International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 2, 2019 (Special Issue)

Parameter Tuning for Software Effort Estimation Using Particle Swarm Optimization Algorithm

Ashok Kumar
Dept. of CSE, Dr. APJ Abdul Kalam Technical University,
Lucknow, India
ash_chh@rediffmail.com

B.D.K Patro
Dept. of CSE, REC Kannauj,U.P
Kannuj, India
bdkpatro@rediffmail.com

Brajesh Kumar Singh
Deptt. of CSE, RBS Engineering Technical Campus,
Agra, India
brajesh167@gmail.com

***Abstract*— Estimation is a vital piece of software engineering projects. This gives the effect on key financial procedures, including planning and offer recommendations. And this also plays a vital role in deciding the execution boundaries of the project that depends on the ability to produce accurate effort estimates. The Constructive Cost Model (COCOMO) is one of the common models of Software Effort Estimation (SEE). In this model, the Kilo Line of Code (KLOC) is utilized to quantify the effort for improvement of the software projects. This paper features the interrelationship between various elements of software projects viz. project size and effort. Many Swarm knowledge systems have been connected on an expansive scale for effort estimation. This paper highlights a new effort estimation model for software projects using swarm intelligence method that is known as Particle Swarm Optimization (PSO). The effective parameters on effort estimation have studied using the PSO algorithm. The results of the proposed model give better estimation in comparison to the COCOMO model for effort.**

***Keywords*— SEE, SCE, PSO, COCOMO, KLOC**

I. INTRODUCTION

Key monetary procedures, including planning, offer recommendations, and choosing the execution limits of the project [1] are influenced by the capacity to create exact effort estimation which is an essential piece of software engineering projects. Effort estimation is a significant movement for planning, monitoring, and for delivering the product on time that too in budget. not only this, the feasibility of project in terms of cost as well as the ability to meet the customer's requirements is also considered in the process of estimation [2].It is probable that the forecast of effort to be consumed, in a software project is, the utmost preferred variable in the procedure of project management. In the early stages of software project and the planning of remaining activities are determine the value of this variable. The estimation of past projects is an important advance for taking care of the issues in light of the fact that the estimation movement is tormented with vulnerabilities and obstructions. Since the issue of exact effort estimation is as yet open, the project supervisor is stood up to at with indistinguishable messes from a couple of years prior [3]. The software industry's failure to provide accurate estimates of development cost, effort, and time to be consumed is well known [4]. Software development effort has been observed to be one of the most exceedingly terrible evaluated qualities in the course of the most recent couple of years. noteworthy over or thinks little of can be pricey for organization, and the intensity of a product organization vigorously relies upon the capacity of its project managers to precisely foresee ahead of time the effort required in building up the software frameworks [5].It is also found that efforts needs to be estimated reliably to facilitate complete the projects on time and within budget as less than one-quarter of the project is estimated accurately.

Boehm [14] in 1981 gave the Constructive Cost Model (COCOMO) that is extensively known as effort estimation model in which created lines of code (DLOC) are the essential component which influences the effort estimation. The DLOC includes all program instructions and formal statements [6, 7]. Here, the point is to give a premise to the software effort estimation through an efficient survey of past research papers [8]. Some other research contemplates have exhibited that the dimension of exactness in software effort estimates is emphatically impacted by the choice of the info estimations of the parameters of these techniques. The accuracy of software development effort [9] is improved by the combination of input features selection and by the parameters optimization of machine learning methods. The uses of search-based methods have been suggested recently to address the software development effort estimation problem [10, 11]. Such a problem can be formulated as an optimization problem by identifying the estimation model which that provides the best prediction [5]. The success of the project depends on accurate estimation and the incorrect estimations leads the project to be failed. Since the software development needs high costs and will not meet the software needs [12]. SEE is one of the processes which must be done before programming the software projects [13]. SEE also used to determine the scope of the project, estimate the amount of work required and the program is scheduled to run software projects. The different algorithm models are suggested for work estimation, scheduling and costs of the software projects. COCOMO I was not a suitable model for meeting the software needs. COCOMO II was presented in 2000 [15]. COCOMO II is used for getting better estimation of the time and costs activities of software projects. We organize this paper as follows: in Section 2, the previous works are presented; in Section 3, the Software Effort Estimation models is introduced; in Section 4, we have introduced the PSO; in Section 5, the proposed model is explained; in Section 6, the evaluation and the results of the proposed model is explained and finally in Section 7, conclusion and the future works is presented.

