

Parametric Estimation of Software Systems

Kavita Choudhary

Abstract— Software is characterised by software metrics. Calculation of effort estimation is a type of software metrics. Software effort estimation plays a vital role in the development of software. In recent years, software has become the most expensive component of computer system projects. The major part of cost of software development is due to the human-effort, and most cost estimation methods focus on this aspect and give estimates in terms of person-month. In this paper, estimation of effort required for the development of software project is calculated using genetic algorithm approach. Software systems are becoming complex and they desire for new, effective and optimized technique with limited resources. A solution to this problem lies in nature where complex species have evolved from simple organisms and constantly become able to adapt to changes in the environment. In case of species, it takes hundreds of generations and years which are not considerable in the field of software engineering. With the use of genetic algorithm, it can be done instantly by simulating the results on various tools of genetic algorithm.

Index Terms— Effort estimation, Effort estimation models, Walston-Felix model, COCOMO model, SEL model.

I. INTRODUCTION

Software effort estimation is one of the steps to be carried out in project planning. The effective and efficient development of software requires accurate estimates. The objective behind estimation is not only restricted to accuracy but also to control the cost and scope of the project from organization's point of view. The aim of this paper is to provide optimum results. Software researchers and practitioners are providing many effort estimation techniques for several decades but the problem persist in software engineering field. Early software estimation models are based on regression analysis or mathematical derivations. Today's models are based on simulation, neural network, genetic algorithm, soft computing, fuzzy logic modeling etc. The organization of this paper is as follows. Section II elaborates some literature reviews on software effort estimation model. Section III elucidates the analysis of effort estimation. Section IV describes the experimental results. Conclusion and potential future work are discussed in the Section V.

Manuscript received March 30, 2011.

Kavita Choudhary, CSE (PhD Pursuing), Banasthali University, Rajasthan, India. (e-mail: kavitaapunia@gmail.com).

II. RELATED WORK

This section provides some background information of various software effort estimation models to be used in this paper work.

SEL – Model: - The Software Engineering Laboratory (SEL)[2][3] of the University of Maryland has established a model i.e. SEL Model for estimation. Estimation of effort according to SEL model is as follows:-

$$E_{SEL} = 1.4 * (L)^{0.93}$$

Effort (E in Person-Months) and lines of code(L in thousands of lines of code i.e. KLOC) are used as predictors.

Walston-Felix Model: - The model developed by Walston and Felix[1][2] at IBM provides a relationship between delivered lines of source code (L in thousands of lines) and effort E (E in person-month). This model constitutes various aspects of the software development environment such as user participation, customer-oriented changes, memory constraints etc. According to Walston and Felix model, effort is computed by:-

$$E_{W-F} = 5.2 * (L)^{0.91}$$

COCOMO Basic Model: - COCOMO[4][5][7] model is proposed by B.W.Boehm. COCOMO model have three sub-models i.e. basic, intermediate and detailed model. Basic model takes the form

$$E = a_b * (L)^b$$

E is effort applied in person-month. The values of coefficients a_b and b_b are defined by Boehm. The basic model aims at estimating in a quick and rough fashion, most of the small to medium sized software projects. Three modes of software development are considered in this model: organic, semi-detached and embedded. The calculations of effort according to three different modes are as follows:-

Basic Model - Organic Mode

$$E_{B-O} = 2.4 * (L)^{1.05}$$

Basic Model – Semi-detached Mode

$$E_{B-S} = 3.0 * (L)^{1.12}$$

Basic Model - Embedded Mode

$$E_{B-E} = 3.6 * (L)^{1.20}$$

COCOMO Intermediate Model: - In this model, Boehm introduced an additional set of 15 predictors called cost drivers in the intermediate model to take account of the software development environment. Cost drivers are used to adjust the nominal cost of a project to the actual project environment, hence increasing the accuracy of the estimate.

