A Hybrid Approach of Firefly and Genetic Algorithms in Software Cost Estimation

Isa Maleki[1], Laya Ebrahimi[2], Farhad Soleimanian Gharehchopogh[3]

[1]Young Researchers and Elite Club, Urmia Branch, Islamic Azad University, Urmia, Iran

[2]Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

[3]Department of Computer Engineering, Hacettepe University, Beytepe, Ankara, Turkey

Abstract: *Precise cost estimation of software projects is an important topic for software companies. Precise estimation of required efforts, delivery time and project cost are considered as a major challenge for project managers. The purpose of Software Cost Estimation (SCE) is to increase the possibility of project success and to identify and evaluate systematic efforts and costs in software projects. The first model presented for SCE was the COCOMO model. According to the investigations, it can be said that the error value of effort and cost in software projects is high in this model. Therefore, in order to improve the performance of evaluation criteria of software projects, we exploited meta-heuristic algorithms. In this paper, a novel hybrid model based on Genetic Algorithm (GA) and Firefly Algorithm (FA) is presented. The performance of the hybrid model was evaluated on NASA93 software project database. The evaluation of the results show that the hybrid model has decreased the error of Mean Magnitude of Relative Error (MMRE) criterion and increased the accuracy of Evaluation Function (EF) criterion by %2.88 comparing to COCOMO model.*

Keywords: *Software Cost Estimation, COCOMO, Meta-Heuristic Algorithms, Firefly Algorithm, Genetic Algorithm*

1. Introduction

SCE has always been a major challenge in software engineering and in primary stages it contains a lot of ambiguities. Precise and primary estimation is virtually impossible due to the lack of comprehensive data to develop software systems and it is also very important when a project contract is signed and when the possibility of a project and its profitability is evaluated. Among the basic factors that result in the ambiguity and inaccuracy of SCE process are deficiency and ambiguity in requirements and also lack of data about previous projects, another factor is developed and untrained methods in data and tools [1]. One of the major purposes of software engineering community, is to develop effective methods which can precisely control the software development cycle and can estimate software development costs with acceptable accuracy. Recent SCE models, such as COCOMO [2, 3] and

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

function point analysis [4] do not provide precise estimations for the projects due to the linearity of the mathematical functions and inaccuracy of project factors. Therefore algorithm models used in cost estimation cannot show desirable performance due to the presence of ambiguity in the data.

Some of software development companies the efforts done in developing previous projects to maintained document and use those data in activities associated with improving effort estimation and achieving precise estimations in new projects [5, 6]. One reason why companies use data of previous project is that in better identification of the requirements, work experience of developers and the number of development team members can help them [7, 8]. SCE based on previous projects is a process in which the expert estimates the amount of effort and cost needed in new project based on his own idea which is generally based on previous experiences in development or management of similar projects and this estimation model is a techniques which is not likely to become the most common SCE method [9]. When software development companies decide to collect previous project data in order to expand them in to their new projects and use them in improving effort estimation quality, three major problems may arise [10, 11]:

- The time span needed to evaluate previous project data may be too long.
- When the dataset is large, tools and human forces which a company uses to develop its projects will vary and it is possible that in previous projects some tolls had been used that are not used today's projects.
- It is essential that the collected data be of the same type and not be contradictory, otherwise the results may be different and more effort and cost may be spent in software projects.

In order to reduce the error value in COCOMO model, a hybrid model using the hybrid of algorithms GA and FA for SCE is proposed which evaluation results show that the hybrid model has much better performance compared with COCOMO model and estimates the costs more precisely. Despite approximate values and error estimation in both models, it should be kept in mind that by using meta-heuristic algorithms, the error values have been significantly reduced.

We have organized the overall structure of the paper as follows: in Section 2, previous works in the area of SCE is described; in Section 3, we explain meta-heuristic algorithms; in Section 4, hybrid model will be described; in Section 5, we explain the evaluations and results of hybrid model; and finally in Section 6, conclusions and future works are presented.

