

Performance of Cost Assessment on Reusable Components for Software Development using Genetic Programming

T. Tejaswini

BE-III Year, Department of CSE, CBIT, Hyderabad
Email: tejaswini972@gmail.com

J. Sirisha Devi

Department of CSE JNTU (H), Hyderabad
Email: siri.cse21@gmail.com

Dr. N. Murali Krishna

Department of CSE GIT, GITAM University
Email: muralinamana@gmail.com

Abstract—Reusability is the quality of a piece of software, which enables it to be used again, be it partial, modified or complete. A wide range of modeling techniques have been proposed and applied for software quality predictions. Complexity and size metrics have been used to predict the number of defects in software components. Estimation of cost is important, during the process of software development. There are two main types of cost estimation approaches: algorithmic methods and non-algorithmic methods. In this work, using genetic programming which is a branch of evolutionary algorithms, a new algorithmic method is presented for software development cost estimation, using the implementation of this method; new formulas were obtained for software development cost estimation in which reusability of components is given priority. After evaluation of these formulas, the mean and standard deviation of the magnitude of relative error is better than related algorithmic methods such as COCOMO formulas.

Index Terms—COCOMO formulas, cost estimation, genetic programming, magnitude of relative error, reuse of component.

I. INTRODUCTION

In the course of software development, project development is split into variety of activities. Software project price estimates to see the project budget or crucial package costs for the client area unit essential [1][2]. as well as the effective parameters calculated total price of package development is pointed to travel and coaching prices, hardware and package prices and energy prices that's the dominant think about most development (the salaries of engineers concerned within the project, social and insurance costs). There are no easy thanks to create Associate in nursing correct estimate of the hassle needed

to develop a software package. Initial estimates area unit supported inadequate info during a user necessities definition. The folks within the project could also be unknown. The package could run on unfamiliar computers or use new technology. However, organizations ought to estimate the price and work of production package. To attenuate the price of development reprocess of the key elements is important [27] [28].

A component is believed of degree freelance interchangeable a locality of the appliance that provides a clear distinct performs. A component could be a coherent package of code which can be severally developed and delivered as a unit, that offers interfaces by that it's connected unchanged with totally different parts to compose a much bigger system [29].

There are two basic approaches for reutilization of code: develop the code from scratch or establish and extract the reusable code from already developed code. For the organization that has experience in developing package, but has not but used the package recycle construct, there exists extra price to develop the reusable components from scratch to form and strengthen their reusable package reservoir [29]. The worth of developing the package from scratch is saved by characteristic and extracting the reusable components from already developed package systems or bequest systems.

There are two main types of cost estimate models: algorithmic models and non-algorithmic models. The most regular and not the most accurate method of cost estimation, is algorithmic models. Algorithmic models can be created with the cost analysis and completed development characteristics. In this, work genetic programming method is used for software development cost estimation by reusing the components.

In continue of this work, in section 2 related work is described and in section 3 and 4 the problem and genetic programming is investigated. In section 5, cost estimation

using genetic programming and practical results. In the section 6 a discussion on matter of fact and result is depicted. In section 7 results of the experimentation are discussed. Section 8 gives a conclusion and future work.

II. RELATED WORK ON ALGORITHMIC MODELS

To be able to get reliable estimates, we want to record historical knowledge extensively. This historical knowledge can be used to produce estimates for brand spanking new comes. In doing therefore, we have a tendency to predict the expected price owing to measurable properties of the project at hand. even as the price of giving birth out a garden may well be a weighted combination of variety of relevant attributes (size of the garden, size of the grass space, whether or not there's a pond), therefore we might prefer to estimate the price of a computer code development project.

A wide range of modeling techniques have been proposed and applied for software quality predictions. Complexity and size metrics have been used to predict the number of defects in software components. These include: logistic regression [32], discriminant analysis [33], the discriminant power techniques [34], artificial neural network, genetic algorithm and classification trees. Fenton and Neil proposed the Bayesian belief network as the most effective model to predict software quality.

