Tuning of COCOMO II Model Parameters for Estimating Software Development Effort using GA for PROMISE Project Data Set

Chandra Shekhar Yadav
Department of Computer Science & Engineering
Noida Institute of Engineering & Technology
19, Knowledge Park II Greater Noida -201306

Raghuraj Singh
Department of Computer Science & Engineering
Harcourt Butler Technological Institute
Nawab Ganj Kanpur-208002

ABSTRACT
In this paper, we have tuned the parameters of COCOMO II model to estimate the software development effort using genetic algorithm (GA). Results obtained by applying GA are have been compared with results obtained by applying particle swarm optimization (PSO) published in previous paper. COCOMO II model is modified by introducing some more parameters to predict the software development effort more precisely. The performance of this parametric model is tested on the past PROMISE and NASA projects data set.

General Terms
Genetic Algorithm, Particle Swarm Optimization

Keywords
COCOMO81 model, Root Mean Square Error, PROMISE Software Repository data set, Software Development Estimation.

1. INTRODUCTION
Software development effort has been estimated using parametric COCOMO model in terms of person-months. In the basic COCOMO Model development effort is linearly dependent on software size (kloc). There are fifteen multipliers which affect the software development effort. These parameters are analyst capability (acap), programmer's capability (pcab), application experience (aexp), modern programming practices (modp), use of software tools (tool), virtual memory experience (vexp), language experience (lexp), schedule constraint (sced), main memory constraint (stor), database size (data), time constraint for CPU (time), turn-around time (turn), machine volatility (virt), process complexity (cplx) and required software reliability (rely). Increasing the value of multipliers $acap, pcab, aexp, modp, tool, vexp, lexp$ causes decrease in development effort and decreasing the value of multipliers $stor, data, time, turn, virt, cplx, rely$ causes increase in development effort. By estimating the development effort precisely, resources can be utilized efficiently and effectively. GA can be explored to build efficient parametric model.

2. RELATED WORKS
Alaa F. Sheta (2006) modified COCOMO model to explore the effect of the software development effort computation [1]. Alaa F. Sheta et al. (2010) proposed in [2] effort estimation model utilizing lines of code and methodology with the help of GA. Alaa F. Sheta et al. (1996) estimated parameters of nonlinear systems in noisy environment using GA [3]. Anil Kumar et al. (2012) tuned parameters of COCOMO Model using PSO [4]. K.K. Shukla (2000) developed in [5] neuro-genetic predictor for estimating software development effort. Sehra et al. (2011) estimated software project effort using soft computing technique [7]. Sultaan Aljahadali et al. (2010) tuned COCOMO Model parameters using differential evolutions to estimate software development effort [8]. Chandra Shekhar et al. analyzed the reliability of object oriented system using vague lambda-tau modeling which can be used in effort estimation [11]. Idri et al. also used in [12] soft computing for software cost estimation. Hakutta et al. and Shepper et al. proposed software estimation model using analogies [13-14]. H. Garg et al. used in [15] artificial bee colony based lambda-tau technique with weibull fuzzy distribution function for analyzing the behavior of the pulping unit in a paper mill. A. Doostparast Torshizi et al. (2010) adopted Fuzzy Approach for failure analysis using Petri nets [16]. Naveen Kumar et al. (2011) has done reliability analysis of waste clean- up manipulator using real coded genetic algorithms and Fuzzy Lambda Tau Methodology [17]. Somesh Kumar et al. (2013) used evolutionary algorithm with Multi Layer Feed Forward Neural Network for the classification of hand written Hindi "SWARS" [18]. Baily, J. W. et al. proposed in [19] a meta-model for estimating the expenditure of resources used in software development.

The rest of the paper is organized as follows. In section 3 COCOMO model has been described. In Section 4 Genetic Algorithm has been explained. Section 5 gives idea of particle swarm optimization. Section 6 gives result and discussion. Finally, section 7 gives conclusion and future scope of work.