2. Previous Works

When software project information are ambiguous and incomplete, cost estimation models based on artificial intelligence can look more promising. Such models generally work relying on collective knowledge. By investigating previously works in this area, one can point to Particle Swarm

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

Optimization (PSO) algorithm which was proposed to estimate the software effort [12]. Evaluation was done in KEMERER dataset. In order to test and train COCOMO model, PSO algorithms were used. The test results have shown that the proposed model had better accuracy compared to COCOMO model. The value of MMRE error in proposed and COCOMO models were 56.57 and 245.39 respectively. The most important factor affecting the development of software companies is precise estimation of cost and human force. In [13] a new model is proposed for SCE by combining Ant Colony Optimization (ACO) and GA algorithms and its evaluation was done on NASA60 dataset. ACO algorithm was used to train the data and GA was used to test the data. Fitness function of the proposed model is MMRE criterion. The test results have shown that by increasing generation number MMRE error value is decreased in the proposed model. The results on 10 projects of NASA60 dataset software projects show that the MRE error value was less in the proposed model compared to COCOMO model. Testing and training of NASA60 dataset software projects were performed using Artificial Neural Networks (ANNs) [14]. NASA60 dataset which consists of 11 projects among NASA60 dataset software projects show that the MRE error value in ANNs is less compared to COCOMO model. The results show that in more than 90% of the cases ANNs provided much better estimations compared to COCOMO model.

Recently data mining has become very common in SCE [15]. SCE has been simulated using Linear Regression (LR), ANNs, Support Vector Regression (SVR), K-Nearest-Neighbors (KNN) techniques. By using LR model dependencies of traits effective in SCE can be determined. LR model finds the relations between independent and dependent factors. ANNs, through training and testing the data, tries to improve the accuracy of SCE. SVR model has been used to optimize the effective factors in SCE. KNN is a technique in data analysis and is used to classify data in a set of data which were previously classified and their traits were determined. Using KNN the weight of factors effective in SCE is determined. Their test results show that SVR model has less MRE compared to other models. Based on GA methodology, a morphological hybrid has been presented for SCE [16]. Evaluation was done on Desharnais, NASA, COCOMO, Albrecht, Kemerer, and KotenGray datasets. The test results in all datasets show that the hybrid model based on GA had better improvement than different SVR models. In NASA dataset the value of PRED (25) in a hybrid model was 94.44% compared to 88.89% and 83.33% for different SVR models. In Desharnias dataset the value of PRED (25) in a hybrid model was 90% compared to 55% and 60% for different SVR models. In COCOMO dataset the value of PRED (25) in a hybrid model was 90.90% compared to 81.82% and 72.73% for different SVR models. In Albresht dataset the value of PRED (25) in a hybrid model was 75% compared to 58.33% and 66.66% for different SVR models. In Kemerer dataset the value of PRED (25) in a hybrid model was 73.33% compared to 60% and 60% for different SVR models. In KotenGray dataset the value of PRED (25) in a hybrid model was 94.12% compared to 88.24% and 88.24% for different SVR models. MMRE evaluation criterion in hybrid model had less linear error compared to different SVR models. Machine learning techniques such as SVR and MLP in combination with GA has been presented for SCE [17]. Optimization of input data factors and MLP and SVR methods' parameters are two major purposes in using GA. The evaluation was done on Desharnais, NASA, COCOMO, Albrecht, Kemerer, and KotenGray datasets. The test results in all datasets show that hybrid models based on GA had good improvements. MMRE evaluation criterion had less error in combined models compared to MLP and SVR models. PRED criterion also had higher accuracy in hybrid model. Precise programming for developing software projects results in performance increase, better resource usage and time saving.

In [18], by using PSO algorithm, a hybrid PSO-COCOMO model is proposed for SCE and its evaluation has been done on NASA18 dataset. The test results show that PSO-COCOMO has less MMRE error compared to other algorithm models and has less accuracy compared to Fuzzy Logic (FL). MMRE error values were 0.0074% and 0.0046% for PSO-COCOMO and FL respectively. Timely delivery of software is always a major concern for software companies. Optimizing software project factors in order to increase estimation accuracy has been done using GA [19]. Evaluations were done on ISBSG and IBMDP datasets. The weights of project factors were classified according to three methods: Unweight Analogy (UA), Unequally Weighted Analogy (UWA), Linearly Weighted Analogy (LWA) and Non-LWA. And for each class the optimized weight was found by GA. The test results of 132 projects in ISBSG dataset and 33 projects in IBMDP dataset, show that GA had good accuracy in evaluating PRED and MMRE criteria. The hybrid model of Function Link ANN (FLANN) and GA has been used for SCE [19]. The GA-FLANN hybrid model is a kind of three-layered feed forward network. GA algorithm was used in two hybrid models OFWFLANN and OCFWFLANN. OFWFLANN model was used to improve factors affecting the cost and OCFWFLANN model was used to train weights of FLANN vectors. The evaluation was done on NASA93 dataset. Train and test steps in order to decrease MMRE error on NASA93 dataset were 0.43 and 0.37 respectively. Also, train and test steps in order to improve accuracy of PRED (25) criterion were 0.46 and 0.39 respectively.

