic provides inference structure which enables human reasoning capabilities[10] Fuzzy systems are rule based they uses the if then rules. The fuzzy logic systems can be of three categories as pure Fuzzy, Takagi fuzzy and Surgeon Fuzzy, and fuzzy system with fuzzifier and defuzzifier out of the three the fuzzy system with fuzzifier and defuzzifier is the most popularly used which is proposed my Mamdani, it consist of four main components as Fuzzifier, Fuzzy Rule base, Fuzzy inference engine, and defuzzification.

Fuzzifier converts a crisp input to a fuzzy set, rule base uses the fuzzy if then rules, inference engine fuzzifies the linguistics values and defuzzification converts fuzzy output into crisp output. [11]Membership functions are used in fuzzification and defuzzification steps of an fuzzy logic system for the purpose of mapping fuzzy logic linguistic terms and non-fuzzy inputs or vice a versa, membership functions are used for quantification of linguistic terms.

V. Proposed work

The main objective of our research is to use concept of soft computing particularly fuzzy logic with COCOMO II for achieving accurate software effort estimation and reduce the uncertainty in COCOMO II model. Architecture of proposed fuzzy model is as below The proposed fuzzy model has for input as Size of the project, and 17 cost drivers whose values are qualitatively defined as very low, low, nominal, high, very high, and extremely high devised in to two groups depending on their impact on the effort estimation as OG Optimistic group The range values of these multipliers are inversely proportional to effort and PG Pessimistic group The range values of these multipliers are directly proportional to effort.

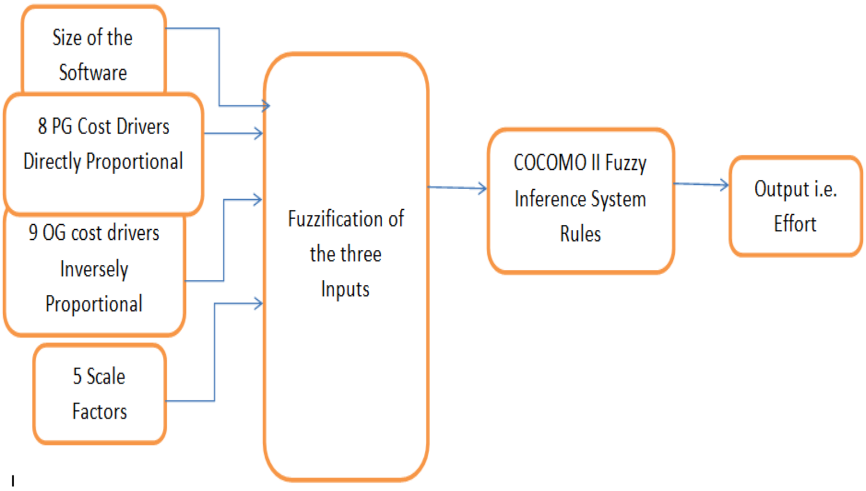
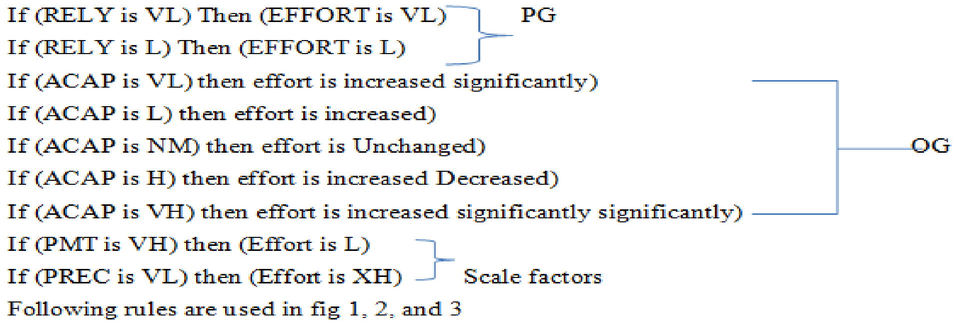

Size of the Software
8 PG Cost Drivers Directly Proportional
9 OG cost drivers Inversely Proportional
5 Scale Factors
Fuzzification of the three Inputs
COCOMO II Fuzzy Inference System Rules
Output i.e. Effort

Fig1 Proposed Methodology

We used Trapezoidal membership function, Gaussian membership function and Triangular mmembership function for analysis and the results of three are compared with the efforts computed by COCOMO II and the actual results of the subset of NASA 93 project dataset from PROMISE software Engineering Repository data set which is publically available for research purpose which consist of 93 projects data from different centers of various years. In fuzzification fuzzy rules are defined by using linguistic variables based on connective AND between Input variables.