II. PREVIOUS WORKS

The software developers try to estimate the costs of the software projects in the primary stages of the software development. The software manager tries to make the software product efficient and in high quality. In software development process, the accurate estimation contributes in effort estimation and software development. Many researchers have done their work in the field of effort estimation. F.S. Gharehchopogh [16] used the Artificial Neural Networks (ANNs) for Software Cost Estimation (SCE). he compared 11 projects from 63 projects of NASA software project dataset in COCOMO model and showed that the error rate of the ANNs model is less with the error rate of the COCOMO model. In [17] utilized Genetic Algorithm (GA) for optimized value of the parameters of COCOMO model. according to the results of the experiments, it is possible to say that better effort estimation could be gained via GA. Researchers [18] have used Fuzzy Logic for SEE in many software projects. They have presented SCE as one of the difficulties and the critical exercises in software development. The suggested method of them shows that the use of FL is a model in software development. For the experiments results, 15 projects of the KEMERER projects set were used. According to the results, it is possible to conclude that the Mean Absolute Relative Error (MARE) is the evaluation factor. This factor is better in the proposed method than the algorithm methods. The cost function has many parameters in software projects. Some of the factors of software process which have direct size on cost estimation are Line of Code (LOC) and KLOC. The fuzzy logic-based experiments give the better result by using MARE percent. GA [19] is also used in SEE. The accuracy of the effort estimation also gives the validity of the software projects and the project manager would be able to manage them better. It has been seen that the Magnitude Relative Error (MRE) rate has decreased in comparison to COCOMO model by using GA. Multi objective particle swarm optimization algorithm have utilized [20] for optimization of COCOMO model parameters to minimize the MARE. according to the results of the experiments, the MARE value for small projects in COCOMO model is 16.1306 %, and is 9.0143% in suggested model, and is 18.1548% in large projects in COCOMO model and is 20.9717% in proposed model. The results of the experiments show that the suggested model is more efficient. Many Researchers [21] have used soft calculations techniques in SCE. They have utilized Fuzzy Logic, PSO algorithm and combination of both for cost estimation. They have used 30 projects of NASA software project dataset for the results of their experiments. According to the results of the paper, the proposed model could have estimated better in comparison to the various models and make Mean Magnitude of Relative Error (MMRE), 7.512%. In [22], it has evaluated and tested GA using SEE. In this research, the COCOMO model is less accurate in comparison to the artificial intelligence models in SEE. So, it is tried make the parameters in the proposed model more optimized and also make the effort estimation more accurate. In this reference the NASA software project dataset is used for the results of the experiments. According to the results, the suggested model could be better in estimation and reduce the MMRE to 0.2298% in comparison to various models.

III. THE SEE MODELS

The primary goal of effort estimation is the resource management to achieve a comprehensive view of costs of software products. It is clear that the software projects effort estimation is a basic and important part of the software engineering. Thus, the software engineering uses the effort estimation to give a suitable economic method for software projects to the project manager. SEE contains different models which presented the better result. One of the main factors of software projects management has many factors. The one of these factors is the accurate information about the time, effort and costs needed for the project execution. The COCOMO model is used for effort estimation of different software projects. The base COCOMO model is identified as equation (1) for SEE

$$E = a * (\text{Program Size})^b \quad (1)$$

Where, the parameters ‘a’ and ‘b’ are the inaccurate estimation of the complexity of the software and ‘Program Size’ is the number of the lines of the program in KLOC which affects the accuracy and the efficiency of the estimation [23]. Parameter ‘E’ shows the amount of effort based on units is Man-Months. This value is directly dependent on the size and complexity of the project. In COCOMO model parameters ‘a’ and ‘b’ depends on the size of the project. The different models of COCOMO for effort estimation using different values of ‘a’ and ‘b’ are showed in Table (1).

