A review paper on Optimization of model for estimating software effort Performance analysis of the software cost estimation methods

G.S.Aditya Rao[1], Dr.Ravindra Gupta[2],

[1] M.Tech Scholar, Computer Science Department, SRK University , Bhopal, India
[2] Professor, Computer Science Department, SRK University , Bhopal, India

Abstract

In programming designing, assessments are every now and again used to decide expected yet obscure properties of programming improvement measures or the created frameworks, for example, costs, time, number of engineers, endeavors, sizes, and complexities. A lot of assessment models exist, yet it is difficult to analyze and improve them as programming innovations develop rapidly. We recommend a way to deal with assessment model plan and robotized enhancement taking into consideration model correlation and improvement dependent on usually gathered information focuses. Along these lines, the methodology disentangles model enhancement and determination. It adds to an intermingling of existing assessment models to meet contemporary programming innovation rehearses and give a likelihood to choosing the most suitable ones..

Keyword *Estimation Models, Optimization, Software Engineering.*

1.1 INTRODUCTION

Software cost estimation is the forecasting of development effort and developmenttime mandatory needed to develop a software project. It is considered to be the very primary step of software development process and at the same time considered to be the key task as accurate assessments of growth of the current project, its delivery exactness and its cost control can only be achieved once calculated estimation is accurate. And at broader perspective an accurate estimation of a currently developing software project will result in landing the organization in a better schedule of its futuristic software projects too. With due above reason, software effort estimation has received a considerable amount of attention of many researchers from past so many decades.

1.2 OBJECTIVES

The aims of this study, alongside the objectives to achieve these aims are stated as follows:

To provide a better conceptual understanding of reason of “cost overruns” which is ascertained through an extensive review of the literature, the factors that contribute to the difference between the initially estimated cost and the resulting final costs at project completion.
To develop an efficient software cost estimation model to forecast likely total cost of projects including its time schedule based on past data of completed projects. The various things to attain it include:

- Identification and collection of reliable datasets for the cost modelling process;
- Establishment of an artificial neural network modelling protocol and its hybridization with a meta heuristic algorithm for developing accurate software cost estimation model.
- Validation of the proposed model using precise evaluation criterions like magnitude of relative, mean magnitude of relative error and median of magnitude of relative error and its comparative evaluation with other existing techniques.

1.3 CONTRIBUTIONS

The contributions of this research to the field of software cost estimation are aligned with the research aims described previously. But the reader of this thesis deserves to understand the intentions, the argument and the original contribution of the work

1. A Software Cost Estimation model based on Input Selection Procedure & Artificial Neural Network.
2. A Software Cost Estimation model based on Functional Link Artificial Neural Network & Improved Particle Optimization algorithm.
3. A Software Cost Estimation model based on Artificial Neural Network Model

2.1 INTRODUCTION

Software cost estimation is a practice of developing an estimate of financial as well astime scheduling resources of any software project to be developed. The various such software development activities are given by [PMBOK, 2000]. Moreover, the various techniques and tools used, given inputs, and expected outputs for any general cost estimate are given below in figure 2.1.

