SEEPC: A Toolbox for Software Effort Estimation using Soft Computing Techniques

Hari .CH.V.M.K
Department of IT,
GITAM Institute of echnology,
GITAM University

Tegjyot Singh Sethi,
Kaushal .B.S.S
Department of CSE, GITAM
Institute of Technology,
GITAM University

Jagadeesh.M
Department of CSE, GITAM
Institute of Technology,
GITAM University

ABSTRACT
Software Effort estimation is the process of gauging the amount of effort required to complete the project. With the proliferation of software projects and the heterogeneity in there genre, there is a need for efficient software effort estimation techniques to enable the project managers to perform proper planning of the Software Life Cycle activates. In this article, a new hybrid toolbox based on soft computing techniques for effort estimation is introduced. Particle swarm optimization and cluster analysis has been implemented to perform efficient estimation of effort values with learning ability. The main aim of the toolbox is to provide an efficient, flexible and user friendly way of performing the effort estimation task, by catering to the needs of both the technical and the nontechnical users. The toolbox also implements the COCOMO model to enable a comparative analysis of the proposed model. It was observed that the model when provided with enough training data gave better results when compared with the standard COCOMO values

General Terms
Soft Computing, Software Cost Estimation, Swarm Intelligence.

Keywords
Constructive Cost Model (COCOMO), K-means algorithm, Particle Swarm Optimization (PSO), Software Effort Estimation, SEEPC: Software Effort Estimation–PSO–Clustering.

1. INTRODUCTION
One of the major tasks in the development of large scale software projects in the IT industry today is that of Planning. With an increase in the demand for large scale and complex software products, the need for efficient software cost estimation techniques is pivotal. The managers leading such projects benefit greatly by accurate estimates as it allows them to allocate resources judiciously and to draft out an optimal schedule [1]. This cost benefit analysis early on in the software development life cycle is useful in defining the profits obtained and in minimizing the risks associated with inadequate planning.

Software cost estimation is a probabilistic science and has an inherent uncertainty associated with it. The estimate is influenced by various factors known as cost drivers which lead to the uncertainty and nondeterministic nature of the process [2, 3]. These uncertainties could be accounted for by making use of soft computing techniques such as genetic algorithms [4], Fuzzy logic [5, 6], Particle Swarm Optimization (PSO), Neural Networks [11,15], etc. The stochastic nature of these techniques efficiently model's the uncertainties involved in the estimation process [7,14].

2. BACKGROUND
The Constructive Cost Model (COCOMO) developed by Boehm, is one of the famous models for estimating the software effort. In this model, the software effort expressed in person-months (pm) is given as a function of the project size in Kilo Delivered Lines Of Code (KLOC) and the effort adjustment factor (EAF). The obtained effort in person months can be converted to actual dollar cost for further processing.

In order to perform the optimization, several soft computing techniques have been suggested. Particle Swarm Optimization is one such technique. PSO is a robust stochastic optimization technique based on the movement of intelligent swarms [8, 9]. It relies on the concept of social interaction in order to reach an optimal solution to a problem.

The software projects encountered in practice usually vary in size and complexity .Hence, it is more effective to identify the common subcategories or partitions in the set of projects and work on them individually. This partitioning can be carried out by the classical centroid based technique: the K-means clustering algorithm [10]. By employing K-means clustering algorithm a set of data values can be partitioned into clusters having data values nearest to the associated centroid.

3. PROPOSED METHODOLOGY
The SEEPC model for software effort estimation is a hybrid model which employs soft computing techniques in order to perform the cost predictions. Here the intermediated COCOMO model is employed and the PSO is used to tune the parameters of the effort estimation equation. The data sets are clustered using k-means clustering algorithm.

In the COCOMO software effort model, the effort is expressed in terms of the size, given in Kilo lines of delivered Code (KDLOC), and the Effort adjustment factor (EAF). The mathematical formulation of the COCOMO model is given below:

$$Effort = a(size)^b * EAF + c \quad (1)$$

Here the terms a, b and c are statistical parameters measured by using regression analysis. The goal of the SEEPC hybrid model is to efficiently and accurately estimate the values of these parameters my making use of PSO and k-means algorithms. The techniques used in the proposed approach are described below

3.1 Particle Swarm Optimization

PSO is used in order to estimate the values of the parameters a, b and c in equation (1). In PSO, a group of particles move in an N dimensional search space in order to locate the optimal solution. Every particle in the swarm is guided by its own personal experience (Pbest) and also the experience of the swarm as a whole (Gbest**).**