TABLE I. COCOMO BASIC MODELS FOR EFFORT ESTIMATION

Model Name	***a***	***b***	***Model Equation***
Organic	2.4	1.05	$E=2.4*(KLOC)^{1.05}$
Semidetached	3.0	1.12	$E=3.0*(KLOC)^{1.12}$
Embedded	3.6	1.20	$E=3.6*(KLOC)^{1.20}$

Basic model of COCOMO is a project estimation model. We identified the effort and software projects management using the models of Table (1). The main goal of COCOMO model is that all elements of the project get the same view of the goals, stages, organization and the technical and management procedures of the project and the effort of these elements are in direction of the software projects goals.

IV. PSO ALGORITHM

Particle Swarm Optimization algorithm is one of the swarm intelligence algorithms which act according to the population and random search. Particle Swarm Optimization algorithm was developed by Kennedy and Eberhart in 1995. This algorithm was being inspired by the birds' social behaviors which live in small and large groups [24]. Particle Swarm Optimization algorithm is a simulation of the social behavior of a group of the birds looking for food in an environment. Any of the birds are informed about the food place but they know in any stage that how far they are from food. So, the best procedure for finding food is following the nearest bird to the food.

In Particle Swarm Optimization algorithm, first the particles of the group are created in the problem space and the search for the optimized answer starts. In the total structure, the search of any particle follows the other particle which is the most optimized fitting function, and also it does not forget its own experiment and follows the estate of itself in which had the best fitting function. In any repetition of the algorithm, any particle changes the next position of itself according to the two values, the Pbest is the best position the particle ever had and the Gbest is the best position created for by all of the particles of the population. Gbest is the general knowledge of the population and when the particles change their positions according to Gbest, in fact they try to increase the knowledge level of themselves to the knowledge level of the population. From the conceptual view, the best particle, relates all particles of the group to each other. Identification of the next position for any particle takes place by the equations (2) and (3)

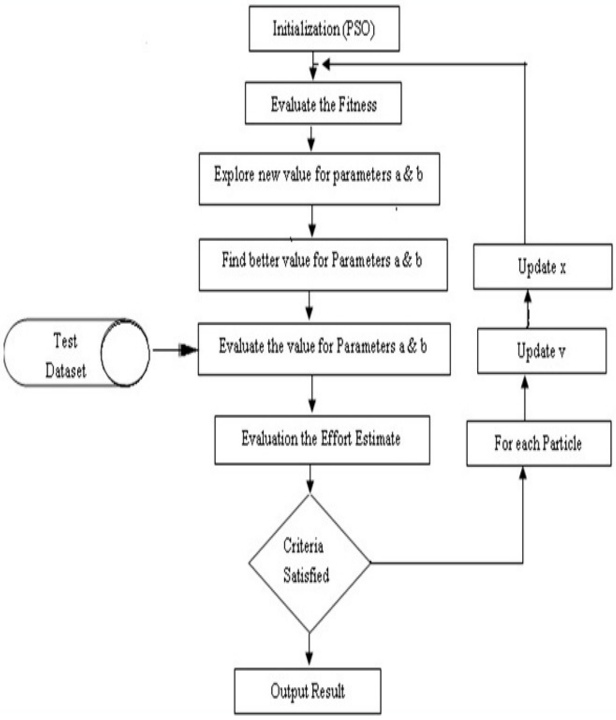
$$v_{i+1} = w.\ v_i + c_1.r_1.(P_{best\ i} - x_i) + c_2.r_2.(G_{best\ I} - x_i) \quad (2)$$

$$x_{i+1} = x_i + v_{i+1} \quad (3)$$

In equation (2), the symbols c_1 and c_2 are the two learning parameters. The symbols r_1 and r_2 are Function that is used to create the random numbers in [0, 1]. x_i parameter is the current position and the current velocity (v_i) is the velocity of the particles. The symbol w is used to control the effect of v_i on the next velocity and creates an equilibrium state between the ability of the algorithm in local and global search and in average we achieve the answer in the least time. So, for optimized operation of the algorithm in the search space, parameter w is defined in equation (4) [25, 26]:

$$\boldsymbol{w} = \boldsymbol{w_{max}} - \frac{((\boldsymbol{w_{max}} - \boldsymbol{w_{min}}) * \boldsymbol{i})}{\boldsymbol{i_{max}}} \quad (4)$$