3. Meta-Heuristic Algorithms

Today application of meta-heuristic algorithms to achieve optimized solutions in optimization problems has had a significant growth [21, 22 and 23]. These algorithms use some qualitative parameters whose values are easily adjustable. Also the speed

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

of convergence of meta-heuristic algorithms is high in the possibility of finding the universal optimized solution. Therefore in solving optimization problems application of meta-heuristic algorithms is possible in achieving the almost optimized solution.

3.1 Firefly Algorithm

FA is a population-based algorithm that was introduced in 2008 [24]. In FA, first some artificial fireflies were randomly generated in the problem space. Then to each artificial firefly, an appropriate light intensity value was assigned using the value obtained for target function in that point. The way in which each firefly is given a light intensity value is such that by increasing the point optimization value of the firefly the intensity of the light was also increased. Fireflies with lower light intensity were attracted towards the fireflies with higher light intensity and this was continues until all fireflies were gathered in one single point that is probably the universal optimum point. The law of updating the movement of low-light fireflies towards high-light fireflies was done by using Eq. (1) [25].

$$x_i \leftarrow x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha(rand - \frac{1}{2}) \tag{1}$$

In Eq. (1) the values of α, β_0 and γ were assumed to be constant. α and β0 were chosen from interval [0,1] and γ was in interval [0,∞). Also r_{ij} was Euclidean distance between two fireflies which is defined by Eq. (2) [25].

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{n}(x_{i,k} - x_{j,k})^2} \tag{2}$$

The absorption coefficient of two fireflies is obtained from Eq. (3) [25].

$$\beta = \beta_0 e^{-\gamma r_{ij}} \tag{3}$$

In Eq. (3) β_0 is the maximum attraction and is in interval [0, 1]. Parameter γ is absorption coefficient and is in interval [0, ∞). Parameter r is the distance of two fireflies and its value can be calculated by Eq. (2). If $\beta_0=0$ each firefly will search the problem space on its own without the cooperation of other fireflies and the search will be random. Also, γ=0 will result in a random search in problem space.

3.2 Genetic Algorithm

GA is a meta-heuristic algorithm which was devised to find optimum solution and hybrid optimization problems [26]. GA produces a list of chromosomes for the optimization problems based on the primary population. Chromosomes approach the optimum solution based on selection, crossover and mutation techniques. In GA with the possibility of crossover, appropriate candidates for the solution are generated in the next generation. Base on mutation possibility chromosomes' genes change randomly. This process results in a new generation of chromosomes which are different from the previous generation. Evaluation in GA is done according to fitness function. Fitness function is determined according to the kind of the problem.

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

4. Proposed Model

Generally the values determined for the software projects are unknown and inaccurate. The purpose of this paper was to find precise values for software project factors. One of the critical factors for SCE, is the number of Source Line of Code (SLOC). Therefore, in this paper, by using the hybrid model we tried to evaluate and investigate the point that how different estimation factors affect the accuracy of SLOC estimations. Some other needed parameters in SCE are [27]: programming language, efficiency, performance, software process evolution, programming skill level, design and reuse, efficiency factors in product manufacturing, complexity, exploitation and timing. Some of these parameters have direct effect on each other. For example both code complexity and programming skill directly affect exploitation and timing. In this paper we proposed a hybrid model has been developed based on GA and FA algorithms for SCE of NASA93 dataset projects [28] which includes 15 Effort Multipliers (EMs). EMs factors have a critical role in effort and cost estimation. EM factors have a linear relation with effort and cost and their value is very effective in project success. In the hybrid model, we evaluate and analyze software criteria according to basic requirements of operational methods. In hybrid model basic software factors such as complexity and effort are evaluated all of which have key roles in improving effort and cost. Figure (1) shows the flowchart of hybrid model.