For the purpose of tuning the parameters in equation (1), the particles of the swarm are assigned three velocity components V_a, Vb and V_c. The equation (2) ,(3) depicts the velocity and distance components for the parameter a. The parameters b and c are also modeled in a similar way. The i^{th} particle's position and the velocity during the k+1 iteration is mathematically formulated as shown below:

$$S_{ai}^{k+1} = wV_{ai}^{k} + c_1 rand_1 \left(Pbest_{ai} - S_{ai}^{k}\right) + c_2 rand_2 \left(Gbest_a - S_{ai}^{k}\right) \quad (2)$$

$$S_{ai}^{k+1} = S_{ai}^{k} + V_{ai}^{k+1} \quad (3)$$

Here , S_{ai}^{k} is current search point; S_{ai}^{k+1} is modified search point; V_{ai}^{k} is the current velocity;V_{ai}^{k+1} is the modified velocity; $Pbest_{ai}$ is the personal best value of the particle i; $Gbest_a$ is the global best location of the swarm; w is the weighting function; c_j are the weighting factors; $rand_j$ are uniformly distributed random numbers between 0 and 1.In the PSO technique the particle searches in the solution space within the range [-s,s] and tries to locate the optimal solution.

The above equation-2, 3 represents a standard PSO with Inertia weight. The particle's position in this case depends on the current velocity of the particle by a factor w called the inertia weight. Decreasing the inertia over time introduces a shift from the exploratory (global search) to the exploitative (local search) mode. The updating of weighting function and the weighting factors c1 and c2 is done with the following formulae:

$$W_{i+1} = \frac{[(T_{max} - T_i) * (W_{init} - W_{fin})]}{T_{max} + T_{min}} \quad (4)$$

$$C_1(t) = 2.5 - 2 * \left(\frac{t}{T_{max}}\right) \quad (5)$$

$$C_2(t) = 0.5 + 2 * \left(\frac{t}{T_{max}}\right) \quad (6)$$

Here, Wi_{+1} is new weight factor, T_{max} is the maximum number of iteration specified, T_i is the current iteration number, W_{init} =0.9 is the initial value of the weight, Wfin=0.4 is the final value of the weight [8, 9]. The objective function taken for the above implementation is the Mean Absolute Relative Error(MARE).It is given mathematically as:

$$MARE = mean\left\{\frac{Abs(MearuredEffort - EstimatedEffort)}{MeasuredEffort}\right\} * 100 \quad (7)$$

3.2 K- Means Clustering Algorithm

The data available for software cost estimation is inherently non linear and hence accurate estimation of results is difficult. In order to enable efficient tuning of parameters through PSO, the data is to be clustered so as to define some relationship between the values. The clustering of data sets is carried out by using the K-means clustering algorithm. In the K-means clustering algorithm the N observations given as a data set is clustered around K centroids, with each value in the data set belonging to the cluster to which it has the least mean distance. The data value is a <*size, EAF*> pair. The distance D, which is the Euclidian distance of coordinate points, and the Centroid values are given by:

$$D(size_i, EAF_i) = \sqrt{(size_i - size_c)^2 + (EAF_i - EAF_c)^2} \quad (8)$$

$$Centeroid(size_c, EAF_c) = \left(\sum_{i \epsilon cluster} \frac{size_i}{N}, \sum_{i \epsilon cluster} \frac{EAF_i}{N}\right) \quad (9)$$

3.3 Modes of Operation

The SEEPC toolbox functions in three modes: the training mode, the learning mode and the testing mode.

In the Training mode, the datasets are clustered using the k-means clustering algorithm. PSO is then employed on these clusters to tune the COCOMO parameters. The resulting parameters obtained for each cluster is stored for future computations.

In the learning phase, the input test data is identified as belonging to a particular cluster by using equation (8). It is then added to that cluster and the centroid values of that cluster is updated using equation (9). PSO is implemented on this cluster to obtain the new parameter values.

In the Testing Phase, the input data is classified into a particular cluster and the corresponding parameters of that cluster is applied to it to evaluate the corresponding effort..

4. SEEPC TOOLBOX DESIGN

The methodology proposed in the previous section has been implemented in the SEEPC toolbox to perform the software cost estimation. The toolbox provides a user friendly and pellucid way of performing the estimation task and analyzing the results and process involved. The toolbox has been developed in Visual Studio 2010(a product of Microsoft Corp.). A modular approach has been followed during the design and coding to make it more maintainable and extensible.

As a result of preliminary analysis of existing software cost estimation software, there was found to be a lack of models that use past experience in order to arrive at accurate results for a wide variety of data values[12]. The SEEPC toolbox was developed to cater to these needs. The vagaries in the data sets was tackled using the K-mean clustering algorithm, the PSO was used for the parameters tuning and the learning experience was obtained through the self K-means training

The toolbox has evolved under various modeling activities which can be summarized under the following three phases:

- **Functional model:** Describes about the functional behavior of the model through use case diagrams. (Fig. 1)
- **Object model:** In order to implement the functionalities, a number of class descriptions and interactions are described using the class diagrams.(Fig. 3)
- **Dynamic model:** The logic involved in each function defined in classes is described in the sequence diagram.(Fig.2)