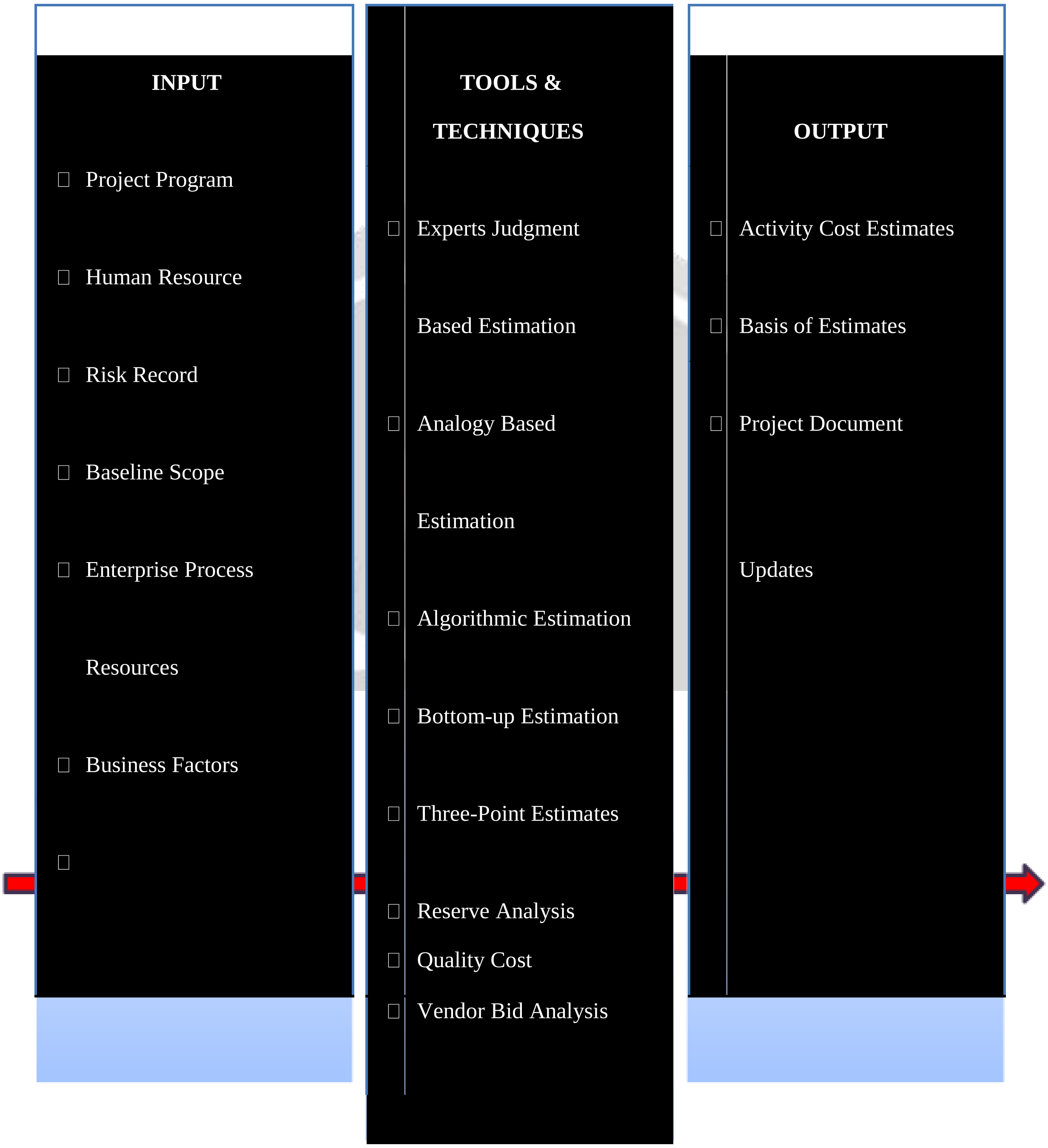


Figure 2.1: Project Management with respect to Cost Estimation

TIME OF ESTIMATION

Software cost estimation is considered as an iterative or an umbrella activity, covering all the phases of software development process and well connected to every other activity of software development process, so never considered as an individual activity to be performed all alone. The various drivers based on which estimate are prepared are coming from requirements gathering of the software project to be developed which never seem to be concrete and clear, thus floating this vagueness to software estimates too. Moreover, as the requirements are powerfully affected by other processes, tools and project attributes, thus add more vagueness to software cost estimation process. As a major part of activities of software development, software cost estimation is given in following

BASIS OF ESTIMATION

The Basis of estimation necessarily must report a complete and unambiguous understanding of how the derivation of cost estimate is done [PMBOK, 2000] as shown in figure 2.1, the information of extra details to be required, including their quantity and kind and more precisely their varying nature as per applications. The various extra details for activity cost estimates include the following:

- A complete estimate basis documentation showing how the estimate was prepared.
- A full blueprint of estimate coverage representing what to estimate.

4. A thorough record of assumptions to be made.
5. Record of constraints.
6. Definition of self-assurance scale of final estimate.
7. Declaration of possible range of estimates that can be possible like an IT item is supposed to cost from a range of 1000 (INR) plus minus 10(INR)

