Project Cost Estimation using hybrid technique Gray Wolf Algorithm and COCOMO Model

Shaymaa chyad*, Eklas Abd Algabar
Department of Software Engineering, Mosul University, College of Computer Science and Mathematics
Mosul, Nineveh, Iraq
Shaymaaahmad85@gmail.com

Abstract: Cost estimation of the Software project is one of the most challenging tasks in software engineering. In this paper, from software projects, a comparison between estimate and actual effort was done by applying the gray wolves algorithm to estimate the cost of the software projects for the NASA dataset. The intermediate COCOMO model was used with gray wolves algorithm by taking the KLOC of the project as input, in addition to 15 cost driver and giving effort as output. The suggested model of the cost estimation to helps the project manager by provide a quick and actually estimate the effort and time of software project, which in closer gives the cost for software projects.

Key-words: Gray Wolf Algorithm , COCOMO Model, Cost Estimation.

***Corresponding author**

E-Mail: Shaymaaahmad85@gmail.com (Shaymaa Chyad)

1. Introduction

Pressman defines the estimation as a developer attempt to determine the amount of money, effort, resources, and time needed to build a program or product system [1]. Estimating the cost, size, schedule and effort of software projects is a critical process in program management and planning. Project effort cannot be a static science, but a combination of historical data and complex techniques that can prove the accuracy of estimate [2]. Software Cost is consists of:

- Workforce means number of management and engineering staff allocated to the project.
- Interval i.e. the amount of time required to complete the project.
- Effort i.e. the personnel effort required to complete a project usually measured in person-months.

Cost estimation usually fails to accurately predict the actual expenses or the time needed to develop the project. Software cost estimation models have two problems. It becomes difficult to predict the costs and effort at the beginning of the project [2].

The data used for the COCOMO model were used to measure the difference between the estimator and the actual effort.

2. Literature Review

Rijwani and Jain (2016) proposed technique to use of ANN (artificial neural network) model technically-based technologically by using the multi-layered neural network forward that given with this practice training on back deployment. The COCOMO data set is used to test and train the network [3].

Kaushik, et al., (2013) investigate the use of Back-Propagation neural networks for software cost estimation. The model is designed in such a manner that accommodates the widely used COCOMO model and improves its performance [4].

Srinivasa, et al., (2011) suggested a new model structure based on Alaa F. Sheta by using Fuzzy logic for uncertainty of controlling prediction and tuned the parameters of the cost model by using swarm intelligence-Particle Swarm Optimization. The verification of proposed model results and comparison with the existing models was done with NASA software dataset [5].

Reddy, et al., (2010) proposed software effort estimation models based on artificial neural networks. The models are designed to improve the performance of the network that suits to the COCOMO Model [6].
Sehra, et al., (2017) use evolutionary computing techniques, Effort Adjusting factors (EAF) that including 15 cost drivers, which has six levels of rating: Very Low, Low, Nominal, High, Very High, Extra High. Bee colony optimization, Particle swarm optimization and Ant colony optimization to employ tune of the COCOMO Model parameters [7].

Kaushik, et al., (2012) the most widely used software cost estimation model, the Constructive Cost Model (COCOMO) is discussed. The model is implemented with the help of artificial neural networks and trained using the perceptron learning algorithm. The COCOMO dataset is used to train and to test the network [8].

Goyal and Padda, (2017) proposed model for tuning parameters of COCOMO model software cost estimation using Particle Swarm Optimization (PSO), and using clustering methods to divide the data items into number of clusters. Once the data has been divided, it will be easier to implement Particle swarm optimization on each cluster. PSO is used for tuning the parameters of each cluster [9].

There is a large number of swarm intelligence algorithms used to solve optimization problems. In this paper, the Gray Wolf Algorithm (GWA) was selected to apply it to the search data that obtained from real projects (60) NASA space.

3. COCOMO Model

The COCOMO model is the most extensively used algorithmic cost estimation technique due to its simplicity. This model provides us with the effort in person-month (PM), the development time of the project in months and the team size in persons. It makes use of mathematical equations to calculate these. The COCOMO model is the most widely used version. According to the researchers, the COCOMO model has greater estimation accuracy than the basic COCOMO model, and at the same time can be compared to the detailed COCOMO model parameters [1, 10].

The COCOMO Model is a divided into basic, intermediate and detailed model. The inputs of intermediate COCOMO model are: a, b, c, d (“Table 1”).

