Use of Computational Intelligence Technique for Accuracy Enhancement in Software Cost Estimation

Aruna Shrivastava[1], Neelabh Sao[2]

[1] M.Tech Student, Department of Computer Science & Engineering, Rungta College of Engineering & Technology, Bhilai, Chhattisgarh

[2] Reader, Department of Computer Science & Engineering, Rungta College of Engineering & Technology, Bhilai, Chhattisgarh

---***---

Abstract - ***For any software organization three fundamental and essential requirements are- delivering a software product on time, with an agreed level of quality and within budget. Both underestimation and overestimation of software cost may cause adverse effect on s/w quality, budget, schedule, company's business reputation and competitiveness. Thus with the hope of managing project well within budget and schedule, a myriad of cost estimation models have been proposed. This paper proposes a new Computational Intelligence based approach that encompasses merits of 3 techniques namely Particle Swarm Optimization, Analogy Based Estimation, and Genetic Programming. This hybrid approach can effectively enhance the accuracy and prediction power of estimation model.***

Key Words: *Computational Intelligence, Particle Swarm Optimization, COCOMO, Cost Drivers.*

1. INTRODUCTION

Now a day's software industry is getting more seasoned & complex because the size and importance of software applications have grown a lot. Without a doubt it is now the driving force of industry area, government & military operations, modern businesses, scientific, medical & technical fields. Precise software estimation provides good support for the decision-making process, for better analysis of the project and efficiently managing the software development process the accurate assessment of costs is very essential, Accurate prediction is still subjective to research, numerous techniques and models have been proposed which can be broadly classified into two categories namely ALGORITHMIC AND NON ALGORITHMIC [7].ALGORITHMIC TECHNIQUES uses mathematical equations to perform software estimations. These mathematical formulae relates independent variables (like cost drivers) to dependent variables (like effort, cost) COCOMO, SLIM, Walston Felix Model, Dotty model etc make use of algorithmic approach of estimation. On the other hand NON ALGORITHMIC TECHNIQUES includes analogy based, expertise based and soft computing techniques.[6]

1.1 *Computational Intelligence in Software Cost Estimation*

CI can be defined as a study of adaptive mechanisms to enable or facilitate intelligent Behaviors in complicated, uncertain, and changing environments. Computational intelligence (CI) is a collection of optimization methodologies - neural networks, fuzzy systems (FS),evolutionary computation (EC), and swarm theory which is used synergistically to avoid the nonlinearity, high complexity, and unpredictability pertaining to SCE. There are some limitations in algorithmic models obtain during the early stage of a software development project these model require as inputs, accurate estimate of certain attribute such as LOC, Function points Complexity and so on which are difficult to obtain. Parametric models are not precisely able to handle categorical data also lack reasoning capabilities, These models also have difficulty in modeling the inherent complex relationships between the contributing factors,. The limitations of algorithmic models led to the exploration of the non algorithmic techniques which are soft computing based. This paper proposes a hybrid approach that encompasses 3 CI techniques namely PARTICLE SWARM OPTIMIZATION (PSO), ANALOGY BASED ESTIMATION (ABE) & GENETIC PROGRAMMING APPROACH (GA).

PARTICLE SWARM OPTIMIZATION – In 1995 based on social behaviors of birds flocking or fish schooling Eberhart and Kennedy discovered a new approach, Particle swarm optimization (PSO) which is biologically inspired computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality [8]. Since they make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions thus these methods are commonly known as Meta Heuristics. PSO learns from a scenario and uses it to solve optimization problems. In PSO, each single potential solution is a ''bird'' which is termed as a particle in the search space. Each particle consists of velocity and fitness value. Velocity directs the flight of the particle and fitness values are evaluated by the fitness function to be optimized. In the problem space these particles fly and follow the current optimum particles. Initialization of PSO is done by a group of random particles (solutions) and then searches for optima by updating generations. In any iteration, updation of each particle is done by following the two ''best'' values. Pbest and Gbest. The first one is the best solution (fitness) it has achieved so far (the fitness value is also stored).while the other ''best'' value is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the population. This best value is a global best and called Gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest. The modification of the particle's positions is required because each particle accelerate towards its Pbest and Gbest position. By making use of following numerical equations purpose is achieved

V(p,i) = chi*(V(p,i) + C1 * (pBestPosition(p,i)-R(p,i)) + C2 * (gBestPosition(i) - R(p,i))); (1)

R(p,i) = R(p,i) + V(p,i) (2)

Here V(p,i) is particle velocity, (p,i) is current solution position, chi is weighting factor, Rand() are uniformly distributed random numbers between 0 and 1.C1,C2 are cognitive parameter and social parameter respectively.