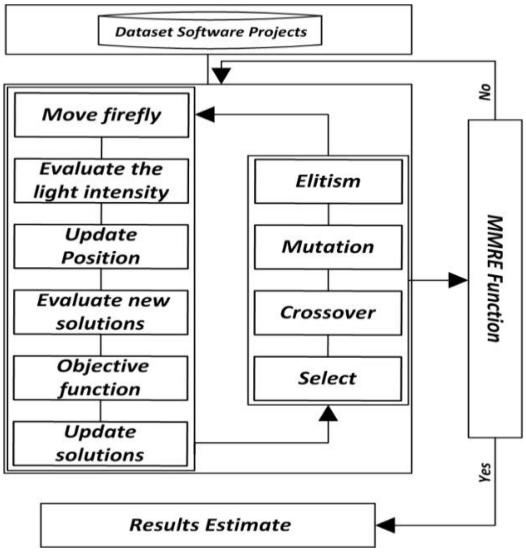

Figure. 1. Flowchart of hybrid model

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

In hybrid model with FA we have proposed an adaptive method to produce primary population in GA in which population genes change adaptively. Although the convergence speed of FA algorithm is good convergence of solutions before achieving absolute optimum can have a negative effect in solution algorithm. On the contrary, GA is slower than FA and there is less possibility that it converges towards local optimum, on the other hand because of chromosome structure, lack of convergence and failure in finding a solution or an inappropriate solution is impossible. The hybrid model adapts itself with continuous changes in FA and injects one of the best solutions found by FA to produce primary population into GA, and GA tries to achieve the best answer for cost factors through elitism and evaluate it in fitness function and presents the solution with the least error as the final solution. Quasi code of the hybrid model is shown in Figure (2).

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

01: Start
02: Input the 15 EMs, KSLOC, Actual Effort
03: Initialize Parameters
04: Repeat
05: Firefly Algorithm
  Initial population of fireflies: $x_i (i = 1,2,...,n)$
  Objective function of f(x), where $x = (x_1, x_2,..., x_d)^T$
  Determine the light intensity of firefly n of I_i *at* x_i *via* $f(x_i)$
  While (the termination criteria is not satisfied)
  For i= 1 : n
  For j= 1 : n
  If ($I_j > I_i$ *)*
    Then move firefly i towards firefly j by using Eq. (1)
    End if
  Attractiveness varies with distance r via $Exp[-\gamma r^2]$
  Evaluate new solutions and update Light Intensity
 End for j;
 End for i;
  Check the ranges of the given solutions
  Update them as appropriate
  Rank the fireflies
  Find the current best
 End while
 Find the firefly with the highest Light Intensity among all fireflies
 End FA
06: Genetic Algorithm
  Assign a fitness value to each chromosome
  Select the most fitting chromosomes
  Crossover
  Mutation
07: Evaluation Solutions
08: Loop
  For i= 1: n (no. of projects)
  EAF[i] = $EM_1 * EM_2 * ... * EM_{15}$
  Estimated Effort[i] = a[j](KSLOC[i] ^ b[j])*EAF[i]*
  MRE[i] = |Actual Effort[i]-Estimated Effort[i]|/Actual Effort[i]
  MMRE =MMRE+MRE[i]
  MMRE /=n
 End Loop
09: Evaluation MMRE Function
10: Until (the termination criteria is not satisfied)
11: Results Estimate
12: End

Figure. 2. Quasi Code of the Hybrid

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

Model

In the hybrid model MMRE was considered as fitness function. The purpose of fitness function in the hybrid model is to minimize MMER compared to FA and GA algorithms and COCOMO model. In the hybrid model this continues as long as the value of MMRE decreases desirably. Fitness function for the hybrid model is defined as Eq. (4) [29].

$$MRE_i = \frac{|act_i - est_i|}{act_i} \times 100 \quad (4)$$

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i \,, i = 1,2,\ldots,n \quad (5)$$

By using Eq. (5) we can compare total error obtained from estimation models. PRED is also an important criterion in estimation accuracy. The commonest method of investigation of prediction accuracy are MMRE and PRED. PRED(x) is defined as Eq. (6) [29].

$$PRED(x) = \frac{1}{n} \times \sum_{i=1}^{n} \begin{cases} 1, & if MRE \le x \\ 0, & otherwise \end{cases} \quad (6)$$

The criterion PRED(x) which is defined based on MRE is the commonest in estimation accuracy and is a good expression of how the models work. In the evaluation of estimation criteria, the model with lower MRE is better than the one with higher MRE and the model with lower MMRE is better than the one with higher MMRE. Also, the model with higher PRED is better than the one with higher PRED. In order to show the superiority of hybrid estimation models two criteria MMRE and PRED is defined as EF as in Eq. (7) [30].