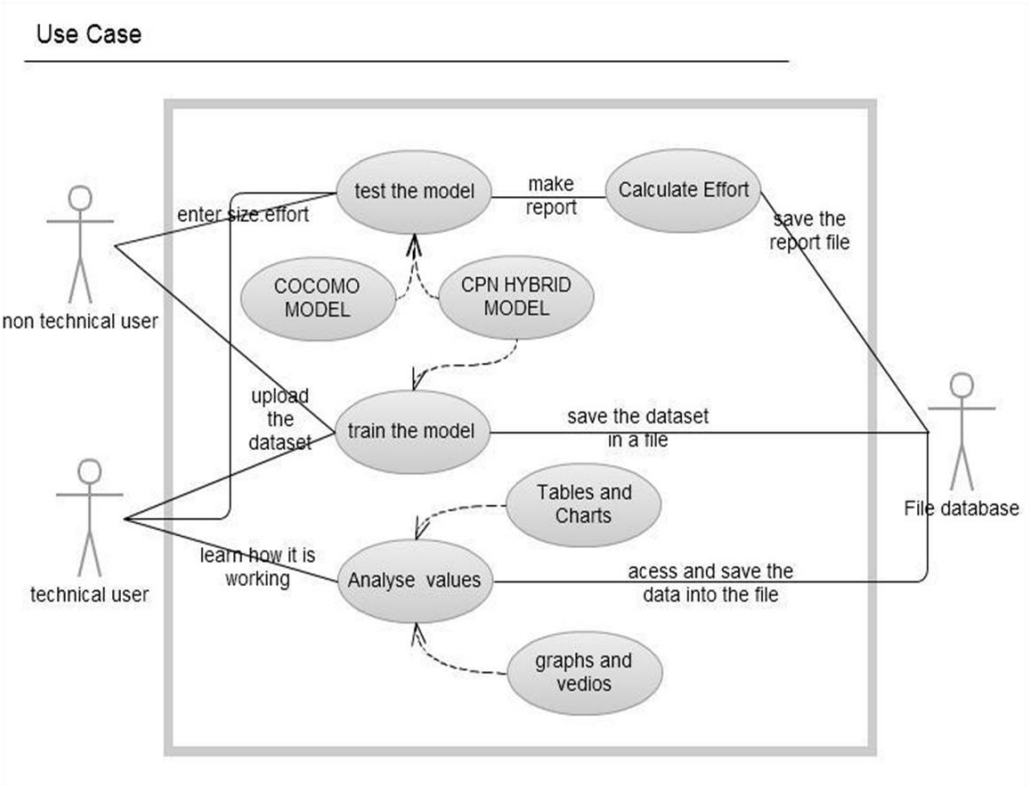

Fig 1. Initial use case diagram for SEEPC .