Some of the rules are


PG
OG
Scale factors

If (RELY is VL) Then (EFFORT is VL)
If (RELY is L) Then (EFFORT is L)
If (ACAP is VL) then effort is increased significantly)
If (ACAP is L) then effort is increased)
If (ACAP is NM) then effort is Unchanged)
If (ACAP is H) then effort is increased Decreased)
If (ACAP is VH) then effort is increased significantly significantly)
If (PMT is VH) then (Effort is L)
If (PREC is VL) then (Effort is XH)
Following rules are used in fig 1, 2, and 3

Fig.2 ACAP Cost Driver using Trapezoidal Membership Function

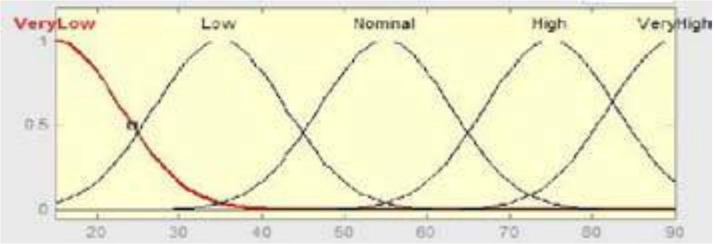
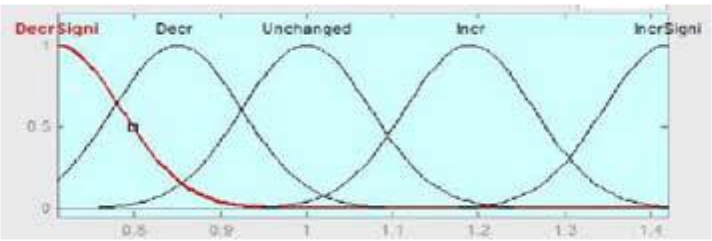


Fig 3 ACAP Cost Driver using Trapezoidal Membership Function



VI. Experiential results

A subset of NASA 93 project dataset from PROMISE software Engineering Repository data set dataset is applied to the proposed fuzzy model i.e. software development effort obtained using COCOMO II and efforts obtained by using fuzzy logic using gauss mf are calculated. It is observed that the effort obtained after applying fuzzylogic was closer to actual effort as compared to COCOMO II. The parameter used for evaluation of proposed model is MRE and is given by

$$\text{MRE} = \frac{|\text{Actual Effort} - \text{Predicted Effort}| \quad \text{X } 100}{\text{Actual Effort}}$$

When the MRE of COCOMO II and MRE of proposed model are compared the MRE of proposed model is found less it indicates that the proposed model gives good results.

VII. Conclusion and future scope

The research shows the direction for further research and the proposed framework can be analyzed for feasibility and acceptance for live industry projects and as the research is on the way in feature we can also improve the estimate by Takagi Surgeon FLC or by determining more fuzzy rule set and deploying ANN uncertainty can be handled more closely and thus more accurate software effort estimation is possible. alsotry to combine the different techniques like Simulated Annealing for calculating the best results.

References

[1]. L. Schroeder, D. L. Sjoquist, and P. E. Setphan (1996), Understanding Regression Analysis: An Introductory Guide. Sage Publications, Inc.The book provides a basic introduction to regression analysis. It aims at beginners in this topic and provides background knowledge for understanding.

[2]. B.W. Boehm, C. Abts, A.W. Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece (2000), Software Cost Estimation with COCOMO II, Upper Saddle river, New Jersey: Prentice-Hall

[3]. Handbook of software cost estimation Karen Lum, Michael Bramble, et.al. JPL may 30 2003

[4]. Software development effort estimation using soft computing by Sandeep kad and VINAY Chopra IJML&C Vol 2 No. 5 October 2012.

[5]. "Analytical structure of fuzzy logic controller for software development effort estimation " Rama Sree p 978-1-4799-2/15 @ 2015 IEEE

[6]. "Software Cost Estimation using Improved Fuzzy logic framework" NeetuKushwaha , Suryakant 978-1-4799-3064-7/14 @2014 IEEE

[7]. " Estimation of software reusability for component based system using soft computing" Charu Singh, AmrendraPratap, AbhishekSinghal 978-1-4799-4236-7/14 @2014 IEEE

[8]. " Proposing an enhanced artificial nural network prediction model to improve the accuracy in software effort estimation" ImanAttarzadeh , AmanMehranzadhe, Ali Barati 2012 4t national Conference on CICSN 978-0-7695-4821-0/12 @2012 IEEE

[9]. A. Idri, A. Abrian and L. Kjiri, " COCOMO cost model using Fuzzy Logic" International Conference on Fuzzy Theory and Technology Atlantic, New Jersey. 2000.

[10]. Ashita Malik, VarunPande, AnupamaKaushik, “An Analysis of Fuzzy Approches for COCOMO II” IJISA 2013 , 05, 68-75 in MECS.

[11]. “Software Cost estimation using the improved Fuzzy logic Framework” , NeetuKushwaha, Suryakant 978-1-4799-3064-7/14@2014 IEEE

[12]. **Somerville,"Software** Engineering", 6th ed., Addison– Wesley Publishers Limited, 2001.

[13]. H. K. Verma, V. Sharma,“Handling imprecision in inputs using fuzzy logic to predict effort in software development”, IEEE International Advance Computing Conference (IACC), pp. 436-442, 2010.

[14]. M.W. Nisar, J. W. Yong, M. Elahi,“Software developmenteffort estimation using fuzzy logic - A survey”, IEEEInternational Conference on Fuzzy Systems and KnowledgeDiscovery, Vol. 1, pp. 421-427, 2008.