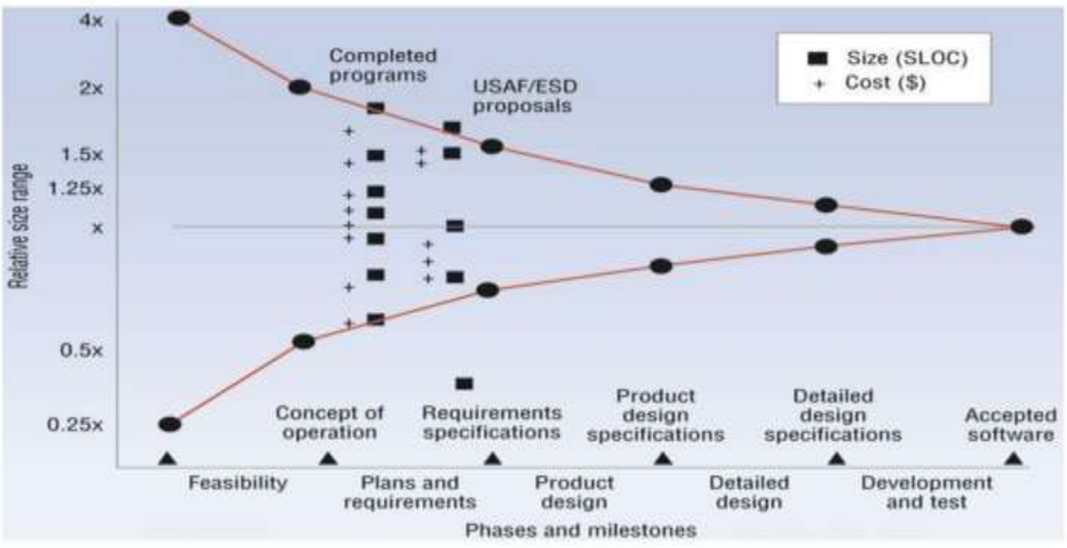
So, it is important to mention here that it becomes easier to go through detailed reviews, to get easy revision of futuristic estimates only once the above information regarding the estimation is clearly defined and documented. However, in contrary to that an ambiguous basis of estimate lets the activity estimates to be useless and hopeless.

SOFTWARE COST ESTIMATION& SOFTWARE QUALITY

In this short section, we will discuss how software cost estimation affects software quality of any software development product. The term software quality is usually observed in a thin logic and in an isolated approach from its process of development. Software quality therefore becomes restricted to concepts like "usability", "error freeness", "extensibility", etc. Product quality thus discounts the two basic parameters of cost and time. Essential software quality, which may in real, be called as software process quality, on the other side, must cover these aspects too. So, the best software in terms of software quality process is often only good enough software in terms of quality of product. This section is aimed to be a support to integral concept of software quality and indirectly to quality of product.

REAL ESTIMATION, MEASUREMENT& ITS RELATIONSHIP

Same applies to software development projects too. Measuring just three economical parameters of effort, time and product can never lead to better software estimation process. Moreover, knowing the average ratio of product and effort, and the ratio between product and time, cannot even give the treasonable values for the parameters before estimation of at least one of them has not been completed. The first significant criteria towards estimation, is to form a model of the final product. A model of one such type is built by measurement of parameters: effort and time of modeling as well as the model‟s product metric is to be done. The second key thing to be done is to draw a correlation between the parameters of the modelling process to those of a complete process of production, with a hope that the earlier predict the latter.



PSYCHOLOGICAL AND ORGANIZATIONAL BARRIERS TO RATIONAL ESTIMATION

Why, in reality, is nobody concerned in a rational cost estimate?

Let us examine this question by looking at the people typically involved in the estimation process:

1.A manager of the software producing unit (SPU).

2.The software products client and

3.The project manager.

4.The SPU‟s manager always keeps his interest in marketing the project just to promote his bonus. Thereby, he is about to sell the software product to any price the customer is ready to pay. The customer, in turn, at every time is willing to pay the least possible. Another significant issue is the knowledge of "truth" regarding the cost prediction at early stage may considerably constrain the deployment of futuristic projects. In other words, we can say that if for any software project one had got the precise knowledge about the cost as early and rationally possible, will surely have a great impact on the decisions to be made regarding the inception or stopping of other forthcoming software projects.

SOFTWARE COST ESTIMATIONAND RISK MANAGEMENT

Software cost and size estimation is a prediction problem of very high magnitude. Software cost estimation is restricted not only to software development companies but is expanded to software‟s to be developed at other places like those of in Research and Development organizations, Aircraft organizations and many alike. Since size of software project is normally considered to be the most dominant factor in defining software‟s cost, good size estimates of any software project size are precarious to good cost estimation.. The impression of risk is dominant to any such kind of analysis, and the two techniques that have got the ability to increase responsibility of holding risks related to software cost estimates include Identification of the zones of uncertainty that later convert and proceed towards risks and Comprehensive Analysis of whole process of software cost estimation process inorder to govern the areas where there is possibility of reduction in vagueness interms of its risk mitigation.

MERITS, DEMERITS& RISKS IN SIZE ESTIMATION

All sizing methods have got both merits and demerits based on their capacity of possessing of knowledge about the system; dissimilarity in the languages made for the purpose of implementing the system and other system compositions which includes the use of new, recycled, and adapted code of the system. Selection of most proper sizing-estimation methods will necessarily aid in mitigation of risks associated with each choice. However, several global issues need to be considered while using a sizing method. A brief discussion about them is given in the following:

1.Counting physical objects like number of lines of code or number of requirements. Advantages include ease of counting and ease of counting automation, Liberty of selection of programming language, Flexibility of storage in a historical database, and flexibility of controlling organizations understanding. Flaws include difficulties in counting early in the development process, dependency on programming or requirements style and inconsistency of methods across different languages.