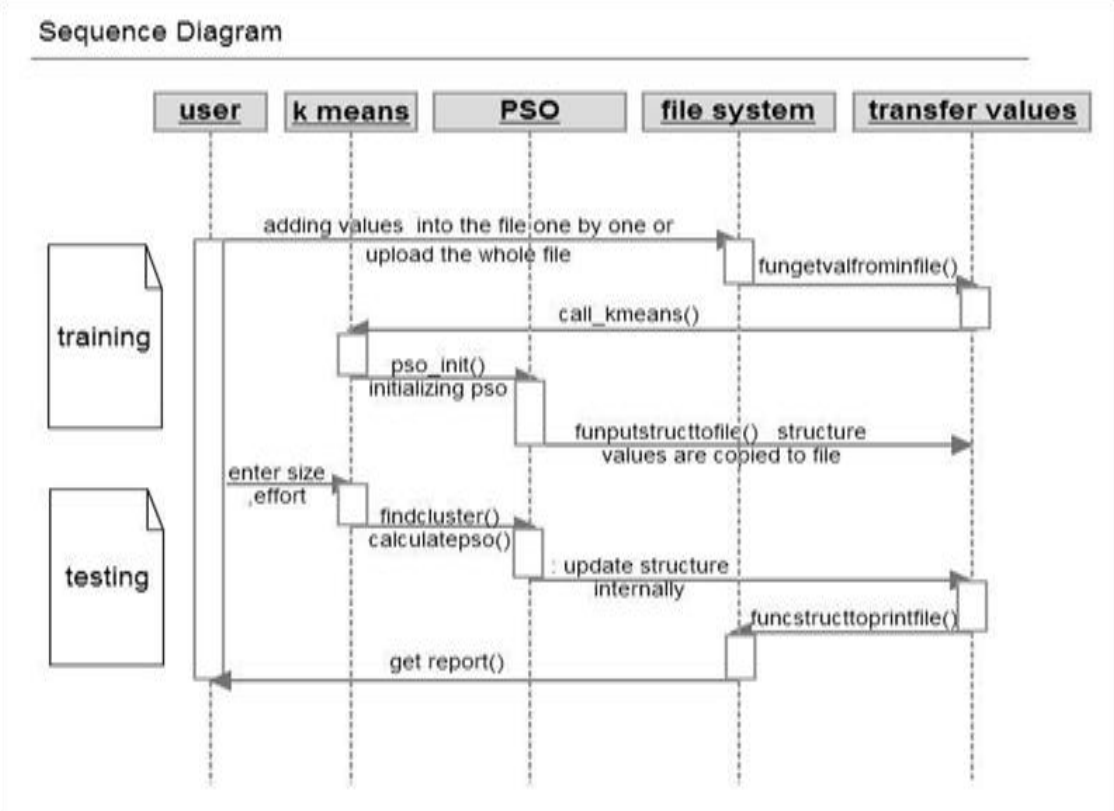

Fig 2. Sequence Diagram depicting flow of control.

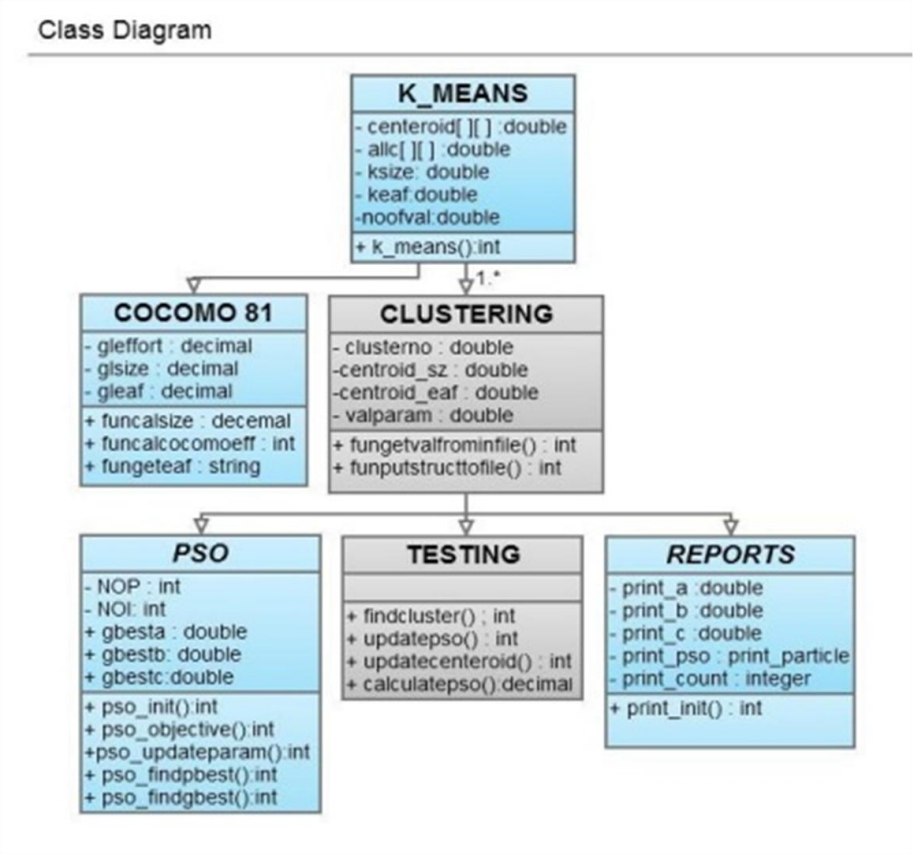

Fig 3. Class diagram depicting tool components.

4.1 Description of the Toolbox

The SEEPC toolbox encompasses two models: The COCOMO model and the SEEPC model. The user can choose any of the models to suit his needs. Upon the selection of the model the user must specify the size and the cost drivers for the project data he want to evaluate. User friendly cues and interactive suggestions tend to make this task less cumbersome. In the Hybrid model the user can upload or add the training data to train his model for future needs. A comprehensive training report is provided to enable the user to view the performance, working and the results of the model applied on the training dataset. In order to perform the estimation, the user is provided with files, reports, cluster information, PSO tuning simulation, MARE, cluster convergence and other features to suit his needs

Our toolbox is a desktop based application with an independent file system. There are three types of file formats used. The .SPC file is the application input file which the user can directly upload or can create using the GUI. The .SPCAPP file is the application file storing the trained data and the cluster information. The .SPCPRINT file is the printable report files. The above mentioned files are portable and can be used to transfer application data easily.

5. RESULTS AND DISCUSSIONS

The SEEPC toolbox provides the above mentioned gamut of feature in a user friendly and coherent manner. The toolbox is designed to cater to a dual audience: the project managers and the researchers. While keeping in view of the needs of the managers, the toolbox has been designed for simplicity and abstraction. The managers can enter the size and the EAF of the project through an interactive window which asks them about several qualitative questions about the project. These qualitative data is then converted to the quantitative data in the background

The managers can also view the reports in a pellucid way and analyze past data to make appropriate decisions.