Table 1: Intermediate COCOMO Model.

Project	a	b	c	d
Organic	3.2	1.05	2.5	0.38
Semidetached	3	1.12	2.5	0.35
Embedded	2.8	1.2	2.5	0.32

Size of the project (KLOC).Very High, and Extra High, The development position of the project depends on one of three categories of program development situations: organic, semi-detached, and embedded. Each rating has a real number of scenes (Effort multiplier) (EM) [11, 12].

The effort estimation and development time can be calculated using the following:

$$E = a(KLOC)^b \times EAF$$

$$EAF = \prod_{i=1}^{15} EM_i$$

$$D = c(E)^d$$

Where:

E is the Effort estimation in person-months.

a, b, c, d Fixed values from the (Table 1).

EAF is Effort Adjustment Factor.

D is development time (duration) of the project.

The Effort Multipliers (EM) values for the EAF calculation are shown in the ("Table 2"):

Table 2: Value of Multipliers for Calculation of Effort.

NO.	Cost Drivers	Ratings					
		Very Low	Low	Nominal	High	Very High	Extra High
	Product Attributes						
1	Required S/w Reliability (RELY)	0.75	0.88	1	1.15	1.4	-
2	Size of Application Database (DATA)	-	0.94	1	1.08	1.16	-
3	Complexity of the Product (CPLX)	0.7	0.85	1	1.15	1.3	1.65
	Computer Attributes						
4	Run Time Performance Constraints	-	-	1	1.11	1.3	
5	Memory Constraints (STOR)	-	-	1	1.06	1.21	
6	Virtual Machine Volatility (VIRT)	-	0.87	1	1.15	1.3	-
7	Turnaround Time (TURN)	-	0.87	1	1.07	1.15	-
	Personal Attributes						
8	Analyst Capability (ACAP)	1.46	1.19	1	0.86	0.71	-
9	Application Experience (AEXP)	1.29	1.13	1	0.91	0.82	-
10	Programmer Capability (PCAP)	1.42	1.17	1	0.86	0.7	-
11	Virtual M/c Experience (VEXP)	1.21	1.1	1	0.9	-	-
12	Programming Language Experience	1.14	1.07	1	0.95	-	-
	Project Attributes						
13	Modern Programming Practices	1.24	1.1	1	0.91	0.82	-
14	Use of Software Tools (TOOL)	1.24	1.1	1	0.91	0.83	-
15	Required Development Schedule	1.23	1.08	1	1.04	1.1	-

4. Gray Wolf Algorithm (GWA)

4.1 Overview

The Gray Wolf Algorithm provided by Mirjalili in 2014 [13], this algorithm imitates the social leadership and hunting behavior of gray wolves in nature and has a strict system gray wolves usually live in groups, where the society is divided into four sections, according to the hierarchical structure "Fig. 1" [14] :

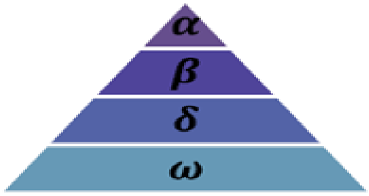

Fig. 1 Hierarchical Structure of Gray Wolves.

- The leader wolf in the group is called alpha and symbolizes it (α), which is located at the top of the pyramid. Alpha may not be the strongest wolf in the group, but it should be the best in leading a herd, be it male or female, responsible for group decisions and hunting decisions, such as predatory behavior, food distribution, time and place of sleep.
- The second level of the pyramid is located in the beta wolf and symbolizes it (β). Plays an important role in helping Alpha manage the group and make decisions. He only needs to respect alpha and can lead others.
- The third level is located in the delta wolf pyramid and symbolizes it (δ), which must follow the alpha and beta instructions. When alpha and beta become obsolete it can be reduced to delta. Scouts, guards, elders, fishermen, and caregivers belong to this category.
- The lower part of the pyramid is called Omega and symbolizes it (ω), plays the role of scapegoat, Omega must obey the orders of the previous three levels.

4.2 Mathematical Formula of Algorithm

The gray level is determined by the fitness function. According to fitness value, the best solution that takes the best fitness function value is like alpha wolf beta wolf and delta wolf. In this paper, these three solutions are set as a master group [15].

The rest of the Wolves are omega wolf. The following figure illustrates the hunting process as shown in "Fig. 2", the predation process is divided into three processes.