Software cost estimation methods can be generally be divided into two categories: algorithmic models and non-algorithmic models.

A non-algorithmic model doesn't follow any special algorithms to estimate the price of package development. Out of those models are often pointed to the subsequent models: knowledgeable judgment methodology, analog cost accounting methodology, Parkinson's law, value to win strategies, bottom-up methodology and top-bottom methodology [3][4].

Also some algorithmic models in the shown below:

Nelson model could be a linear model began with in depth study of the one zero five attributes 169 software package project. The foremost necessary issue is that in the early stages of this technique; search the parameters for proper estimating the software package prices. This technique shows a distribution of nonlinear programming rates [5].

Walson-Flix model may be a increasing model that every x_i variable seizing solely 3 potential values -1,0,+1. This technique uses four separate factors that were extremely correlate for IBM development [6][7].

Doty model is increasing model that every x_i variable taking up solely 2 attainable values: zero, +1. This model exhibited issue of 2 discontinuities at model boundaries thanks to the binary nature of their inputs [6][7].

PRICE-S model is proprietary package price estimation model that by RCA developed and maintained. Originally, is developed to be used internally on package development like a part of the Apollo program. This model could be a proprietary model that estimate package development like the NASA, department of defense and different government package development [8][9].

ESTIMACS model is life software package and is includes 9 module that the foremost vital module is effort module. Users answer the twenty-five input queries. A part of these queries and also the alternative part is regarding the scale of the quality of software package developed [8][11].

SEER-SEM model may be a product that in time was a tool and supporting top-down and bottom-up methodologies. Its modeling equations square measure proprietary and take a constant methodology to estimation. Vary of the model is in depth and covers all phases of the project life cycle like style, development, delivery and maintenance [8][12].

Check purpose model is software package project estimation tool based mostly data. It's information with 8000 software package development. Check purpose model uses operate purpose and size is its main input [8].

COCOMO model is Associate in nursing experimental model that with collects information from abundant package development is obtained. This information is analyzed to some formulas to get the most effective work with the observations. These formulas, size and system, product, team and project factors square measure joined effort needed for system development. COCOMO model is wide and has several parameters that every of them settle for a spread of values and square measure terribly advanced. In COCOMO'81 assume that package is formed supported the water development method and victimization the quality command programming languages like C or FORTRAN. 3 levels of this model proposed: basic, intermediate and elaborate [10].

On the other hand since the primary version given COCOMO several changes since the software package were developed. Prototyping and incremented development area unit the models unremarkably used. Presently software package is building from reusable parts with prepared systems and connecting them via scripting languages. Current software package is re-engineering to the new software package to be designed. There are a unit CASE tools that support of software package method activities. For these changes, COCOMO II model is about totally different strategies for software package development by combining parts and uses the information programming [30] [31]. This technique uses the spiral model and includes area unit many sub-models. It is spiral in consecutive periods used [13][14].

WEBMO model is algorithmic model for estimate the hassle within the internet development. This model is that the extended COCOMO II model and internet part (WO) is activity consider its. This model has used of the information collected from sixty four internet development and also the model professional judgments. This model permitting its users to see the options of internet development, effort and prices area unit calculated near reality [15].

III. PROBLEM STATEMENT

Currently, computer code quality and utilize may be a main consider the competitive market. For this purpose,

computer code quality needs the event method and project management. During this context, price and energy estimation may be an important step once beginning a replacement computer code project. Because accurate estimates of cost, an important element for competitive prices and success in the market. Price estimates, a region of a method is effective and economical, whether the process of design, testing or software development. In alternative words, estimation may be a basic step within the starting of every project.