To cater to the needs of users with research interest having an understanding of the basic concepts of soft computing and SEE, the toolbox provides a series of reports and visual aids to explain the internal working of the model and its efficiency. The user is provided with a training report as well as a testing report to analyze the components individually. In order to understand the K-means clustering process, the user is provided with scatter diagram depicting the clusters which were formed out of the initial sporadic data, as depicted in Fig. 4. The cluster wise PSO implementation is depicted through a simulation of the convergence process of the particles of the swarm as they move in the search space to locate the optimal solution; the same is shown in Fig 5. The user can also view the estimated effort, MARE, cluster density and the partitions in the datasets. The testing report is depicted in Fig. 7.In order to facilitate a comparative study; the SEEPC toolbox also provides an option to implement the standard COCOMO model onto the same data set (Fig. 6). This would be helpful for the users to analyze the efficacy of the two models on their pertinent datasets

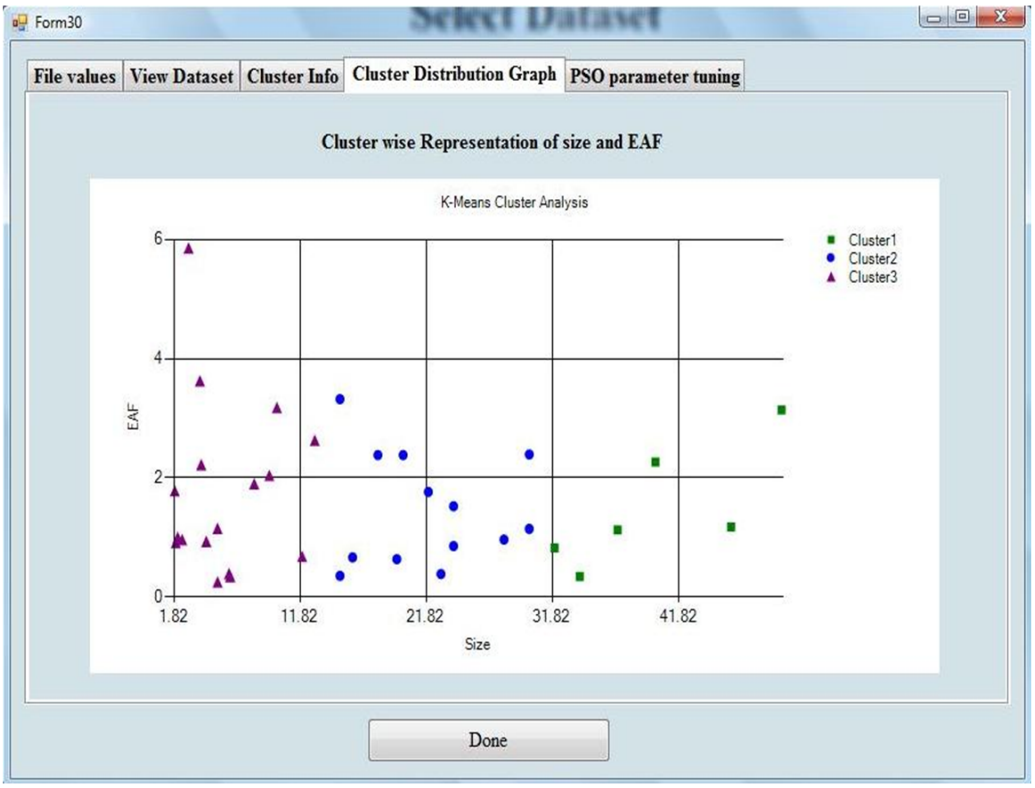

Fig 4. clusters obtained after application of K-means algorithm.