Fig. 2 Hierarchical Structure of Gray Wolves.

- Encircling the prey:

$$D = \left| C \times X_p(t) - X(t) \right|$$

$$X_p(t+1) = X_p(t) - A \cdot D$$

In these equations, $t + 1$ represent the next repetition step, X represents the location of one wolf, and D represents the position of the prey. A and C are coefficients calculated according to the following equation:

$$A = 2a \cdot r_1 - a$$

$$C = 2 \cdot r_2$$

Where r_1 and r_2 are a random number whose values are between [0, 1], a is decreasing with the number of iterations from 2 to 0 linear decrease during the repeat process.

- hunting prey

When the group of wolves determines the location of the prey, then beta, alpha and delta lead the wolf group to surround the prey. They assumed they knew the location of the prey. Thus, store the best three solutions and update the location of each wolf in the group according to the main group. These equations appear to update the site as follows.

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3}$$

$$X_1 = |X_\alpha - A_1 \cdot D_\alpha|$$

$$X_2 = |X_\beta - A_2 \cdot D_\beta|$$

$$X_3 = |X_\delta - A_3 \cdot D_\delta|$$

Where X_a, X_β, X_δ represent the top three solutions so far during the repetition process. Other parameters are given from the following equations.

$$D_\alpha = |C_1 \cdot X_\alpha - X|$$
$$D_\beta = |C_2 \cdot X_\beta - X|$$
$$D_\delta = |C_3 \cdot X_\delta - X|$$

- Attack the prey

In nature, gray wolves attack prey when they attack. Therefore, the prey is designed by the following equation.

$$\mathrm{a} = 2 - 2\left(\frac{t}{max}\right)$$

Where: *t* means the current number of times, a numerical value between 0 and *max* (maximum number of repetitions). “Fig. 3” Shows the Gray Wolves Algorithm (GWA)[16].

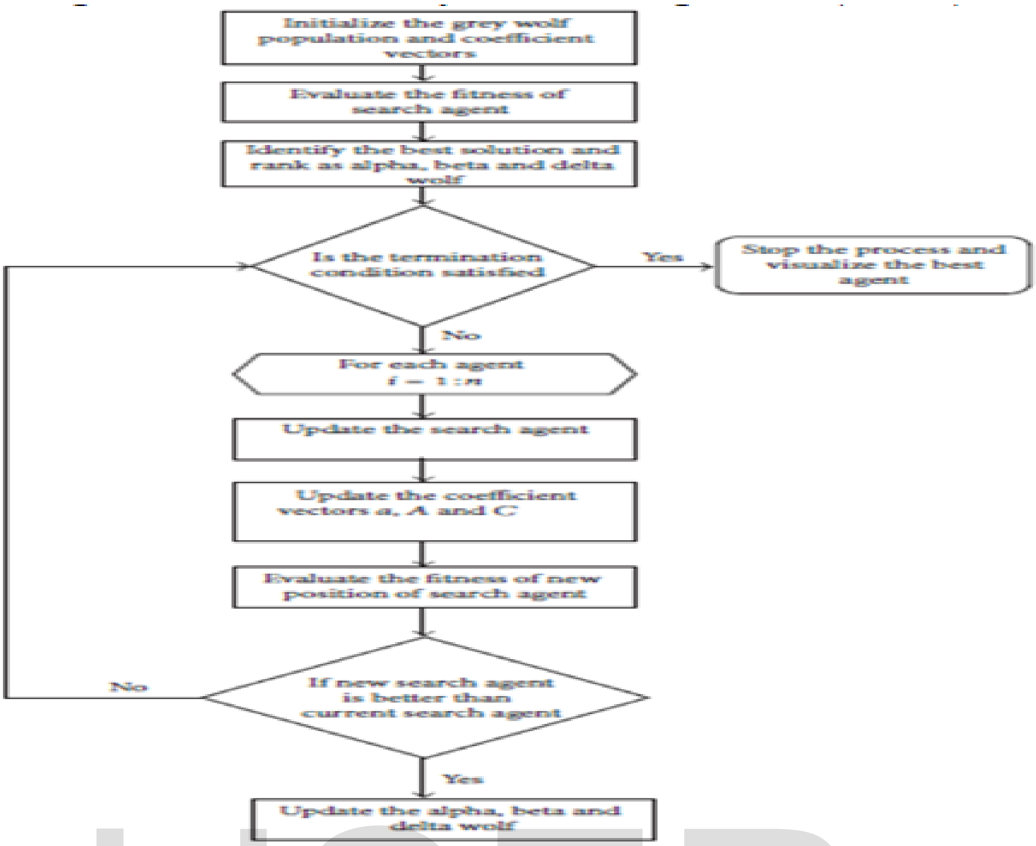

Fig. 3 Gray Wolf Algorithm.