3. COCOMO MODEL
Barry Boehm et al. (1981) in [6] proposed Constructive Cost Estimation Model (COCOMO) for estimating development effort using following formula

$$\text{Effort} = \text{a}(\text{kloc})^{\text{b}} \quad (1)$$

Here a and b are project specific parameters

Multipliers can be added in the above formula for the complex project as follows

$$\text{Effort} = \text{a} * \text{kloc}^{\text{b}} * \text{p} \quad (2)$$

Here p is product of $acap, pcap, aexp, modp, tool, vexp,$

$lexp, sced, stor, data, time, turn, virt, cplx, rely$

4. GENETIC ALGORITHM
According to John Holland (1975) GA is an evolutionary algorithm based on binary or real valued string representations [9]. By applying crossover, mutation operators along with different selection strategies such as roulette-wheel, rank, elitist, and tournament over the string, generation of other strings have been evolved in order to solve complex

optimization problems. Here fitness function is represented in terms of root mean square error(rmse).

$$\mathrm{rmse} = \mathrm{sqrt}\left(\frac{1}{Q}\sum_{k=1}^{Q} \xi_k\right) \quad (3)$$

$$\xi_k = (\text{actual effort} - \text{estimated effort})^2 \quad (4)$$

Here Q is the number of projects

The objective of the problem is to optimize the root mean square error of PROMISE project data set [10].

5. PARTICLE SWARM OPTIMIZATION

Kennedy et al. (1995) introduced in [20] particle swarm optimization algorithm to solve the complex problem in terms of social and cognitive behavior. Here individual is called particle. Each particle has a possible solution in multidimensional search space. The movement of particle is effected by information from iteration to iteration and from particle to particle. There are two types of solutions such as p_{best} and g_{best} in this optimization algorithm. The local optimal solution is called p_{best} recorded by each particle in its own path. The global optimal solution is called g_{best} obtained by particle to particle interaction. The i^{th} particle of the swarm is represented by n dimensional vector $X_i = (x_{i1}, x_{i2} \dots x_{in})^T$. The velocity of i^{th} particle is shown by vector $V_i = (v_{i1}, v_{i2}, \dots v_{in})^T$. The previously visited best solution of i^{th} particle is represented by $P_i = (p_{i1}, p_{i2}, \dots p_{in})^T$. The velocity of i^{th} particle is updated by the following equation.

$$v_{id} = v_{id} + c_1 * \mathrm{rand}(r_1)(p_{id} - x_{id}) + c_2 * \mathrm{rand}(r_2)(p_{gd} - x_{id}) \quad (5)$$

The position of i^{th} particle is updated by using the following equation.

$$x_{id} = x_{id} + v_{id} \quad (6)$$

$Here\ d = 1,2, \dots n; i = 1,2, \dots S. S\ is\ swarm\ size. c_1\ and$

$c_2\ are\ called\ cognitive\ and\ social\ parameters.$

$r_1\ and\ r_2\ are\ random\ numbers.$

Kennedy et al. in [21] included an inertia weight (w) in the velocity updating equation (5) as shown in equation (7).

$$v_{id} = w * v_{id} + c_1 * rand(r_1)(p_{id} - x_{id}) + c_2 * rand(r_2)(p_{gd} - x_{id}) \quad (7)$$

6. RESULT AND DISCUSSION

Experiments have been conducted on the PROMISE project data sets to optimize the value of root mean square error (rmse) of all the 40 projects. Root mean square error is 8.8648 by using COCOMO II Model represented in equation (2). The values of project specific parameters a and b are 2.564 and 0.862 respectively by using the control parameters of GA mentioned in table 1. Anil et al (2012) in [4] represented root mean square error as 8.894343 and the values of a and b as 2.727945 and 0.862 respectively using PSO. Here, as shown in figure 1, the results obtained using GA are almost same as that obtained by using PSO.

Model 1: By modifying the COCOMO Model as given in the following equation (8) root mean square error has been found on the normalized PROMISE project data set as 0.0423128. The values of the parameters a, b, c and d are 1.392, 1.241, 1.081 and 0.014 respectively.