The multiplying factors for all 15 cost drivers are multiplied to get the effort adjustment factor (EAF). Typical values for EAF range from 0.9 to 1.4. The intermediate COCOMO considering three modes takes the form:-

Intermediate – Organic Mode

$$E_{I-O} = 3.2 * (L)^{1.05} * EAF$$

Intermediate – Semi-detached Mode

$$E_{I-S} = 3.0 * (L)^{1.12} * EAF$$

Intermediate – Embedded Mode

$$E_{I-E} = 2.8 * (L)^{1.20} * EAF$$

III. ANALYSIS OF EFFORT ESTIMATION USING GENETIC ALGORITHM MODEL

The procedural analysis of proposed effort estimation model will be carried out in two stages i.e. Conceptual View and Data Analysis. Details on each stage are described below:-

TABLE-I

SN	Model	L (Size)	Equation	Actual Effort	GA Effort
1	SEL Model	400	ESEL=1.4*(L)0.93	368.1661	257.299
2	Walston-Felix	400	EW-F=5.2*(L)0.91	1213.0479	899.1549
3	Basic-organic	40	EB-O=2.4*(L)1.05	115.4448	51.3607
4	Basic-semidetached	250	EB-S=3.0*(L)1.12	1454.8361	876.7732
5	Basic-embedded	400	EB-O=3.6*(L)1.20	4772.8138	3279.114
6	Intermediate-organic	40	EI-O = 3.2*(L)1.05 * EAF	143.8904	46.0347
7	Inter- semidetached	250	EI-S = 3.0*(L)1.12 * EAF	1359.9808	1094.761
8	Inter.-embedded	400	EI-E = 2.8*(L)1.20 * EAF	3470.1538	2559.478

A. Conceptual View

In this paper, various software effort estimation [13][16] models are considered and some interesting data is calculated by subsequently varying size of software i.e. in KLOC (Kilo Lines of Code). Values of actual effort, effort using genetic algorithm, magnitude of relative error (MRE) and mean of magnitude of relative error are computed. MRE [3][11][15] is computed by following:-

$$MRE = (EffortActual - EffortGA) / Effortactual \%$$

In this work, Initial Range[3][4] of Population is taken for various estimation models in GA effort calculation , like 100-101 for 100 KLOC , 400-404 for 400 KLOC,1000-1010 for 1000 KLOC, etc. Initial range is to be given to make individuals that are best suited according to fitness function to participate in crossover and mutation for optimum results[3][4]. At first, SEL model is considered with a range of 10 KLOC to 100000 KLOC. From the calculated data it is clear that by increasing size of project in KLOC, the effective MRE i.e. Magnitude of Relative Error decreases. For small size project the GA effort is optimum comparative to actual effort but MRE is not coming appropriate. After that, Mean of Magnitude of Relative Error i.e. MMRE is computed. The value of MMRE of SEL Model, COCOMO Intermediate – Embedded and Walston-Felix Model is at lower edge. In SEL Model analysis, as we increase the size of the software project, the corresponding magnitude of relative error decreases. At large size the magnitude of relative error becomes negligible. Next model is Walston-Felix Model. The effort calculated by Walston-Felix equation is three times larger than effort calculated according to SEL Model. Here

also, while constantly increasing the size, the MRE corresponding to actual effort and GA effort also declines and becomes negligible at very huge size. After that, basic-organic model is evaluated. Here, it is noticed that when size is small then trend of MRE is not appropriate. In basic-semidetached, the decline trend is visible. In basic-embedded model, as the size increases then the difference between actual effort and GA effort becomes almost negligible. Again the same problem of basic-organic is encountered in intermediate-organic. The range of size is small i.e. 0 to 50 KLOC, the trend is not recorded accurately. In Intermediate-Semidetached model, the value of MRE decreases as we increase the size of software. In Intermediate-Embedded, the value of MRE is calculated appropriately as range for size is vast.

B. Data Analysis

This section is classified into two phases.