ANALOGY BASED ESTIMATION – This approach compares new projects with similar projects from the past, make relationship and find similarity in order to find accurate result. Historical knowledge of previous similar projects is very helpful in estimation process.

GENETIC PROGRAMMING- One of the evolutionary methods for the effort estimation is Genetic Algorithm. In this approach solution is achieved by means of a cycle of generations of candidate solutions that are reduced by criteria, survival of the fittest. Genetic Programming (GP) overcomes problem of local optima which is encountered in PSO & other evolutionary techniques.

1.2 *Purpose of Proposed Hybrid Approach*

The purpose of this hybrid approach is to tune the effort parameter of COCOMO model. The equation of effort in terms of size is considered as follows:

Effort= a * (Size) ^{*b}

Where a, b, are constant. Based on the development mode namely- Organic mode, Semi-detached mode and embedded mode projects are classified and Bohem tabulated different values of co efficient for these different mode [2]

Software project	A	B
Organic	2.4	1.05
Semi-Detached	3.0	1.12
Embedded	3.6	1.20

The proposed model with the help of PSO, ABE & GA tune these effort parameter in order to gain more accuracy in calculation.

1.3 Dataset Description

To explore the real performance, the evaluation of the estimation model must be carried out by applying real data sets. In this study COCOMO 81 dataset which includes 63 projects is considered to investigate the accuracy of the proposed model. Number of attributes are 17,in which 15 attributes are for effort multipliers,1 for size of project in terms of LOC and 1for actual effort.

2. PROPOSED METHODOLOGY

Particle swarm optimization is a population based stochastic optimization technique, each potential solution is termed as particle and finding the optimal solution is its aim. I/P – size of s/w projects, measured efforts, EAF.O/P- optimized parameter for estimating effort

STEP 1 –initialize n particles with random position & velocity

STEP 2–initialize weight function value W= 0.25 with Weighting parameter cognitive learning factor C1= 2.05, C2= 2.05

STEP 3 – repeat step 4 to 9 until number of iteration exhaust

STEP 4 – For i = 1,2,..n i.e. 63 do// for all particles
Evaluate fitness function using following formula
est(i) = (pop(i,1) * res(16) ^ pop(i,2)) * eaf ;
end
fit = abs(est - res(17));

STEP 5 – Determine Pbest for each particle by evaluating & comparing measured effort & estimated effort values of current & previous parameter values.

STEP 6 – **S**et best of Pbests as global best – Gbest

STEP 7 – Update weight function using following eqn
chi = 2.0/abs(2.0-phi-sqrt(phi^2-4*phi));

STEP 8 – Check for crossover case

STEP 9 – Update velocity & position of tuning parameters with following equation

V(p,i) = chi*(V(p,i) + C1 * (pBestPosition(p,i)-R(p,i)) + C2 * (gBestPosition(i) - R(p,i))); (3)

R(p,i) = R(p,i) + V(p,i) (4)

STEP 10 – Give Gbest values as the optimal solution

STEP 11 – Stop.