Best Fitness value of Model 1 is represented in fig 2 using control parameters of GA mentioned in table 2.

$$\mathrm{Effort} = a * (\mathrm{kloc})^b * c * (p)^d \quad (8)$$

Best fitness value of model 2 is represented in fig. 3 obtained by using control parameters mentioned in table 3.

Model 2: Again by modifying the COCOMO Model as given in the equation (9) root mean square error has been found on the normalized PROMISE project data set as 0.042080208. The values of parameters a, b, c and d are 1.355, 1.315, -0.228 and 0.305 respectively. Using this model only a little improvement occurs in the root mean square error of 40 PROMISE projects.

$$\mathrm{Effort} = a * (\mathrm{kloc})^b + c * (p)^d \quad (9)$$

Model 3: Again by modifying the COCOMO Model as given in the equation (10) root mean square error has been found on the same data set as 0.0247382333. The values of $a, b, c, d, e, f, g,$ and h are 1.159, 1.224, 1.191, 0.287, 0.812, -0.38, and 1.176 respectively.

$$\mathrm{Effort} = a * (\mathrm{kloc})^b + c * (p_1)^d + e * (p_2)^f + g * (\mathrm{sced})^h \quad (10)$$

$Here\ p_1 = \mathrm{acap} * \mathrm{pcab} * \mathrm{aexp} * \mathrm{modp} * \mathrm{tool} * \mathrm{vexp} * \mathrm{lexp}$

$and\ p_2 = \mathrm{rely} * \mathrm{data} * \mathrm{cplx} * \mathrm{time} * \mathrm{store} * \mathrm{virt} * \mathrm{turn}$

Best fitness value of model 3 is represented in fig 4.

From the above results obtained by using control parameters mentioned in table 4, model 3 has been found suitable for predicting the software development effort more precisely in comparison to the other parametric models.

7. CONCLUSIONS AND FUTURE SCOPE

In this paper by applying the GA, suitable parametric model has been found to predict the software development effort accurately. Accurate prediction of development effort helps software professional to manage the resources of the organization. In future multilayer feed forward network can be used to predict the development effort. Combination of neural network and GA can also be used to develop predictor for predicting software development effort more precisely than the existing parametric models. In place of GA, PSO can be used for developing predictor. Since nothing is certain in the real world, so fuzzy set or vague set can be used in place of crisp set for estimating the software development effort.