An estimation method includes the subsequent operations:

- Obtain information regarding previous development or maybe the previous phases of the project.
- Identity metrics measured, factors and metrics price.
- Create model formulation effort, regarding its relationship with the hassle.
- Create models that for estimating project effort apply.
- Evaluation of estimate model. In alternative words, live of model and confirm the accuracy of estimate model.

In summary, purpose of estimating the hassle, predict the number of effort needed for finishing a given task, supported info and options of previous similar development.

IV. GENETIC PROGRAMMING

Genetic programming is a branch of evolutionary algorithms that solves problems without the need of users. In the genetic programming using genetic algorithms [25][26] and concepts of parse trees, instead of application code to be written, allow the computer only by knowing the concept of work, to prepare the desired application. In fact, a high level command is given to the computer and program desired by the computer is ready, then run the program and provides the desired output.

In genetic programming a population of computer program will be created. That is, generation by generation, randomly from the population of new programs and improved turns. Fig. 1 shows the process control for genetic programming, where natural selection to find the solutions used. Genetic programming, like nature, is a random process, and it can't guarantee results. Moreover, GP has much success in evaluating ways to solve problems in new and unexpected [16].

Basic steps of GP shown below [17]:

- Generate a random initial population using a combination of functions and terminals.
- Execute each of the program population and assign a fitness value to them.
- Create new computer programs, called offspring.
- Copy the best existing programs called reproduction.

- Create a new program by mutation.
- Create new programs by crossover.
- The best computer program obtained in every generation, the best so far, is shown as a result of implementation of GP.

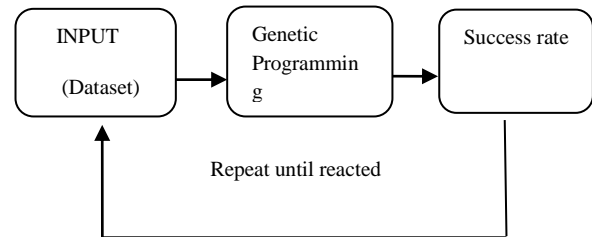


Fig. 1. Process control for genetic programming

In GP, the programs usually are expresses in the syntax tree. For example, fig. 2 shows the tree program $\min(2*x, x-y)$. The constants and variables in the program (2, x, y) are leaves of the tree called terminals, the arithmetic operations (min, +,-,*) are internal nodes that in GP they are called functions.

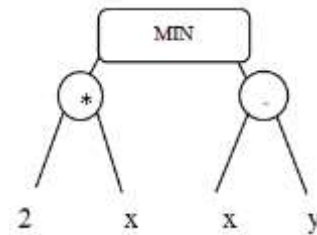


Fig. 2. GP Syntax Tree Representing $\min(2*x, x-y)$

The sets of allowed terminals and functions together form the initial set of a genetic programming. Here the terminal x has been reused.

A. Genetic Programming Parameters

Fitness function can be calculates by different methods such as total or average error between output and desired output.

To execute the program must be set run parameters. Population size is the most important run control parameter, other control parameters include: number of GP generation, possibility of doing genetic operations (mutation and crossover operations) [25], the maximum size of the programs (maximum depth of the trees), determine production method the initial population, determine the selection methods and determine the replacement methods.

Initial population production methods are: full, grow, and ramped half and half methods. Selection methods include roulette wheel method, rank method and tournament method. Two methods generalization and steady-state are used for replacement methods where generalization method can with elitism be done or not [16][18].

V. COST ESTIMATION USING GENETIC PROGRAMMING

In this work new formula exploitation genetic programming for estimate the price of software system development are going to be conferred. The aim of this work is to get formulas that putting software system parameters and merchandise size property, higher estimates than the previous strategies obtained. Considering that the previous strategies scale terribly low to additional high, during this work, the parameters square measure encoded with 2 strategies that Table 1 has shown. These parameters with fixed values and merchandise size per LOC with real price per person-month the applying that runs doctor arrived.