In equation (4), i_{max} is the maximum number of the repetitions of the algorithm and the i parameter is the counter of the finding optimized answer. In equation (4), the w_{max} and w_{min} are the primary and the final values of inertia weight in algorithm execution, respectively. The value of inertia weight changes in linear manner from 0.9 to 0.4 in the execution time of the program. The large values of ω lead to the global search and the small values of it led to local search. To make equilibrium between local and global search, it is necessary to reduce the inertia weight in execution time of the algorithm. So, reducing the ω value, the search will take place locally around the optimized answer.



V. PROPOSED MODEL

The most important challenge we face in development of the large and complex software projects development is value accurate SEE. First the software projects were small and the costs of producing them included a small percent of the total costs and the error of the effort estimation did not affect the software execution considerably. But by the increase of the number, size and importance of the software projects and the costs of the software development, the software production is the most expensive element in software engineering and the increase of the costs has led the software teams to be defeated in production of the software projects.

The point to be considered in SEE is the method of selecting the suitable model among the estimation models in which the most accurate effort estimation takes place for the development of the software projects. Also, one of the important goals of studying SEE is studying the accessibility of the quality adjectives like reliability of the efficiency and success of the project. So, when we face the software projects the aspects like costs and utilization of the software are very important. The goal of estimation is to provide the utilization and the control factors of the project and contributes the project manager to define the problem making fields. In the proposed model it is tried to use the PSO algorithm and evaluate the COCOMO model parameters and find the more accurate value. Flowchart of the proposed model is presented in Figure (1).

Fig. 1 Flowchart of the Proposed Model

SEE in primary stages of the development of the software projects is an advantage for controlling the costs. In the proposed model, the PSO algorithm is used and the most optimized value for parameters 'a' and 'b' is found and replaced in COCOMO model. The Algorithm of proposed model is given below:

Step 1: Initialization PSO Parameters
Step 2: Evaluate the Fitness Function
Step 3: Loop
  Explore new values for Parameters a, b
  Find the better values for Parameters a, b
  Evaluate the values of Parameters a, b
  Replace the values a and b in COCOMO Basic Model
  Test Dataset
  Calculate the Effort Estimate
  Update velocity of Particle, V
  Update position of Particle, X
Step 4: Until (Stopping Criteria not meet)
Step 5: Return the best Effort

VI. EVALUATION AND RESULTS

SEE the base of which is effort and quality of the software projects, is one of the procedures of the success of the software projects and is very important many software projects. Different methods are presented for SEE any of which try to present the accurate estimation of the costs of the software production. In this paper the KEMERER [27] dataset is used for evaluation and results.

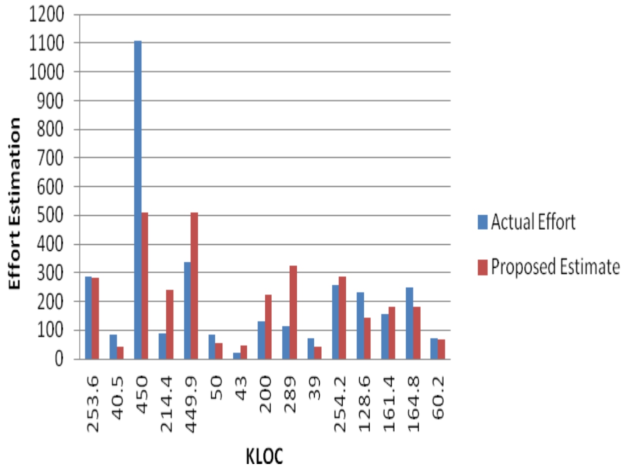
TABLE II. KEMERER DATASET [27]