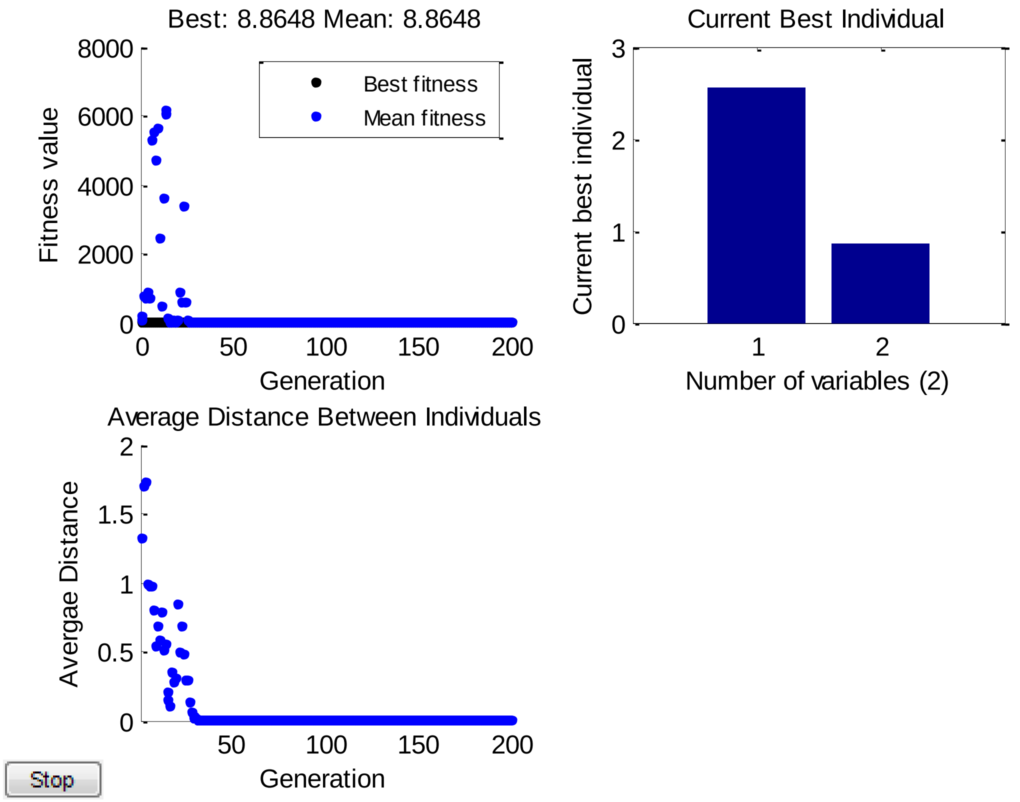

Fig 1: Best fitness value of $rmse$ and the values of project specific parameters a and b for COCOMO

Table 1: Control parameters of GA to tune the parameters a and b of COCOMO

Parameters	**Value**
Population Size	5000
Elite Count	4
Crossover Fraction	0.7000
Generations	200
Initial Population	[5000x2 double]
Selection Function	Roulette
Cross Over Function	Heuristic
Number of variables	2
Scaling function	Rank
Domain of a	[-10;40]
Domain of b	[-10;40]

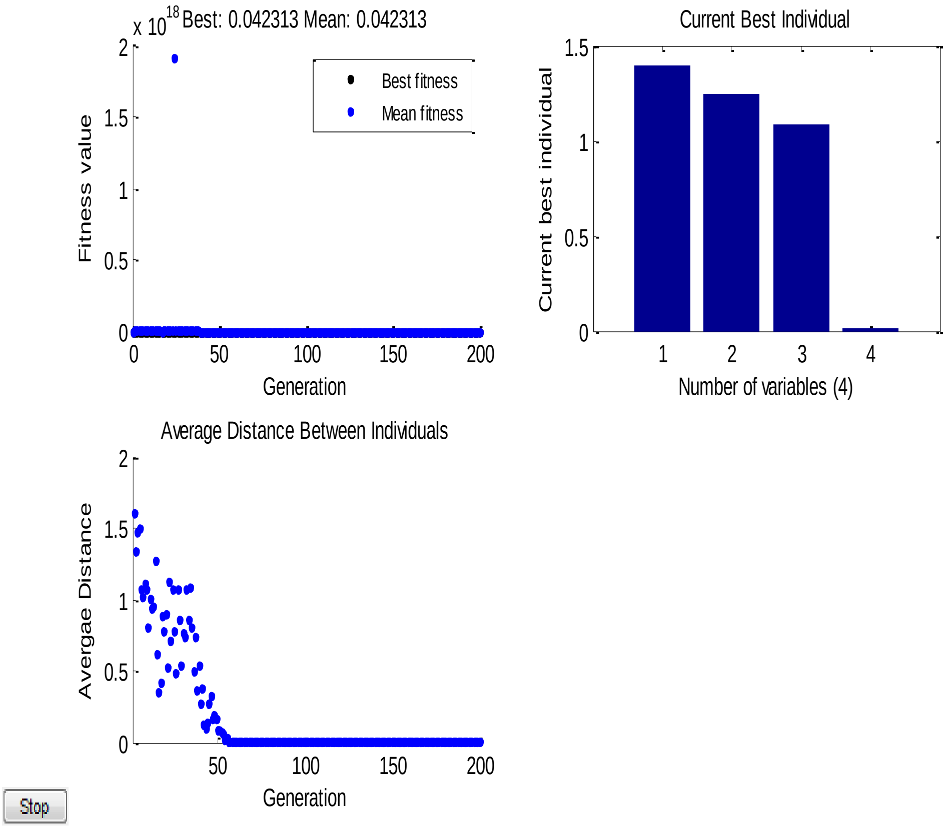

Fig 2: Best fitness value of $rmse$ and the values of project specific parameters a, b, c, and d of modified COCOMO Model 1