2.Counting of estimated constructs like application points or function points. These objects unlike physical objects can be easily defined in the early in the software development process, but as theoretical ideas they are most often difficult to track over the course of development. Advantages include ease of generation from a clear specification and persistence across intermediate products (such as design or early code modules). Disadvantages include irregularities as analysts interpret the estimated constructs and the difficulty in calculation of size of embedded systems.

3.Absence of empirical proof, specifically in-case of novel sizing methods. A novel sizing method can be appropriate for any novel technique than any existing methods, but without having any empirical evidence to be available to recommend suitable values for the input variables.

BASIC COCOMO

It is the fundamental software cost estimation model made to estimate software development cost of projects of the nature of small to medium size. Being simple and easy to use, this model provides quick cost estimates but this simplicity lets its accuracy to be limited by some measure, thus this model is normally used for rough and initial estimates of cost of software projects.

Table 3.2: Basic COCOMO Coefficients

Software Project	ab	bb	cb	db
Organic	240	105	250	38
Semi-attached	300	112	250	35
Embedded	360	120	250	32

However, there exist so many factors of cost which are excluded by basic COCOMO. These cost factors include Constraints interms of hardware, Capability, experience and capacity of the personnel involved in overall software development process.

INTERMEDIATE COCOMO

The software development effort estimation using intermediate COCOMO is a function of two things namely 15 attributes of cost factors and program size. These 15 attributes are categorised into following four groups:

1.Attributes related to Software product,

2.Attributes related to Machine hardware,

3.Attributes related to People involved,

4.Attributes related to overall project.

SOFTWARE COST ESTIMATION BASED ON FUNCTIONAL LINK ARTIFICIAL NEURAL NETWORK MODEL& IMPROVED PARTICLE SWARM OPTIMIZATION

In this study, software development effort and development time is estimated by choosing a functional link artificial neural network. A functional link artificial neural network is a high order, single-layer feed forward artificial neural network. It consists of a layer for taking inputs, called as input layer and a layer for dispatching output, called as output layer. Moreover, it contains no hidden layers as the whole processing to be done on inputs in hidden layers is primarily done using functional expansion before being passed to the input layer. Output interms of software cost estimation in our study is generated by the functional link artificial neural network by simply intensifying the input (cost drivers). Every single input neuron keeps a correspondence to a component of input vector. The output layer on the other hand is composed of a single output neuron that calculates the software development effort as a linear weighted sum of the outputs originating from the input layer.

DATA SETS USED IN THE EXPERIMENT

For the purpose of assessing the software cost estimation using the proposed artificial neural network based models, four data sets from different companies are chosen. One of the data sets has been got from the study of Mair et al. In this study, 32 data sets[Mair, C.,*et*.al, 2005] were available publicly among which only one data set COCOMO 81 has been selected as this is the lone dataset containing data of more than 50 software development projects.

EXPERIMENTATION RESULTS& THEIR ANALYSIS

This subsection presents the experimentation results of the proposed technique discussed in section 4. Here in this subsection, we‟ll first present the results obtained while implementing the software cost estimation model based on hybrid of input selection procedure and artificial neural network model. Next we‟ll present the obtained experimentation results of software cost estimation model based on functional link artificial neural network and improved particle swarm optimization.

Table 5.2

MRE (%) of Proposed Model and other Two Existing Techniques on 11 randomly Selected Projects of COCOMO81 Dataset.

S No	Project ID	MRE (%) on COCOMO dataset		
		COCOMO II Model	B.T Rao, *et al.* Model	Proposed Model
01	05	7.44	5.23	3.44
02	12	19.83	9.18	7.12
03	30	6.49	3.21	2.22
04	38	50.98	14.40	11.10
05	40	12.40	4.89	3.89
06	45	5.35	3.69	2.72
07	47	16.40	7.60	4.07
08	59	8.66	3.20	2.97
09	61	13.10	8.61	4.12
10	62	6.22	3.30	2.88

Table 5.3

MRE of Proposed Model and other Two Existing Models on 10 Randomly Selected Projects of Cocnasa Dataset.