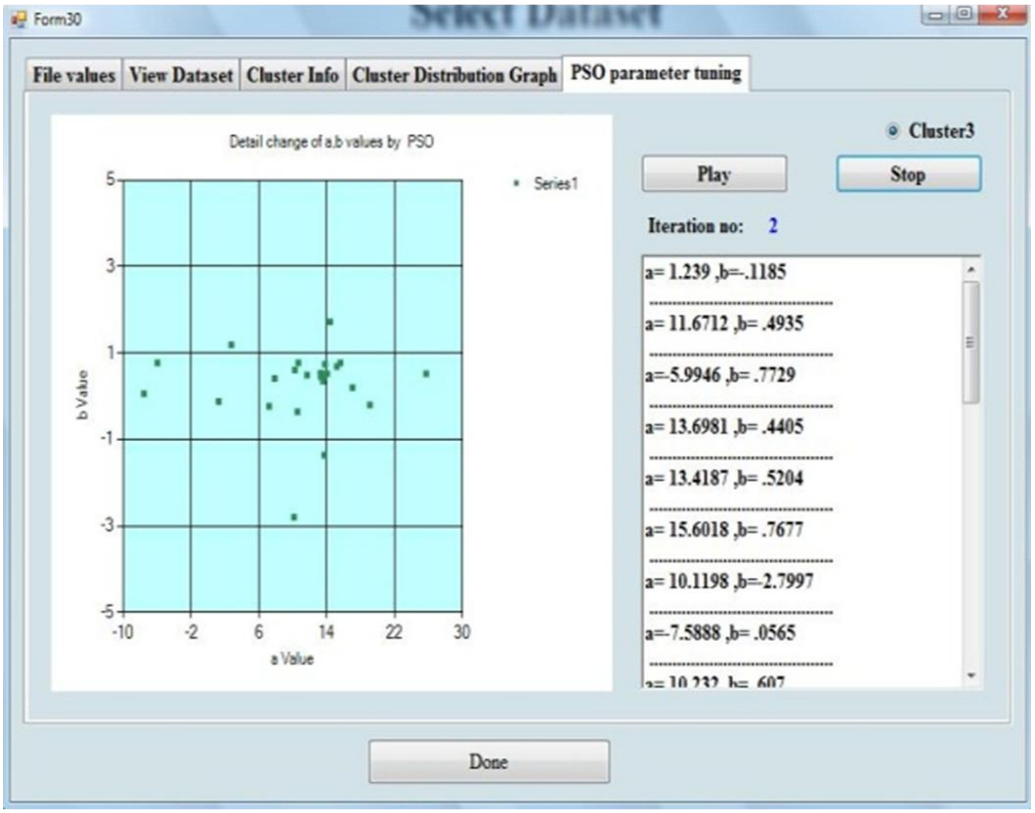

Fig 5. Simulation screen depicting the PSO convergence phenomenon.

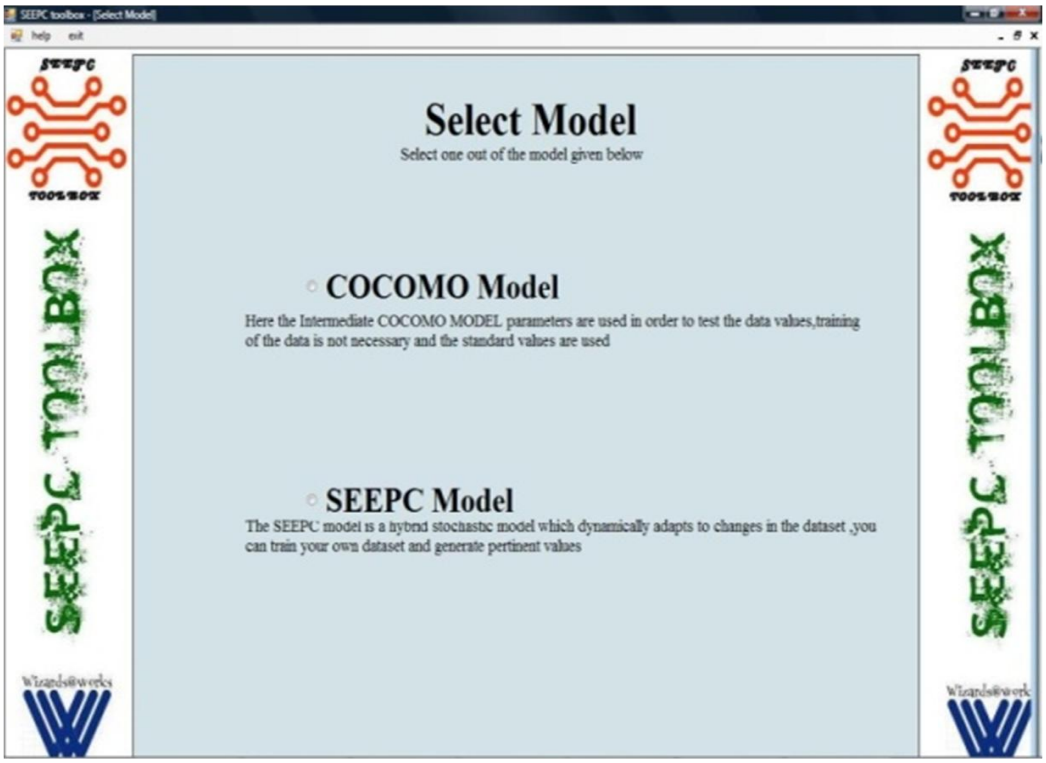

Fig 6. Screen depicting the two available models: COCOMO and SEEPC.