Table 2: Control parameters of GA to tune the parameters a, b, c and d of Modified COCOMO Model 1

Parameters	Value
Population Size	5000
Elite Count	4
Crossover Fraction	0.7000
Generations	200
Initial Population	[5000x4 double]
Selection Function	Roulette
Cross Over Function	Heuristic
Number of variables	4
Scaling function	Rank
Domain of a, b, c, and d	[-10;40]

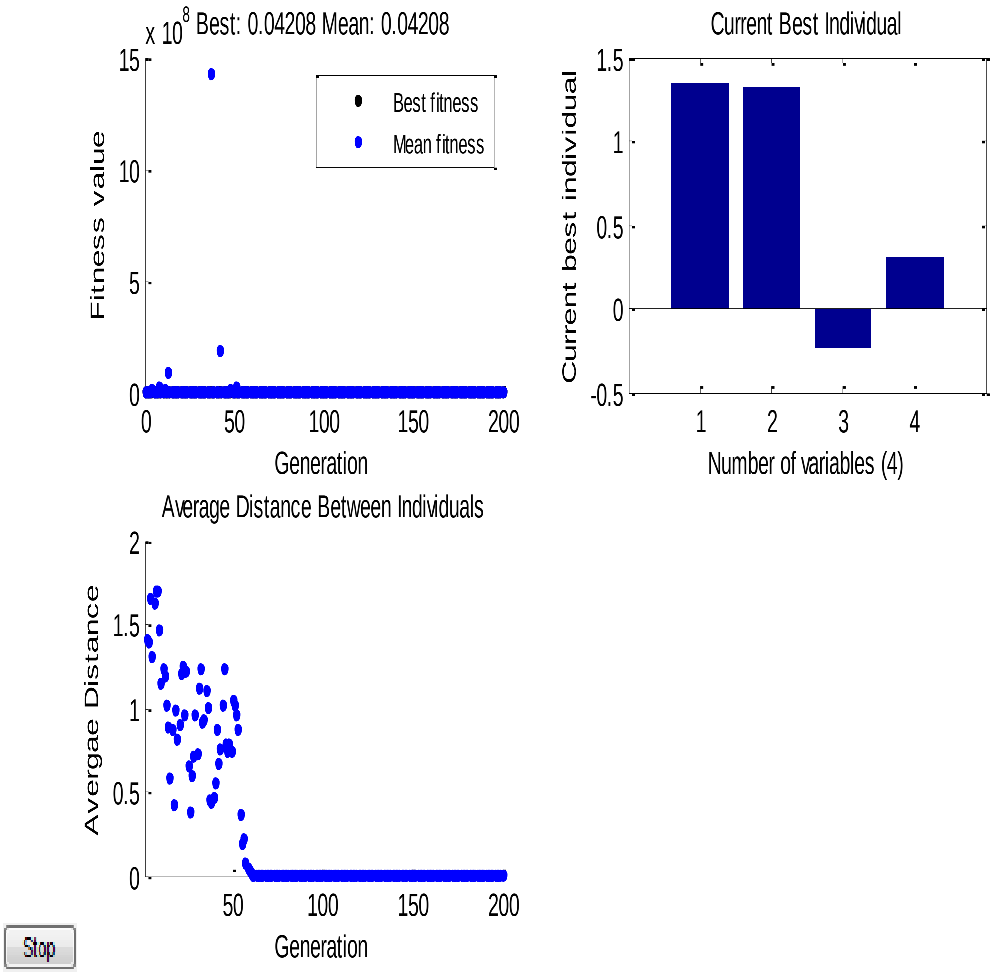

Fig 3: Best fitness value of $rmse$ and the values of project specific parameters a, b, c, and d of modified COCOMO Model 2

Table 3 Control parameters of GA to tune the parameters a, b, c and d of Modified COCOMO Model 2