3. RESULT

In the very first stage of experiment, initial population of 10 individuals was generated, after that optimization of COCOMO model coefficient was performed using the proposed algorithm. This hybrid approach gives following output

- 63 values of optimized parameter i.e.63 Global best values [fig 1]
- Individual best fitness value for each 63 project [Fig. 2]
- Individual best population i.e. 63 specific optimized a, b, values for 63 projects [Fig.3]
- 1 Best fitness value
- 1best population i.e. a, b, which gives optimal solution among 10 global best solution.
- The Proposed model is enhancing the accuracy of estimation model with the efficiency of 98.3863%

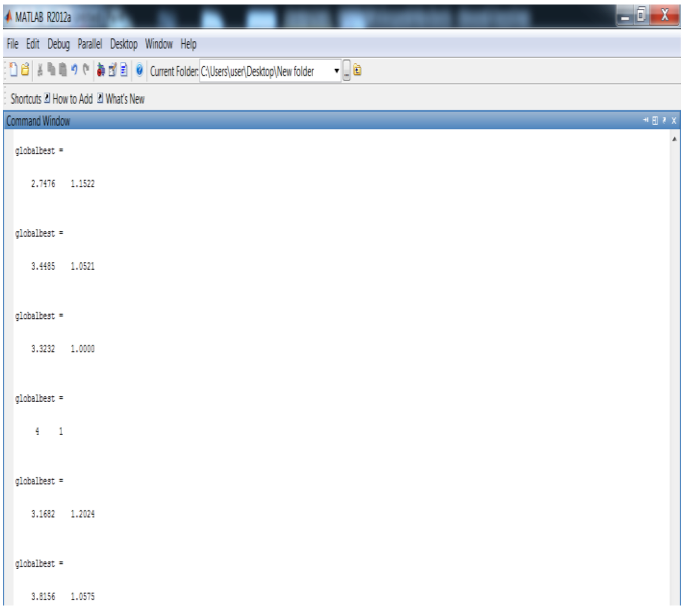

Fig -1: 63 Global best values

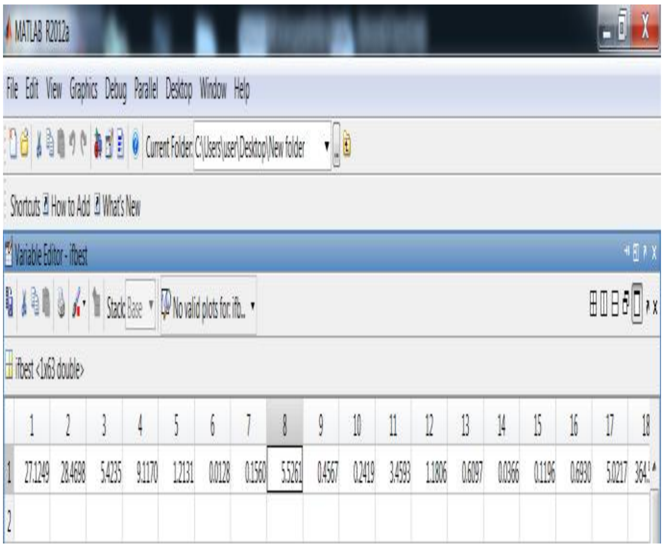

Fig -2: 63 Individual best fitness values

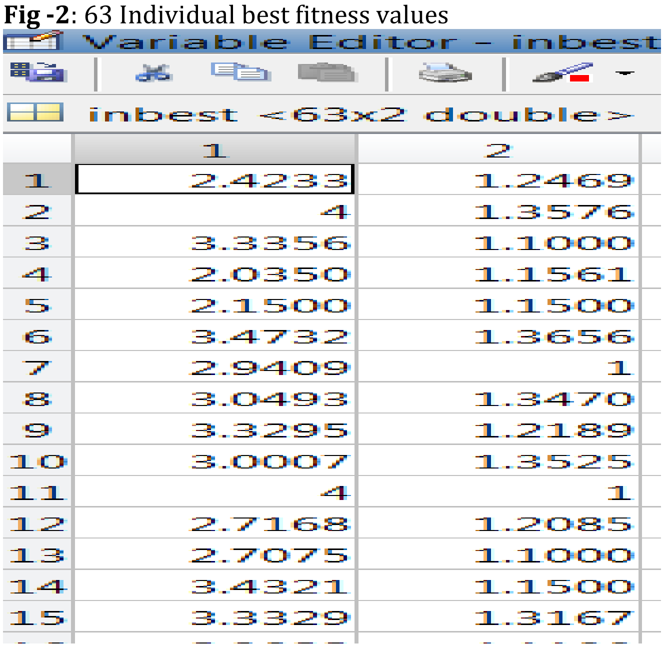

Fig -3: Individual best population i.e. 63 specific optimized a, b, values for 63 projects inbest <63x2 double>