Table 1. Coding Software Parameters

	Very low	Low	Nominal	High	Very high	Extra high
Method 1	1.0	2.0	3.0	4.0	5.0	6.0
Method 2	0.8	0.9	1.0	1.1	1.2	1.3

An initial population of formulas exploitation the ramped half-and-half is made. Any formula exploitation the fitness operates is evaluated and performance of every formula is known. As a result of the structures evaluated in medico are laptop programs, typically the fitness analysis needs implementation could be a program tree suggests that nodes within the tree, the order should be dead that guarantees that nodes don't seem to be dead before the worth of their arguments is understood. This action is completed by the navigation tree recursively ranging from the foundation node [16].

To produce the next generation of formulas using selection methods, formulas are selected for the middle generation. Then genetic operations (mutation and crossover) are done on some members of the population and next generation population is generated. To repeat the genetic programming formulas to switch from this generation an action is completed exploitation replacement ways. This method is once more resumed of analysis formula by fitness operate. The termination condition for genetic programming to get the formula that its fitness is appropriate. Termination condition checked for the simplest formulas obtained in every generation.

With the tip of genetic programming, the specified formula is obtained. This methodology referred to as CESP GP (Cost Estimation of computer code Development exploitation Genetic Programming) methodology is known as inline with the 2 totally different coding ways, 2 formulas with names CESP GP one and CESP GP a pair of it is obtained.

VI. DISCUSSION ON MATTER-OF-FACT

After the performance created for the implementation of genetic programming [25] [26], two formulas using two strategies encryption for information in COCOMO'81 format were obtained. To match the

potency, the two formulas square measure compared with previous algorithmic strategies include: basic and intermediate COCOMO'81, with native standardization (LC) and also the results [23] [24]. As a result of the results square measure clear, COC81 and NASA93 datasets square measure classified to the nineteen subsets.

VII. RESULTS

Data employed in this work, incorporates 3 customary dataset COC81, NASA93 and SDR. COC81 dataset comes from Bohm et al [19] and NASA93 dataset comes from a study funded by the orbiter Freedom Program. Independent agency93 includes information from six totally different NASA centers [20] [25]. SDR information set includes data collected by SoftLab [21][22]. General and statistical properties of used datasets are shown in Table 2.

Table 2. General and Statistical Properties of used Datasets

Dataset	Features	Projects	Data Format	Min	Max	Mean
COC81	16	63	COCOMO I	5.9	11400	683.32
NASA93	16	93	COCOMO I	8.4	8211	624.41
SDR	23	12	COCOMO II	1	22	5.73

The main analysis issue employed in this work is magnitude of relative error (MRE) between the calculable price and therefore the actual price (Equation 1). During this work, exploitation of the mean magnitude of relative error (MMRE) and variance price of relative error (SD) that's shown in equation 2 and 3 and additionally PRED(30) the MD technique with different ways square measure compared. PRED(N) reports the common share of estimates that square measure at intervals just about the particular values (Equation 4).

$$MRE = \frac{(|actual_i - predicted_i|)}{actual_i} \quad (1)$$

$$MMRE = \frac{100}{N} \sum_{N_i=1}^n \frac{(|actual_i - predicted_i|)}{actual_i} \quad (2)$$

$$SD(MRE) = \sqrt{\frac{\sum_{N_i=1}^n (MRE_i - MMRE)^2}{N-1}} \quad (3)$$

$$PRED(R) = \frac{100}{N} \sum_{N_i=1}^n 1 \text{ if } MRE_i \leq R \text{ or } 0 \text{ otherwise} \quad (4)$$

For software package development price estimation victimization genetic programming, desired knowledge with the parameters, size of product and therefore the actual price of the project (in term of person-month) as were input within the program with C#.NET was created. Once reviewing totally different modes for decisive implementation parameters of document, the ultimate implementation was through the parameters in Table 3.