Parameters	**Value**
Population Size	5000
Elite Count	4
Crossover Fraction	0.7000
Generations	200
Initial Population	[5000x4 double]
Selection Function	Roulette
Cross Over Function	Heuristic
Number of variables	4
Scaling function	Rank
Domain of a, b, c, and d	[-10;40]

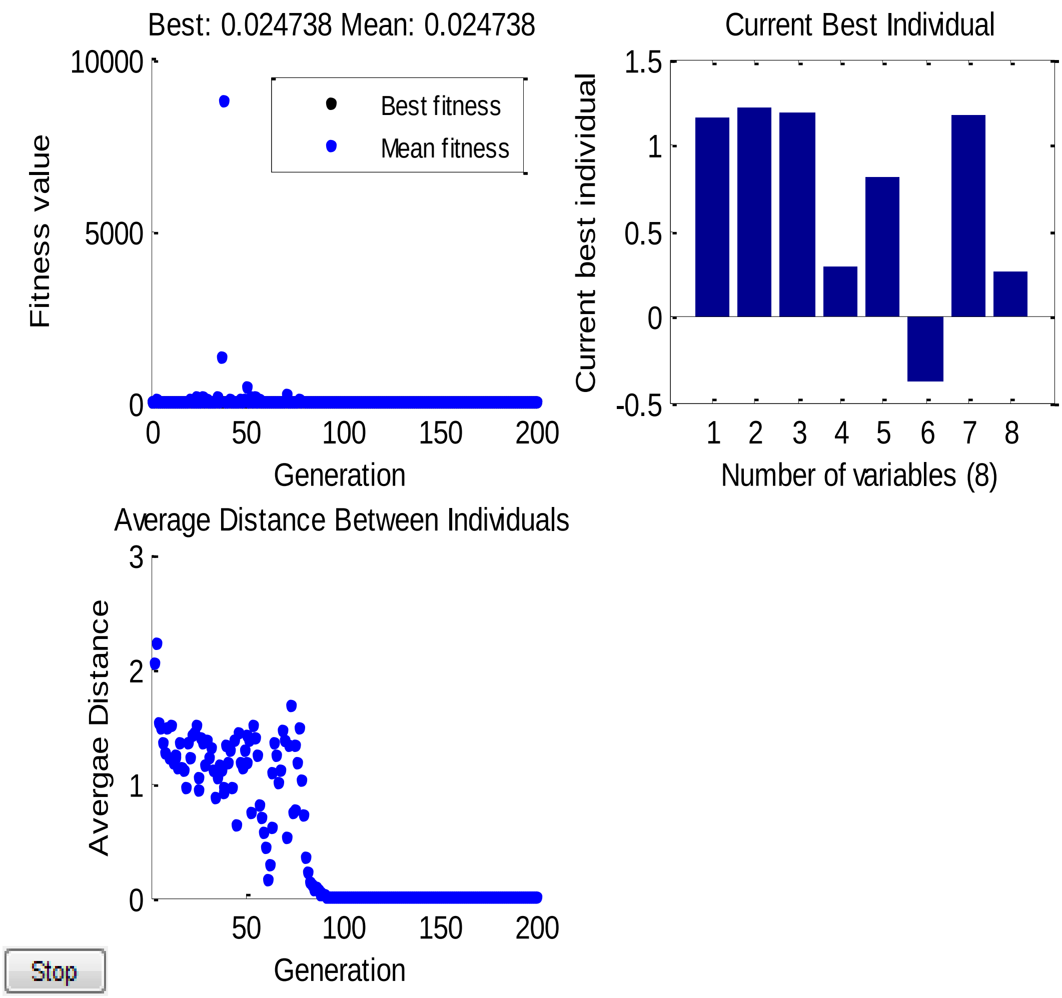

Fig 4: Best fitness value of $rmse$ and the values of project specific parameters a, b, c, d, e, f, g, and h of modified COCOMO Model 3

Table 4: Control parameters of GA to tune the parameters a, b, c, d, e, f, g, and h of Modified COCOMO Model 3

Parameters	**Value**
Population Size	5000
Elite Count	4
Crossover Fraction	0.7000
Generations	200
Initial Population	[5000x8 double]
Selection Function	Roulette
Cross Over Function	Heuristic
Number of variables	8
Scaling function	Rank
Domain of a, b, c, d, e, f, g, and h	[-10;40]

8. ACKNOWLEDGMENT

Our thanks to the expert Prof. K.K. Shukla IIT BHU who have provided guidelines in applying GA in this research paper.