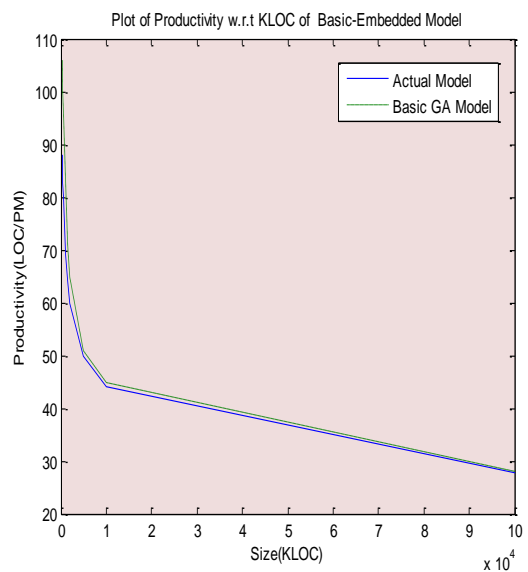
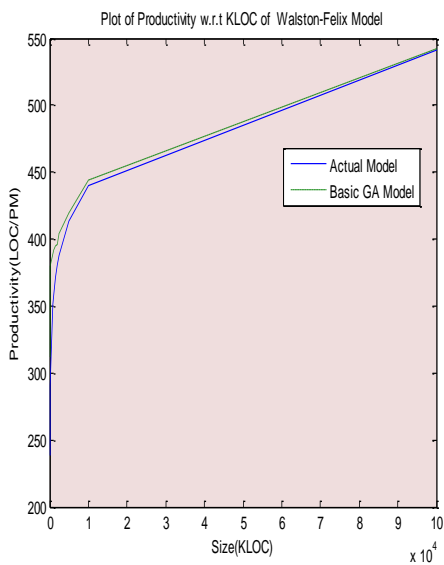
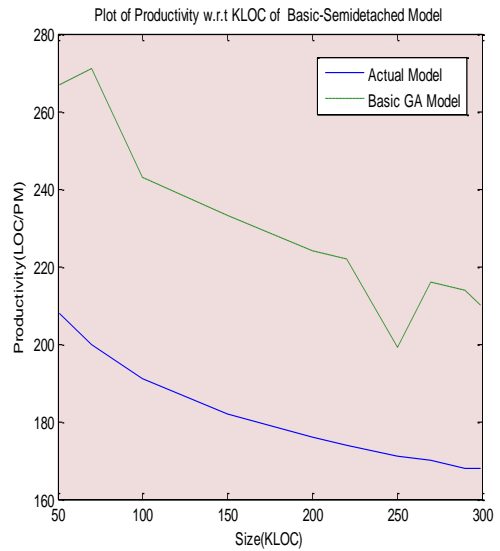
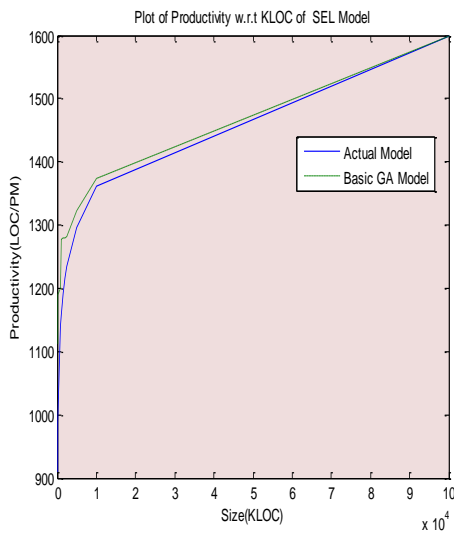
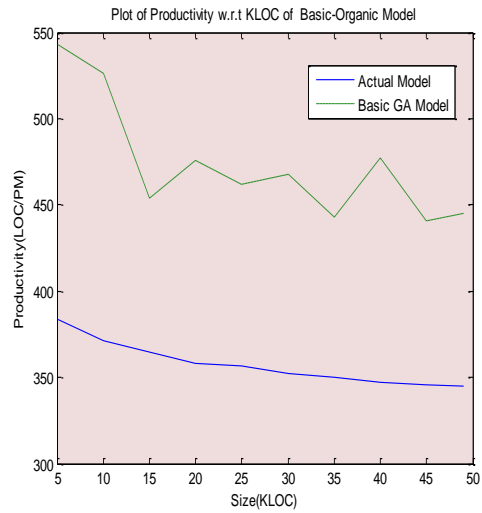
Phase I :- In SEL model, it is noticed that as size of software increases, the difference between productivity with respect to actual effort and GA effort becomes negligible at huge size of software. Effort is inversely proportional to productivity. Next is Walston-Felix model, when compared with SEL model, at same size the productivity is less in Walston-Felix model. In this model also, the productivity with respect to