5. Measuring Accuracy and Analysis of Results

Through a detailed review, the criteria affecting the cost of the project were discussed, methods that were developed earlier to effort, good practices and bad practices in detail. COCOMO has discussed which is the most popular tool to effort the cost of the program and uses code lines to evaluate the size of the program. A typical COCOMO form is implemented with multiple projects for the workaround listed. This model was corrected in the Gray Wolf Algorithm model and a detailed comparison was made between the estimated effort and the actual effort that was performed, it was found that this algorithm matches the effort and cost estimation data and gives better real results [17].

For the purpose of comparing the methods used to the effort in this search, the following metrics will be used:

- The Root Mean Square Error (RMSE) This measure calculates the square root of the rate of the calculated error between actual effort (E) and estimation effort ($\dot{E}$) from i=1 to (N); N number of projects, and divided by N, its equation is [19]:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(E_i - \dot{E}_i)^2}$$

- The Mean Magnitude Relative Error (MMRE %) is the percentage of the sum of the absolute values for the rate of Mean Relative Error (MRE) divided by N and its equation is [6]:

$$MMRE = \frac{1}{N}\sum_{I=1}^{N} MRE_i \times 100$$

$$MRE = \left|\frac{E - \dot{E}}{E}\right|$$

- Balanced Relative Error (BRE) is the absolute value of the error divided by the smallest value between the estimation effort $\dot{E}$ and the actual effort E [6].

$$BRE = \frac{|E - \dot{E}|}{\min(E - \dot{E})}$$

- Variation ratio for Variance-Accounted-For (VAF) it is used to measure the extent of the real closeness of the effort and its equation is [18]:

$$VAF = \left[1 - \frac{var(E - \dot{E})}{var(E)}\right] \times 100\%$$

Input data is taken from (60) NASA dataset projects that include COCOMO attribute values with KLOC and Actual effort. Estimated effort, development time, the proposed model provides Mean Relative Error and the balance relative error, as shown in the following ("Table 3"):

Table 3: Estimated Effort, Development time and metrics.

P. No.	KLO	Actual	Estimated	Dev.	MRE %	BRE
1	2.2	8.4	8.4	5.6	0.0025	0.0025
2	3.5	10.8	10.7	5.3	0.0017	0.0017
3	5.5	18	17.9	6.2	0.0050	0.0051
4	6	24	24.1	8.3	0.0062	0.0062
5	9.7	25.2	26.3	7.1	0.044	0.0442
6	7.7	31.2	31.1	9.2	6.6e-05	6.6e-05
7	11.3	36	36.1	7.8	0.0039	0.0039
8	8.2	36	35.9	9.7	0.0001	0.0001
9	6.5	42	30.2	9.1	0.2807	0.3904
10	8	42	32.6	9.4	0.2222	0.2857
11	20	48	48.0	8.6	0.0007	0.0007
12	10	48	40.3	8.1	0.1586	0.1886
13	15	48	47.8	8.6	0.0025	0.0025
14	10.4	50	50.2	11.0	0.0046	0.0046
15	13	60	54.3	11.4	0.0935	0.1032
16	14	60	59.7	11.8	0.0048	0.0048
17	19.7	60	61.5	9.3	0.0263	0.0263
18	32.5	60	96.5	10.7	0.6095	0.6095
19	31.5	60	79.9	10.1	0.3332	0.3332
20	12.8	62	62.0	11.9	4.3e-06	4.3e-06
21	15.4	70	69.9	9.7	3.6e-05	1.3e-05
22	20	72	71.9	9.8	0.0009	0.0009
23	7.5	72	72.3	12.7	0.0048	0.0102
24	16.3	82	82.0	13.3	4.3e-05	1.5e-05
25	15	90	72.1	9.8	0.1987	0.2479
26	11.4	98.8	89.5	13.7	0.0934	0.1030
27	21	107	106.9	14.7	0.0007	0.0007
28	16	114	75.0	12.9	0.3414	0.518
29	25.9	117.6	117.3	15.2	0.0019	0.0019
30	24.6	117.6	117.8	15.3	0.0020	0.0020
31	29.5	120	120.0	15.4	0.0003	0.0003
32	19.3	155	155.6	17.0	0.0044	0.0044
33	32.6	170	169.9	12.9	5.7e-05	5.7e-05
34	35.5	192	192.0	13.4	0.0001	9.6e-05
35	38	210	217.8	14.0	0.0372	0.0372
36	100	215	387.1	16.8	0.8005	0.8005