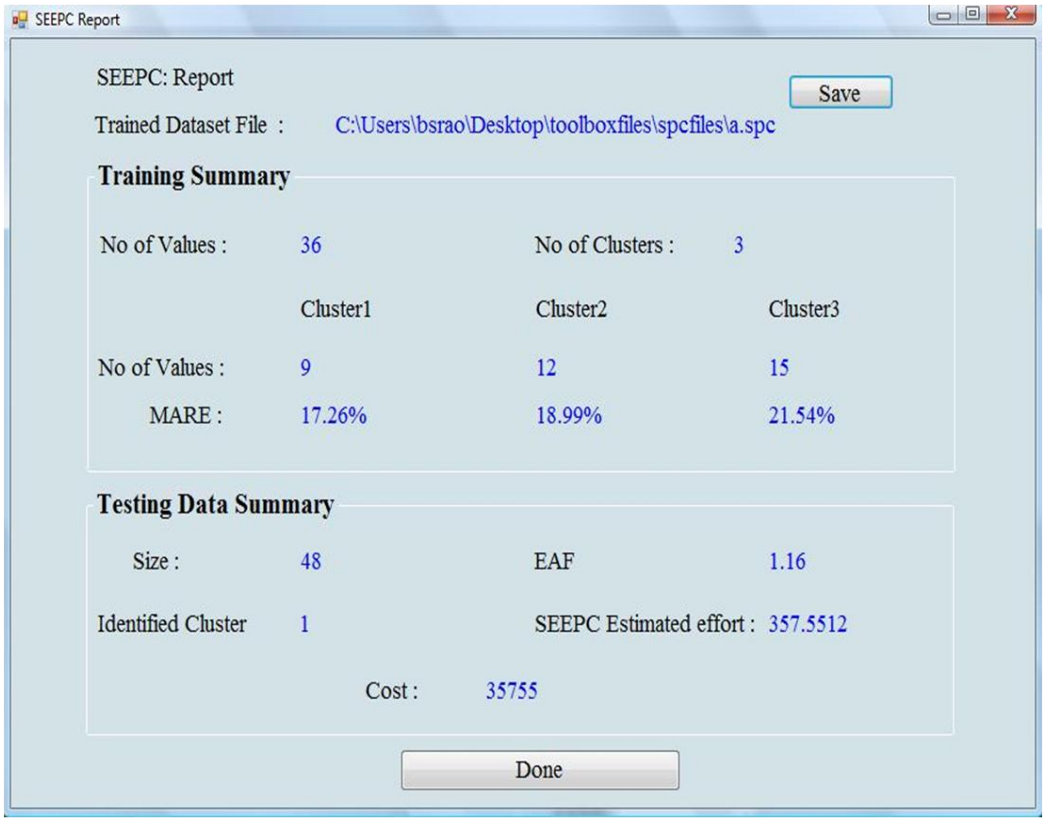

Fig 7. Final report for displaying the result.

One of the features that set the SEEPC toolbox apart from other software effort estimation toolbox is its ability to learn from historical data. The users can enter their own data sets and train the model to provide personalized results

5.1 Experimentation and Sample Results:

For the purpose of experimentation, the SEEPC toolbox was provided with a training set of 36 values [13] derived from the COCOMO 81 dataset. Upon training the model, 3 clusters were obtained. These clusters are illustrated in Fig 4. The parameter values obtained for each of the clusters is shown as follows:

Cluster 1: a=0.1449; b=1.9806; c=-6.2797.

Cluster 2: a=8.8274; b=0.7938; c=-14.5155.

Cluster 3: a=2.7291; b=0.9921; c=1.4942.

To illustrate testing and comparison, 9 values were taken and provided to the model. The Estimated Effort values (SEEPC), COCOMO Estimated Effort(C-EE), Measured Effort (ME) and the Cluster Numbers (C. No.) obtained are shown in Table 1.The Graph in Fig 8 shows the correspondence of the different effort values.

Table 1. Measured (ME) and Estimated Effort (SEEPC)

C.No	Size	EAF	ME	C-EE	SEEPC
1	28	0.45	50	47	40.28
1	48	1.16	387	239	357.4
2	25	1.09	130	145	132.42
2	21	0.87	70	68	77.37
2	23	0.38	36	33	34.66
2	13	2.81	98	133	123.54
3	9.1	1.15	38	42	39.83
3	6.9	0.4	8	9.8	10.3
3	3.7	2.81	40	38	35.68

It is seen that the effort values obtained are closer to the actual values in case of SEEPC model. The MARE values are shown in Table 2. The SEEPC model makes use of efficient clustering and PSO to learn and develop in its course of usage [13]. As such, it is more efficient for making long term accurate predictions than the traditional COCOMO model. It was found that the MARE obtained was lower when compared with the standard COCOMO model, indicating better performance. The model is also capable of dealing with large and complex data efficiently.

Table 2. Comparison of MARE (%) Values.