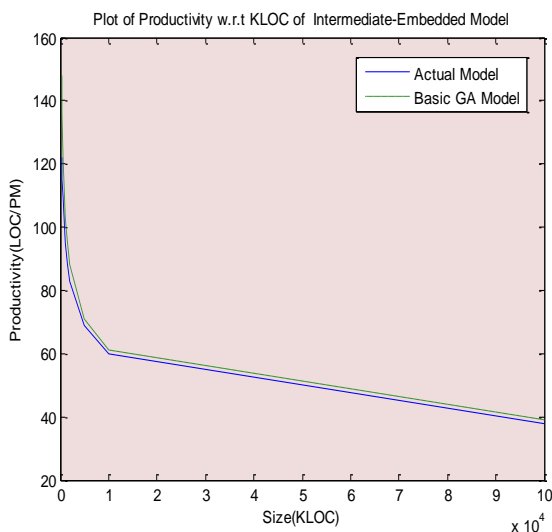
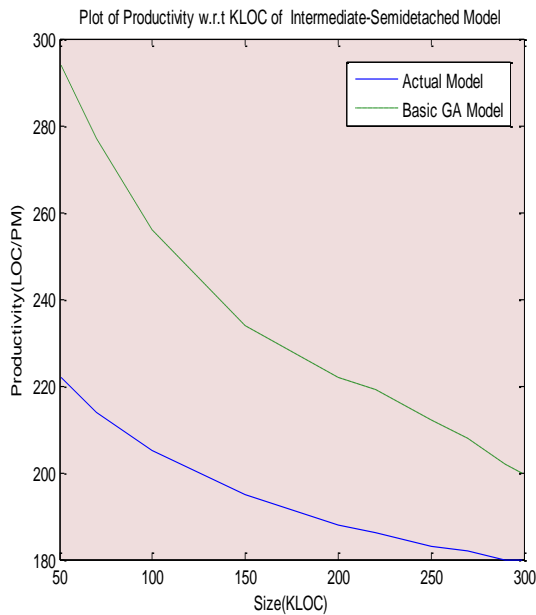
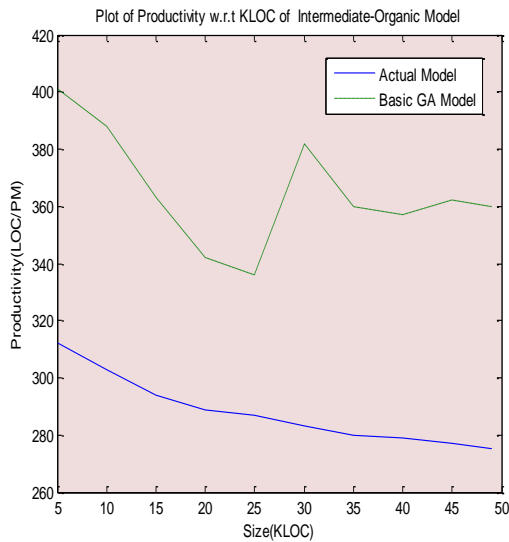
actual effort and GA effort become almost same at largest size. In basic-organic, trend of productivity is not accurate because the range for size is small enough i.e. 0 to 50 KLOC in case of organic model. In Intermediate-organic the same problem persists. In case of basic-semidetached, basic-embedded, intermediate -semidetached and intermediate -embedded models, the trends are evaluated.

Phase II :- The above graph represents productivity with respect to KLOC. The first graph shows productivity of SEL Model calculated by actual effort and size and also productivity calculated by GA effort and size. Both gradually increase after a point. As size increases the productivity calculated by actual effort and productivity calculated by GA effort becomes almost same. The second graph shows productivity of Walston-Felix Model calculated by actual effort and size and also productivity calculated by GA effort and size. Both of them gradually increase after a point. As size increases the productivity calculated by actual effort and productivity calculated by GA effort becomes almost same. The third graph shows productivity of Basic-Organic Model calculated by actual effort and size and also productivity calculated by GA effort and size. Both of them decline. Productivity with respect to GA effort has zigzag cuts. The fourth graph shows productivity of Basic-Semidetached Model calculated by actual effort and size and also productivity calculated by GA effort and size. Both of them are inclined to decline. The fifth graph shows productivity of Basic-Embedded Model calculated by actual effort and size and also productivity calculated by GA effort and size. Both of them have straight decline. As size increases the productivity calculated by actual effort and productivity calculated by GA effort becomes almost same. The sixth graph shows productivity of Intermediate-

Organic Model calculated by actual effort and size and also productivity calculated by GA effort and size. Both of them decline. Productivity with respect to GA effort has zigzag cuts. The seventh graph shows productivity of Intermediate-Semidetached Model calculated by actual effort and size and also productivity calculated by GA effort and size. Both of them decline. Productivity with respect to GA effort has straight decline. The eighth graph shows productivity of Intermediate-Embedded Model calculated by actual effort and size and also productivity calculated by GA effort and size. Productivity with respect to GA effort shows intense decline. As size increases the productivity calculated by actual effort and productivity calculated by GA effort becomes almost same.