37	48.5	239	239.0	20.0	1.3e-05	1.3e-05
38	47.5	252	252.6	14.6	0.0027	0.0027
39	70	278	277.8	21.2	0.0004	0.0004
40	66.6	300	300.5	21.8	0.0019	0.0019
41	150	324	381.7	16.7	0.1782	0.1782
42	66.6	352.8	352.9	23.2	0.0004	0.0004
43	100	360	360.5	16.4	0.0014	0.0014
44	100	360	403.0	24.4	0.1196	0.1196
45	50	370	356.3	23.3	0.0369	0.0383
46	79	400	400.3	20.3	0.0009	0.0009
47	190	420	564.1	18.9	0.3430	0.34309
48	90	450	441.3	17.5	0.0191	0.01948
49	115.8	480	482.1	26.1	0.0045	0.0045
50	78	571.4	570.6	27.8	0.0012	0.0012
51	101	750	753.8	30.9	0.0050	0.0002
52	161.1	815	814.9	31.9	9.0e-05	9.0e-05
53	284.7	973	1231.4	37.3	0.265	0.265
54	227	567.5	603.0	19.3	0.0625	0.0625
55	177.9	1248	1247.9	24.4	2.5e-05	2.5e-05
56	282.1	1368	1376.4	38.9	0.0062	0.0062
57	219	2120	1799.7	27.5	0.1510	0.1779
58	423	2300	2299.0	29.7	0.0004	0.0004
59	302	2400	2411.0	30.2	0.0046	0.0046
60	370	3240	3235.2	33.1	0.0014	0.0014

Values of metrics applied on the above projects:
RMSE= 0.062,
MMRE= 0.075,
MMRE%= 7.5.
Comparison between Actual Effort and Estimated Effort in "Fig. 4" where X axis characterize KLOC and Y axis characterize Effort.

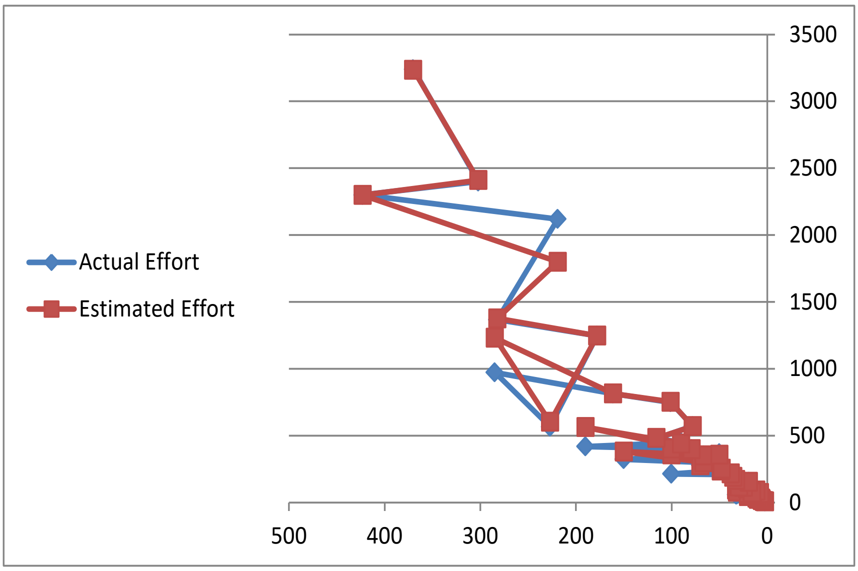

Fig. 4 Graph shows Actual Effort VS. Estimated Effort.