Model	Training	Testing
SEEPC Model	19.62	12.65
COCOMO Model	22.13	15.63

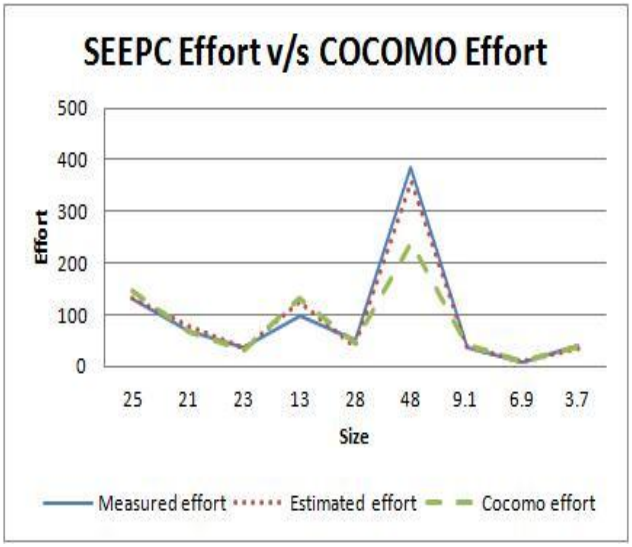

Fig 8. Comparison of SEEPC, ME and C-EE.

6. CONCLUSION

In this paper a toolbox for software effort estimation was introduced. The PSO and K-means self learning models were employed to account for the variability in the data and as a result make efficient predictions . The Toolbox implements the hybrid cost estimation model and is tested on a sample dataset. The results obtained proves to be more accurate than the standard COCOMO and also useful for practical purposes.

Software effort estimation can never be an exact science , however if enough historical data is provided , efficient predictions can be made. The SEEPC toolbox provides the feature to enable learning from past project data and hence enable domain specific projection of future resource requirements.

7. REFERENCES

[1] Bailey, J.W., Basili, R.: A Meta model for software development resource expenditures. In. Fifth International conference on software Engineering, CH-1627-9/81/0000/0107500.75@ 1981 IEEE, PP 107-129(1981).

[2] Briand, L.C., Emam , K.E., Bomarius, F.: COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment. International Software Engineering Research Network Technical Report ISERN-97-24, Revision 2, PP 1-24, (1997).

[3] Gruschke, T.: Empirical Studies of Software Cost Estimation: Training of Effort Estimation Uncertainty Assessment Skills.In: 11th IEEE International Software Metrics Symposium (METRICS 2005),doi: 1530-1435/05 © IEEE(2005).

[4] Sheta,A.F.: Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects. Journal of Computer Science 2 (2): PP: 118-123, (2006).

[5] Auer, M., Trendowicz , A., Graser, B., Haunschmid, E., Biffl, S.: Optimal Project Feature Weights in

Analogy-Based Cost Estimation: Improvement and Limitations. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 32, NO. 2, PP: 83-92 FEBRUARY (2006).

[6] Hari, CH.V.M.K. Prasad Reddy, P.V.G.D., Jagadeesh M.: Interval Type 2 Fuzzy Logic for Software Cost Estimation Using Takagi-Sugeno Fuzzy Controller. In. Proceedings of 2010 International Conference on Advances in Communication, Network, and Computing.DOI 10.1109/CNC.2010.14, 978-0-7695-4209-6/10 © IEEE (2010).

[7] Jørgensen, M., Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 33, NO. 1, PP: 33-53, JANUARY (2007).

[8] Poli, R., Kennedy, J., Blackwell, and T.: Particle swarm optimization an overview. Swarm Intell,PP: 33–57, Springer, DOI 10.1007/s11721-007-0002-0 (2007).

[9] Felix, Chan, T.S., Tiwari, M.K.: Swarm Intelligence: Focus on Ant and Particle Swarm Optimization. I-TECH Education and Publishing, ISBN 978-3-902613-09-7, PP: 1- 548, (2007).

[10] Bin, W., Yi, Z., Shaohui, L., and Zhonghi.S: CSIM: A Document Clustering Algorithm Based on Swarm Intelligence. 0-7803-7282-4/02@2002 IEEE PP: 477-482,(2002).

[11] Huang, X., Ho, D., Ren, J., Capretz, and L.F.: Improving the COCOMO model using a neuro-fuzzy approach. applied soft computing 7,pp-29-40,2007.

[12] Zamani, M., Netaji, H., et. al.: Toolbox for Interval Type-2 Fuzzy Logic Systems.In: Proceedings of the 11th joint Conference on Information Sciences, atlantis press(2008).

[13] Sethi, T.S., Hari, CH.V.M.K. Kaushal, B.S.S., Sharma,A.:Cluster Analysis & Pso for Software Cost Estimation.In: Das, V.V., Thomas, G., Gaol, F.L.(eds.) AIM 2011.CCIS,147, pp.281-286,Springer-Verlag Berlin, Heidelberg(2011).

[14] Sheta A., Rine D. and Ayesh A.: Development of Software Effort and Schedule Estimation Models Using Soft Computing Techniques, IEEE Congress on Evolutionary Computation (CEC 2008).978-1-4244-1823-7/(2008).

[15] Tadayon N. : "Neural Network Approach for Software Cost Estimation", Proceedings of the International Conference on Information Technology: Coding and Computing(ITCC'05),IEEE,2005.