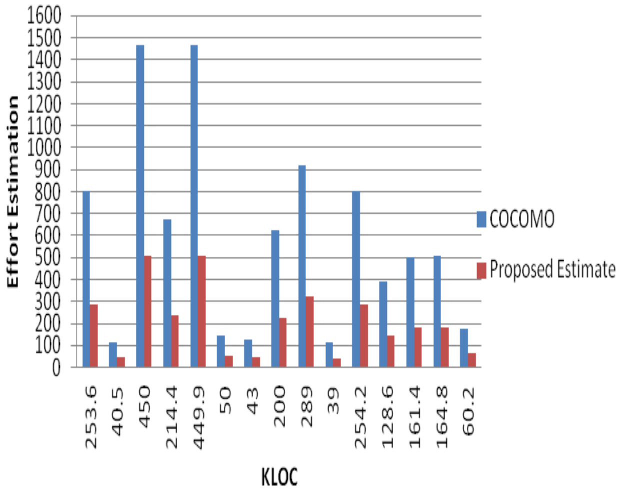
Project No.	***KLOC***	***Actual Effort***
1	253.6	287
2	40.5	82.5
3	450	1107.31
4	214.4	86.9
5	449.90	336.3
6	50	84
7	43	23.2
8	200	130.3
9	289	116
10	39	72
11	254.2	258.2
12	128.6	230
13	161.4	157
14	164.8	246.9
15	60.20	69.90

In Table (2) the actual effort estimation values are shown. Although many algorithms are presented for effort estimation, each of them achieves different values for software projects development. In PSO algorithm, there are some parameters affecting the operation of the algorithm. The proposed model population number the 1000 and the number of iterations of the algorithm have considered 100 and also parameters value are c_1, c_2 = 1.5 which theses contribute to particles learning in effort estimate more accurate.

TABLE III. COMPARISON VARIOUS MODELS OF EFFORT ESTIMATE TO THE PROPOSED MODEL

Project No.	***KLOC***	***Actual Effort***	***COCOMO Estimate***	***Proposed Estimate***
1	253.6	287.00	802.72	284.03
2	40.5	82.50	116.96	44.24
3	450	1107.31	1465.83	507.97
4	214.4	86.90	672.97	239.58
5	449.9	336.30	1465.49	507.85
6	50	84.00	145.92	54.77
7	43	23.20	124.55	47.00
8	200	130.30	625.59	223.27
9	289	116.00	920.77	324.26
10	39	72.00	112.41	42.58
11	254.2	258.70	804.72	284.71
12	128.6	230.70	393.47	142.70
13	161.4	157.00	499.47	179.65
14	164.8	246.90	510.52	183.49
15	60.20	69.90	177.33	66.11





In Table (3) estimation using the five models SEE is shown. Also, in Table (3) the COCOMO basic model is used for effort estimation. The results of the experiments of Table (3) show that there is no specific control on the effort estimation of the software projects and often the work done for development of a software project is estimated more or less than the real value according to the different projects. In contrast, the proposed model estimates the more actual value in comparison to the algorithmic models. In effort estimation models of the Table (3), the proposed model estimates the value very close to the actual model. From the effort point of view, the proposed model has led to the

Changes in the actual model and these changes play important role in costs of effort estimation of the software projects. So, the proposed model is very beneficiary for effort estimation and could estimate the effort better in comparison to the various models. To evaluate the actual estimation and proposed model of the effort estimation factor, the KLOC is used. In Figure (2), the comparison diagram of actual estimation and proposed model is showed.