	1	2
16	2.3022	1.1100
17	2	1
18	3.4452	1.2050
19	3.0404	1.1353
20	3.8382	1.1000
21	3	1.2381
22	2	1.2436
23	3.3321	1.2000
24	3.8201	1.1500
25	3.0075	1.3657
26	3.7182	1.1962
27	2.1282	1.4137
28	2.2667	1.0500
29	3.1156	1
30	2.4559	1.2902
31	3.5133	1.1120
32	3.1267	1
33	2.5000	1.2535

Fig -4: Individual best population i.e. 63 specific optimized a, b, values for 63 projects

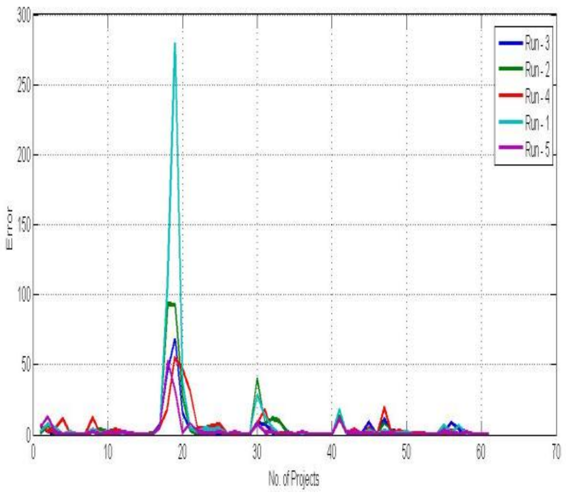

Fig.5 Measured Effort v/s Estimated Effort

4. CONCLUSIONS

The Estimation of software cost is based on a probabilistic model and hence it does not generate exact values. Still if a systematic technique is employed and good historical data is provided and much better results can be achieved. In terms of its error rate accuracy of the model is measured and it is desirable to be as close to the actual values as possible. This study highlighted a new hybrid model to estimate the software effort. It is observed that combination of computational techniques provides precise optima solution efficiently and enhances the accuracy of estimation model with the efficiency of 98.3863%.

REFERENCES

[1] Allan J. Alberecht and John E. Gaffhey, November 1983, Software Function, Source Lines of Code and Development Effort Prediction : A software Science Validation. IEEE transactions on Software Engineering

[2] Boehm, 1981 “Software Engineering Economics”, Prentice Hall

[3] J. S. Pahariya, V. Ravi, M. Carr, M. Vasu, “Computational Intelligence Hybrids Applied to Software Cost Estimation”, International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM),Vol. 2, pp. 104-112, 2010.

[4] M. V. Deshpande, S. G. Bhirud. August 2010. Analysis of Combining Software Estimation Techniques. International Journal of Computer Applications

[5] Prasad Reddy.P.V.G.D & CH.V.M.K.Hari, “Software effort estimation using Particle Swarm Optimization with intertia weight”

[6] Parvinder S. Sandhu, Porush Bassi, and Amanpreet Singh Brar. 2008. Software Effort Estimation Using Soft Computing Techniques. World Academy of Science, Engineering and Technology 46 2008

[7] R. Sweta Kumari and Shashank Pushkar, “ Comparison and Analysis of Different Software Cost Estimation Methods”, International Journal of Advanced Computer Science and Applications, Vol. 4, No.1, 2013.

[8] Vahid Khatibi Bardsiri • Dayang Norhayati Abang Jawawi •Siti Zaiton Mohd Hashim • Elham Khatibi , “A PSO-based model to increase the accuracy of software development effort estimation” Published online: 22 September 2012 Springer Science+Business Media, LLC 2012

Ms.Aruna Shrivastava is currently pursuing M.Tech in Computer Science and Engineering with specialization in Software Engineering from CSVTU, Bhilai (C.G.) Her research area includes field of Software Engineering and Data Mining.