S No	Project ID	MRE (%) on Cocnasa dataset		
		COCOMOII Model	B.T Rao, *et al.*	Proposed Model
01	1	9.33	5.12	3.68
02	5	8.84	3.78	2.89
03	15	16.75	8.80	6.44
04	25	14.09	5.68	3.88
05	30	8.81	4.40	3.12
06	42	13.9	6.60	4.17
07	54	13.67	7.80	4.90
08	60	11.78	6.83	5.22
09	62	13.2	5.43	2.12
10	75	17.09	8.66	5.10

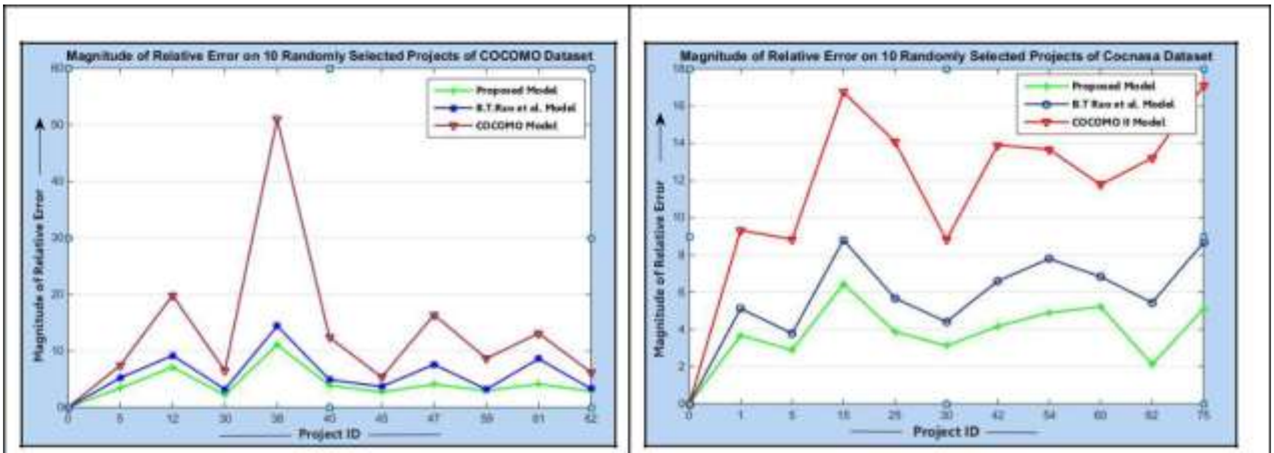


Figure5.1

MRE (%) Based Graphical Description of TwoExisting

Models and Proposed Model on Cocnasa Data Set

6 CONCLUSION

Designing a software system requires software effort estimation significantly. Numerous research works have been carried out to increase the precision of effort estimate of the software system. This paper has proposed a novel approach to estimate the software effort precisely. The approach has been contributed by neural network classification process and an optimization process. The neural network has classified various software parameters. For betterment of classification performance, ABC has been used to optimize the weights of neural network. Error parameters such as MRE, MMRE and MARE have been determined and performance comparison has been made with the existing method. The experimental outcomes have demonstrated the proposed system outperform the existing method in estimating the software effort more precisely.

REFERENCES

- Albrecht, A. (1979). Measuring Application Development Productivity. In Press, I., editor, IBM Application Development Symposium, pages 83–92.

- Beck, K. and Andres, C. (2004). Extreme Programming Explained: Embrace Change (2nd Edition). AddisonWesley Professional.

- Boehm, B. W., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., Madachy, R., Reifer, D. J., and Steece, B. (2009). Software Cost Estimation with COCOMO II. Prentice Hall Press, Upper Saddle River, NJ, USA, 1st edition edition.

- Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD ’00, pages 93–104, New York, NY, USA. ACM.

- Brooks, Jr., F. P. (1995). The Mythical Man-month (Anniversary Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

- Clark, B. (2015). COCOMO R III project purpose. http://csse.usc.edu/new/wp-content/uploads/2015/04/ COCOMOIII Handout.pdf. Accessed April 12, 2019.

- Fletcher, R. (1987). Practical Methods of Optimization; (2nd Edition). Wiley-Interscience, New York, NY, USA.

- Galorath, D. D. and Evans, M. W. (2006). Software Sizing, Estimation, and Risk Management. Auerbach Publications, Boston, MA, USA.

- Huang, L. (2017). A quasi-newton algorithm for large-scale nonlinear equations. Journal of Inequalities and Applications, 2017(1):35.

- Humphrey, W. S. (1996). Using a defined and measured personal software process. IEEE Software, 13(3):77– 88.

- Jenson, R. L. and Bartley, J. W. (1991). Parametric estimation of programming effort: An object-oriented model. Journal of Systems and Software, 15(2):107 –