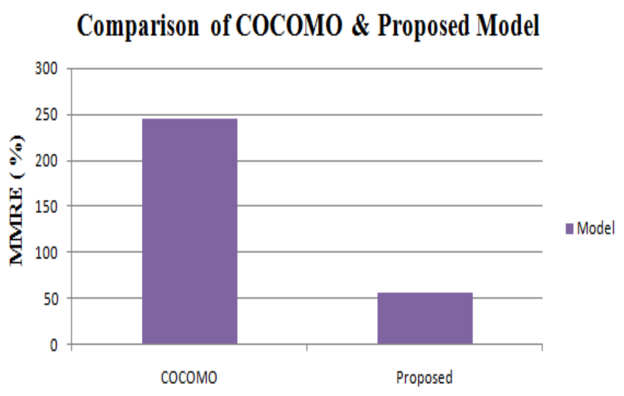


Fig. 2 Comparison Actual Effort with Proposed Estimate

In Figure (3) the comparison of effort estimation of proposed model and the COCOMO model is presented. As it could be seen, the proposed model gives more favored results.

Fig. 3 Comparison COCOMO with Proposed Estimate

For efficiency measuring of the proposed model with other models, MMRE factor is compared. MMRE is identified according to the equation (5) [28].

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} \frac{(|\text{Actual Effort}_i \ -\text{Estimated Effort}_i|)}{\text{Actual Effort}_i} \quad (5)$$

In Table (5), the comparison of the efficiency measuring of the proposed model to the various models is presented.

TABLE IV. KEMERER DATASET [27]

Model Name	*MMRE(%)*
COCOMO	245.39
Proposed	56.57

The results of the comparisons of the Table (4) show that the proposed model gives a better estimation in comparison to the various models and as it is clear the proposed model is more accurate than COCOMO model.

Fig. 4 Comparison MMRE of COCOMO with Proposed model

In Figure (4), MMRE the comparison flowchart of the various models is presented.

In Figure (4) it is clear that the efficiency of the different modes of effort estimation is different in results from MMRE point of view and the proposed model is MMRE less efficient in comparison to the various models which show that the proposed model is more accurate in estimation.

VII. CONCLUSION AND FUTURE WORKS

Software projects development is very complex in nature and many resources must be estimated before project execution procedure. Also, often the software companies try to increase the software projects quality and reduce the development costs of it. So, SEE must take place for development of the software projects for project manager to better support the process. Effort estimation of the software projects is a key parameter in project management. Achieving this goal needs accurate and reliable effort estimation and the estimation takes place according to the project conditions. In this paper, the effort estimation is done by PSO algorithm and as it is seen in the results of the comparisons the effort estimation of the proposed model is more accurate than the COCOMO model and the various. We hope in future, will design and implement an algorithm for SEE which will estimate other software factors as accurate as effort and will be effective in software projects management.

REFERENCES

[1] K. M. Furulund and K. Moløkken-Østvold, “Increasing software effort estimation accuracy-using experience data, estimation models and checklists,” Proceedings of the 7th International Conference on Quality Software (QSIC ’07), pp. 342–347, Portland, USA, October 2007.

[2] Q. Alam, P. Bhatia, and S. Sarwar, Systematic Review of Effort Estimation and Cost Estimation, Institute of Management Studies, Roorkee, India, 2012.

[3] J. J. Dolado, On the Problem of the Software Cost Function, Facultad de Informatica, Universidad del Pais Vasco-Euskal Herriko Unibertsitatea, Gipuzkoa, Spain, 2000.

[4] K. Molokken and M. Jorgensen, “A review of software surveys on software effort estimation,” Proceedings of the International Symposium on Empirical Software Engineering (ISESE ’03), pp.220–230, 2003.

[5] F. Ferrucci, C. Gravino, R. Oliveto, and F. Sarro, “Genetic programming for effort estimation: an analysis of the impact of different fitness functions,” Proceedings of the 2nd International Symposium on Search Based Software Engineering (SSBSE ’10), pp. 89–98, IEEE Computer Society, DMI, University of Salerno,Benevento, Italy, October 2010.

[6] A. F. Sheta, “Estimation of the COCOMO model parameters using genetic algorithms for NASA software projects,” Journal of Computer Science, vol. 2, no. 2, pp. 118–123, 2006.

[7] B. W. Boehm, Software Engineering Economics, Prentice Hall, IEEE, 1984.

[8] J. Magne and M. Shepperd, “A Systematic Review Of Software Development Cost Estimation Studies,” IEEE Transactions on Software Engineering, vol. 33, no. 1, pp. 33–53, 2007.