6. Conclusion

In this paper, we analyzed and studied the software cost estimation model using the COCOMO model based on the calculation of time and effort development. Output values are calculated by varying values ranging from very low to very high. Here, in the research work we have reviewed the detailed cost estimation models that were developed earlier. A detailed comparison between the estimated effort and the actual effort was provided through the "Fig. 4" by applying input values obtained from NASA projects. The work performed shows the estimated effort amount different from the actual effort based on the actual values obtained. We have proposed a model using a Gray Wolf Algorithm that takes the input values obtained through the COCOMO model and gives effort and time Development. The proposed model may help the project manager provide a more realistic estimate of the project effort and development time that includes the cost of the programs.

7. References

[1]. Pressman R. S. Software Engineering: a Practitioner's Approach. 7th Edition, McGraw-Hill Company, 2010: 889.
[2]. Nagar C., Dixit A. Software Efforts and Cost Estimation with a Systematic Approach. Journal of Emerging Trends in Computing and Information Sciences, 2011; 2(7): 2079-8407.
[3]. Rijwani P., Jain S. Enhanced Software Effort Estimation Using Multi Layered Feed Forward Artificial Neural Network Technique. Procedia Computer Science, 2016;89: 307 – 312.
[4]. Kaushik A., Soni A.K., Soni R. A Simple Neural Network Approach to Software Cost Estimation. Global Journal of Computer Science and Technology Neural & Artificial Intelligence, 2013; 13(1): 0975-4350.
[5]. Srinivasa R.T, Prasad Reddy P.V.G.D, Hari Ch.V.M.K Fuzzy and Swarm Intelligence for Software Cost Estimation. Global Journal of Computer Science and Technology, 2011; 11(22): 0975-4350.
[6]. Prasad Reddy P.V.G.D, Sudha K.R, Rama S. P., Ramesh S.N.S.V.S.C Software Effort Estimation using Radial Basis and Generalized Regression Neural Networks. Journal of Computing, 2010; 2(5): 2151-9617.
[7]. Sehra S. K., Brar Y. S., Kaur N. EVOLUTIONARY COMPUTING TECHNIQUES FOR SOFTWARE EFFORT ESTIMATION. IJCSIT, 2017; 9(2): 123-130.
[8]. Kaushik A., Chauhan A., Mittal D., Gupta S. COCOMO Estimates Using Neural Networks. Intelligent Systems and Applications, 2012; 9: 22-28.
[9]. Goyal S., Padda Sh. Back Propagation Algorithm Based Model for Software Cost Estimation. International Journals of Advanced Research in Computer Science and Software Engineering, 2017; 7(6): 2277-128.
[10]. Bell D. Software Engineering for students. programming approach, fourth edition, 2004, 419.
[11]. Aljahdali S., Sheta A. F. Software Effort Estimation by Tuning COOCMO Model Parameters Using Differential Evolution. Taif University, Taif, Saudi Arabia (aljahdali@tu.edu.sa), 2010.
[12]. Boehm B. W. Software Engineering Economics. IEEE Transactions on Software Engineering, 1984; 10 (1): 641_686.
[13]. Mirjalili S., Mirjalili S. M., Lewis A. Gray Wolf Optimizer. Advances in Engineering Software, 2014; 69: 46-61.
[14]. Masadeh R., Alzaqebah A., Hudaib A. Grey Wolf Algorithm for Requirements Prioritization. Published by Canadian Center of Science and Education, 2018; 12(2): 1913-1844.
[15]. Vinothini J., Bakkiyaraj R. A. Gray Wolf Optimization Algorithm for Colour Image Enhancement Considering Brightness Preservation Constraint. International Journal of Emerging Trends in Science and Technology, 2016; 3(5): 4049-4055.
[16]. Rajakumar R., Amudhave J., Dhavachelvan P., Vengattaraman T. GWO-LPWSN: Grey Wolf Optimization Algorithm for Node Localization Problem in Wireless Sensor Networks. Journal of Computer Networks and Communications, 2017: 10.

[17]. Kaushik A. A Comparative Study on Fuzzy Approaches for COCOMO's Effort Estimation. International Journal of Computer Theory and Engineering, 2012; 4(6): 990-994.
[18]. Sheta A. F., Al-Afeef A. A GP Effort Estimation Model Utilizing Line of Code and Methodology for NASA Software Projects. IEEE, International Conference on Intelligent Systems Design and Applications, 2010: 290-295.
[19]. Kaur J., Singh S., Kahlon K. S., Bassi P. Neural Network-A Novel Technique for Software Effort Estimation. International Journal of Computer Theory and Engineering, 2010; 2(1): 1793-8201.