9. REFERENCES

[1] Alaa F. Sheta 2006 Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects. Journal of Computer Science 2(2) Science Publication pp 118-123

[2] Alaa F. Sheata, Alaa Al-Afeef 2010. A GP Effort Estimation Model Utilizing Line of Code and Methodology for NASA Software Projects" 978-4244-8136-1 IEEE Transaction

[3] Alaa F. Sheta et al. 1996. Parameter estimation of nonlinear systems in noisy environment using genetic algorithms. Proc. IEEE Intl. Symp. Intelligent Control(ISIC'96), pp:360-366

[4] Anil Kumar, C.S.Yadav et al 2012. Parameter tuning of COCOMO Model for software effort estimation using PSO. ICIAICT ISBN 978-93-81583-34-0 pp 99-105

[5] K.K. Shukla 2000. Neuro-genetic prediction of software development effort. Information and Software Technology 42 701-713 ELSEVIER

[6] Boehm,B. 1981. Software Engineering Economics. Englewood Cliffs NJ Prentice Hall.

[7] Sehra S.K. et al 2011. Soft Computing Techniques for Software project Effort Estimation. International Journal of Advanced Computer and Mathematical Science, ISSN 2230-9624, vol-2 Issue 3, pp 160-167

[8] Sultaan Aljahdali, Alaa F. Sheta 2010. Software Effort Estimation by Tuning COCOMO Model parameters using differential evolutions.

[9] Holland, J.H. "Adaptation in Neural and Artificial Systems", University of Michigan Press, Ann Arbor, MI, 1975.

[10] Data Web Site: http://promise.site.uottawa.ca /SERepository/datasets/cocomo81.arff

[11] Chandra Shekhar Yadav et al. 2014. Reliability of Object oriented system using vague lambda-tau modeling. International journal of Fuzzy computation and modeling. InderScience Publishers [in Press].

[12] Idri et al. Investigating Soft Computing in Case-Based Reasoning for Software Cost Estimation. Engineering Intelligent Systems for Electrical Engineering and Communications, 2002 10(3): p. 147-157.

[13] Hakutta et al. A Software Size Estimation Model and Its Evaluation. Journal of Systems and Software, 1997. 37(3): p. 253-263.

[14] Shepper. et al. 1997. Estimating software project effort using analogies. IEEE Tran. Software Engg. 23; 736-743

[15] H. Garg et al. Behavior Analysis of Pulping Unit in a paper mill with Weibull Fuzzy Distribution Function using ABCBLT Technique. Int. J. of Appl. Math. And Mech. 8(4):86-96, 2012.

[16] A. Doostparast et al. A Fuzzy Approach to Sequential Failure Analysis Using Petri nets. 2010 21(2) pp. 53-60 IJIE

[17] Naveen Kumar et al. Reliability analysis of waste clean-up manipulator using genetic algorithms and fuzzy methodology. Computers & Operations Research 39(2012) 310-319.

[18] Somesh et al. Hybrid evolutionary techniques in feed forward neural network with distributed error for classification of handwritten Hindi 'SWARS' 2013 Connection Science vol. 25, No. 4, 197-215 Taylor & Francis.

[19] Baily, J.W. et al. A meta model for software development resource expenditure" Proc. Int. Conf. Software Engineering, pp. 107-115 1981.

[20] Kennedy, J. and Eberhart, R. Particle Swarm Optimization. Proceeding of the fourth IEEE international Conference on Neural Networks, Perth, Australia IEEE Service Center (1995)

[21] Kennedy, J. et al. SWARM Intelligence, Morgan Kaufmann, 2001.