[9] P. L. Braga, A. L. I. Oliveira, and S. R. L. Meira, “A GAbased feature selection and parameters optimization for support vector regression applied to software effort estimation,” in Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC ’08), pp. 1788–1792, Cear´a, Brazil, March 2008.

[10] M. Harman and B. F. Jones, “Search-based software engineering,” Information and Software Technology, vol. 43, no. 14, pp. 833–839, 2001.

[11] J.Clarke, J. J.Dolado, M.Harman et al., “Reformulating software engineering as a search problem,” IEEE

Proceedings: Software, vol. 150, no. 3, pp. 161–175, 2003.

[12] A. Sharma, D.S. Kushwaha, "Estimation of Software Development Effort from Requirements Based Complexity", Procedia Technology, Vol. 4, pp. 716-722, 2012.

[13] M. Jørgensen, "Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation", Information and Software Technology, Vol. 53, Issue 12, pp. 1382-1390, Elsevier B.V, December 2011.

[14] B.W. Boehm, "Software Engineering Economics", Prentice-Hall, Englewood Cliffs, New Jersy, (1981).

[15] B.W. Boehm, "Software Cost Estimation with COCOMO II", Prentice Hall PTR, Englewood Cliffs, New Jersy, (2000).

[16] F.S. Gharehchopogh, "Neural Networks Application in Software Cost Estimation: A Case Study", International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011), pp. 69-73, IEEE, Istanbul, Turkey, June 2011.

[17] A.F. Sheta, "Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA

[23] T. Menzies, D. Port, Z. Chen, J. Hihn, S. Stukes, "Validation methods for calibrating software effort models," 27 th international conference on Software Engineering (ICSE'05), New York, NY, USA, pp. 587-595,2005.

[24] J. Kennedy, R. C. Eberhart, "Particle Swarm Optimization", IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.

[25] Y. Shi, R.C. Eberhart, "A modified particle swarm optimizer", IEEE International Conference on Evolutionary Computation, pp. 69-73, Anchorage, AK, 1998.

Software Projects", Journal of Computer Science, Vol. 2, No. 2, pp.118-123,2006.

[18] A. Mittal, K. Parkash, H. Mittal, "Software Cost Estimation Using Fuzzy Logic", ACM SIGSOFT Software Engineering, Vol. 35, No. 1, pp. 1-7, 2010.

[19] S. Joseph.K, T. Ravichandran, "Genetic approach for Software Effort Estimation: A Variant for COCOMO", International Journal of Computer Applications & Information Technology Vol. 1, Issue 3, pp. 58-64, 2012.

[20] P. Reddy P.V.G.D, Hari CH.V.M.K, S. Rao.T, "Multi Objective Particle Swarm Optimization for Software Cost Estimation", International Journal of Computer Applications, Vol. 32, No.3, pp. 13-17, 2011.

[21] Ziauddin, Sh. K. Tipu, K. Zaman, Sh. Zia, "Software Cost Estimation Using Soft Computing Techniques", Advances in Information Technology and Management (AITM),Vol. 2, No. 1, pp. 233- 238, 2012.

[22] B.K. Singh, A. K. Misra, "Software Effort Estimation by Genetic Algorithm Tuned Parameters of Modified Constructive Cost Model for NASA Software Projects", International Journal of Computer Applications, Vol. 59, No.9, pp. 22-26, December 2012.

[26] Y. Shi, R.C. Eberhart, "Parameter selection in particle swarm optimization", In Proceedings of Evolutionary Computation VII (EP'98), Springer-Verlag, pp. 591-600, San Diego, California, USA, 1998.

[27] C.F. Kemerer, "An Empirical Validation of Software Cost Estimation Models", Communication of the ACM, Vol. 30, No 5,pp:416-429,May 1987.

[28] S.G. Mac Donell, A.R. Gray, "A Comparison of Modeling Techniques for Software Development Effort Prediction", Proceedings of International Conference on Neural Information Processing and Intelligent Information Systems, pp. 869-872, 1997.