Table 3. GP Implementation Parameters for Cost Estimation

Parameter	Value
Population size	200
Iteration(generation)	5000
Crossover rate	0.8
Mutation rate	0.2
Max level of trees	9
Create initial population method	Ramped half and half method
Selection method	Rank method
Replacement method	Generalization method with elitism
Function set	+,-,*,./,power,-(unary)
Fitness function	MMRE

VIII. CONCLUSION AND FUTURE WORK

We determined that the genetic programming method is a good algorithmic method for software development cost estimation. Also, formula obtained using this method is a good alternative to previous methods of estimating costs of software development. In this work, using genetic programming that a branch of evolutionary algorithms, a new algorithmic approach was presented for software development cost estimation. After the implementation of genetic programming and insert standard data, formula that was obtained that the mean, standard deviation and PRED(30) magnitude of relative error of this formula was better than previous algorithmic approaches such as COCOMO formulas.

In order to improve software development cost estimation other intelligent methods can be used such as genetic algorithms, fuzzy theory, neural networks and etc. also, using combine intelligent methods and genetic programming method better results and thus a better formula for software development cost estimation obtained.

REFERENCES

- [1] Sonia Manhas, Rajeev Vashisht, and ReetaBhardwaj (2010) "Framework for Evaluating Reusability of Procedure Oriented System using Metrics based Approach", International Journal of Computer Applications (0975 - 8887), Volume 9- No.10, November 2010.
- [2] L. Hareton and F. Zhang, "Software Cost Estimation ", Polytechnic University, Hong Kong, 2004
- [3] D. Hemer, "Specification-based retrieval strategies for component architectures", Proceedings of the 2005 Australian Software Engineering Conference (ASWEC'05), pp.233-242, 2005.
- [4] V. R. Basili, L. C. Briand, and W. L. Melo. A Validation of Component-Oriented Design Metrics as Quality Indicators *IEEE Transactions on Software Engineering*, 22(10):751- 761, Oct. 1996.
- [5] Ajay Kumar (2012) "measuring software reusability using svm based classifier approach", International Journal of Information Technology and Knowledge Management January-June 2012, Volume 5, No. 1, pp. 205-209.
- [6] H. Leung and F. Zhang, " Software Cost Estimation, Department of Computing", the Hong Kong Polytechnic University, Hong Kong, 1998
- [7] R. Valerdi, "Pioneers of Parametric", MIT, Cambridge, USA, 2007
- [8] B. Boehm and C. Abts, " Software Development Cost Estimation Approaches - A Survey", University of Southern California, Los Angeles, USA, 2002
- [9] Cost Estimating Software PRICE Systems, <http://www.pricesystems.com>
- [10] N. Merlo and M. Martin." COCOMO", Department of Computer Science, University of Zurich, Switzerland, 2003
- [11] F.G. Heetstra, "Software Cost Estimation", faculty of Public Administration and Public Policy, Twente University, POB 217, Enschede, Netherlands, 1992
- [12] SEER -Cost Estimating Software ,<http://www.gaseer.com>
- [13] B.Boehm, C. Abts, B. Clark, S.D.Chulani, E. Horowitz, R. Madachy, D. Refier, R. Selby and B. Stecce, " COCOMO II Model Definition manual", 1998
- [14] B.Boehm," COCOMO 2.0: Recent Development", USC COCOMO/SCM, 1994
- [15] Parvinder Singh Sandhu and Hardeep Singh, 2006,"Automatic Reusability Appraisal of Software Components using Neuro-Fuzzy Approach", International Journal Of Information Technology, vol. 3, no. 3, pp. 209- 214.
- [16] R. Poli, W.B. Langdon and N.F.McPhee, " A Field Guide to genetic Programming", University of Essex -UK, university of Minnesota, Morris -USA, 2008
- [17] I. Soute, " Genetic Programming", 2000
- [18] S.D.Lee, Y.J.Yang, E.S.Cho, S.D.Kim, S.Y.Rhew, "COMO: A UML- Based Component Development Methodology", IEEE, 1999.
- [19] <http://unbox.org/wisp/trunk/cocomo/data/coc81modetypel angtype.csv>
- [20] <http://unbox.org/wisp/trunk/cocomo/data/nasa93.csv>
- [21] H.K.Kim, Y.K.Chung, "Transforming a Legacy System into Components", Springer-Verlag Berlin Heidelberg, 2006.
- [22] Rajesh K Bhatia, Mayank Dave, R.C Joshi, "Retrieval of most relevant reusable Component using genetic algorithms", Software Engineering Research and Practice 2006, 151-155.
- [23] <http://unbox.org/wisp/var/dan/sandbox/cocomost/log/LC.crunch>
- [24] R. Kazman, S.G.Woods, S.J.Carril, "Requirements for Integrating Software Architecture and Reengineering Models: *CORUM II*", IEEE, 1998
- [25] Stender (1994) "Introduction to genetic algorithms", IEEE Colloquium on Genetic Algorithms, Volume 2, March 15, 1994 pp. 1-4.
- [26] Melanie Mitchell (1996) "An Introduction to Genetic Algorithm", MIT Press, 1996.
- [27] Esteva, J. C. and Reynolds, R. G. (1991) "Identifying Reusable Components using Induction", International Journal of Software Engineering and Knowledge Engineering, Vol. 1, No. 3 , 1991, pp. 271-292.
- [28] Caldiera, G. and Basili, V. R. (1991) "Identifying and Qualifying Reusable Software Components," IEEE Computer, February 1991.
- [29] Prof. KulwinderS.Mann and Amanpreet Singh, "A SVM Based Approach For Reusability Evaluation of Object Oriented Based Software Components", International Journal of Research in Engineering and Technology (IJRET) Vol. 1, No. 3, 2012 ISSN 2277 - 4378.
- [30] Brad Clark. "Calibration of COCOMO II.2003", 17th International Forum on COCOMO and Software Cost Modeling, [http://sunset.usc.edu/events/2002/cocomo17/Calibration%](http://sunset.usc.edu/events/2002/cocomo17/Calibration%20)