IV. EXPERIMENTAL RESULTS





V. CONCLUSION

This work explores the inter-relationship among different dimensions of software projects namely models, project size

and effort. All the above graphs are representing productivity in LOC/PM w.r.t. size in KLOC of various estimation models by varying sizes. This shows the inter-relationship between size in KLOC, effort, duration and productivity. The future scope of this work is to calculate Average-staff manning size can also be calculated by this data. Average staff manning size is equal to effort/duration.

REFERENCES

- [1] Yeong-Seok Seo, Kyung-A Yoon, Doo-Hwan Bae, An Empirical analysis of software effort estimation with outlier elimination, Proceedings of the 4th International workshop on Predictor models in Software Engineering © ACM 2008 (ISBN: 978-1-60558-036-4/08).
- [2] Pichai Jodpimai, Peraphon, Advanced Virtual and Intelligent Computing Centre , Analysis of Effort Estimation based on Software Project Models © IEEE 2009.
- [3] Parvinder S. Sandhu, Manisha Prashar, Pourush Bassi and Atul Bisht , A model for estimation of efforts in development of software systems, World Academy of Science and Technology 56-2009.
- [4] Alaa F Sheta, Estimation of the COCOMO model parameters using genetic algorithm for NASA Software Projects, Journal of Computer Science 2006 ISSN: 1549-3636 © 2006 Science Publications.
- [5] Andreea Fanea and Laura Diosan, Components execution order using Genetic Algorithm, STUDIA UNIV. BABES BOLYAI, INFORMATICA, 2005.
- [6] Westley Weimer, ThanhVu Nguyen, Claire Le Goues and Stephanie Forrest, Automatically Finding Patches Using Genetic Programming, ICSE'09, © IEEE.
- [7] Mark Harman, The Current State and Future of Search Based Software Engineering, Future of Software Engineering (FOSE'07) © 2007 IEEE.
- [8] Indrajit Bhattacharyya, Vector GA - A Novel Enhancement of Genetic Algorithms for Efficient Multi-variable or Multi-dimensional Search, November 2009 Volume 34 Number 6, ACM.
- [9] Anne Martens, Optimising Multiple Quality Criteria of Service-Oriented Software Architectures, QUASSOS'09, August 25, 2009, ACM Publications.
- [10] Yujie Yao and Haopeng Chen, QoS-aware Service Composition Using NSGA, ICIS 2009, November 24-26, 2009 Seoul, Korea © 2009 ACM 978-1-60558-710.
- [11] Hsin-yi Jiang, Carl Chang, Deriving Evaluation Metrics for Applicability of Genetic Algorithms to Optimization Problems, GECCO'08, ACM 978-1-60558-130.
- [12] Outi Raiha, Applying Genetic Algorithms in Software Architecture Design, University of Tampere Department of Computer Sciences, February 2008.
- [13] Wasif Afzal, Richard Torkar, Suitability of genetic programming for software reliability growth modeling, Blekinge Institute of Technology.
- [14] John R. Koza, Introduction to Genetic Programming, GECCO-2008—ATLANTA JULY 12–16, 2008.
- [15] Hsin-yi Jiang, Can the Genetic Algorithm Be a Good Tool for Software Engineering Searching Problems? Compsac, vol. 2, pp.362-366, 30th Annual International Computer Software and Applications Conference (COMPSAC'06), 2006.
- [16] Kavita Choudhary GA based Optimization of Software Development Effort Estimation (ISSN No-0976-8491), Vol 1 Issue 1 (IJCSST 0910/963/229), Dec 2010.



Kavita Choudhary working as Assistant Professor in Department of Computer Science in ITM University, Gurgaon. Currently, she is pursuing Phd. from Banasthali University. She did her Post graduation with distinction from Guru Gobind Singh Indraprastha University. She did her graduation, B.Tech (CSE); in honors from University of Rajasthan. She have more than five year of teaching experience. Her core subjects are Software Engineering, Computer Networks, and Software Testing.