20fo%20COCOMO%20II.2003%20Presentation%20-%20Clark.pdf

- [31] Brad Clark. COCOMO II Database, CSE Annual Research Review, March 11, 2002
- [32] Basili, V.R., Briand, L.C., and Melo, W.L.: A validation of object oriented design metrics as quality indicators. IEEE Transactions on Software Engineering 22(10) (1996) 751-761
- [33] Khoshgoftaar T. M., Allen, E. B., Kalaichelvan, K. S., and Goel, N.: Early quality prediction: A case study in telecommunications, IEEE Software 13(1) (1996) 65-71.
- [34] Khoshgoftaar T. M., Pandya, A.S., and Lanning, D. L.: Application of neural networks for predicting faults, Annals of Software Engineering 1(1) (1995) 141-154

Authors' profiles



J Sirisha Devi was awarded B. Tech (Computer Science & Engineering) from Acharya Nagarjuna University -2003. She was awarded M. Tech in Computer Science from GITAM University, Visakhapatnam - 2010. At present she is working as Associate Professor in Computer Science and Engineering Department, JNTU

Hyderabad.



Dr. N. Murali Krishna was awarded B.Tech from Computer Science & Engineering department in the year 2002. He was awarded M. Tech in Computer Science & Engineering, in the year 2006. He was awarded doctorate in the year 2014. At present he is working as Assistant Professor in Computer Science and

Engineering Department, GIT, GITAM University, Visakhapatnam.

How to cite this paper: T.Tejaswini, J. Sirisha Devi, N. Murali Krishna, "Performance of Cost Assessment on Reusable Components for Software Development using Genetic Programming", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.9, pp.46-51, 2015. DOI: 10.5815/ijitcs.2015.09.07