$$EF = \frac{PRED(25)}{1 + MMRE} \quad (7)$$

5. Evaluation and Results

In order to evaluate the hybrid model NASA93 software projects dataset was used which is consisted of 15 EM factors. The algorithm simulations were conducted in VC#.NET 2010 programming environment. From accuracy point of view meta-heuristic algorithm performances depend on value of primary parameters. Therefore in order to reach the best estimation in the hybrid model parameter values were assigned based on test and repeat. Parameter value assignment was shown in Table (1).

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

Table 1. Parameter Values

Parameters	Value
No. Population	100
β0	1
γ	1
α	0.3
Pc	0.9
Pm	0.5
Elitism	20
No. Generation	50
Fitness Function	MMRE

In Table (2), 93 projects from NASA93 dataset software projects are evaluated and compared. The results of Table (2) show that the hybrid model reduces the MRE criterion compared to FA and GA algorithms. Therefore, the hybrid model is useful in cost estimation and contains less error compared to COCOMO model and GA and FA algorithms.

Table 2. Comparison of MRE of Models on 93 projects of NASA93 Dataset Software Projects

No. Projects	KSLOC	Actual Effort	MRE COCOMO	MRE GA	MRE FA	MRE Hybrid
1	0.9	8.4	72.12	25.56	17.41	9.13
2	2.2	8.4	24.12	10.26	8.11	5.14
3	3.5	10.8	0.73	3.65	2.89	1.23
4	6.2	12	638.09	420.13	386.20	256.74
5	5.5	18	1.19	0.95	1.80	0.55
6	6	24	56.16	19.88	25.49	15.41
7	9.7	25.2	33.25	21.28	17.63	12.68
8	7.7	31.2	16.90	8.11	11.35	5.75
9	8.2	36	22.72	12.53	9.38	7.66
10	11.3	36	22.23	10.44	7.16	5.41
11	3	38	2.46	1.45	3.86	0.65
12	6.5	42	23.13	12.93	9.42	7.20
13	8	42	15.66	5.47	8.39	5.71
14	10	48	35.22	15.68	7.62	10.52

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

15	15	48	38.84	13.22	10.83	9.44
16	20	48	15.58	5.93	8.17	7.53
17	10.4	50	27.39	16.08	12.13	6.30
18	13	60	1.69	5.87	3.68	2.55
19	14	60	15.59	7.33	8.45	3.23
20	19.7	60	23.74	9.79	12.79	5.11
21	32.5	60	134.17	96.14	74.24	36.68
22	31.5	60	24.65	10.19	11.52	8.12
23	12.8	62	18.47	12.04	9.72	7.41
24	15.4	70	11.17	3.11	5.37	2.81
25	7.5	72	36.85	19.84	15.93	12.39
26	20	72	54.19	27.65	23.30	18.01
27	34	72	85.09	32.46	28.58	21.38
28	16.3	82	19.19	10.18	4.61	6.64
29	15	90	30.87	13.25	10.16	9.20
30	165	97	1014.53	876.92	715.01	578.94
31	11.4	98.8	35.56	14.35	9.64	12.70
32	21	107	58.99	25.01	17.14	13.21
33	16	114	25.04	13.44	9.30	5.17
34	24.6	117.6	19.03	8.43	11.68	3.06
35	25.9	117.6	14.23	6.41	3.28	4.79
36	29.5	120	2.75	6.48	3.07	1.10
37	40	150	45.37	27.62	19.13	11.27
38	19.3	155	25.93	16.24	12.30	8.52
39	90	162	18.52	10.89	9.77	6.58
40	32.6	170	15.28	8.50	5.42	6.54
41	35.5	192	17.47	12.67	7.49	2.16
42	240	192	199.07	112.13	84.13	32.63
43	38	210	13.06	8.59	5.33	4.12
44	100	215	92.47	45.62	32.07	26.15
45	48.5	239	5.97	8.26	4.67	7.65
46	20	240	73.35	45.13	25.39	19.13
47	47.5	252	22.92	10.45	13.48	8.34
48	70	278	0.02	4.29	5.76	2.35
49	66.6	300	3.16	8.37	5.04	1.20
50	85	300	31.88	8.73	12.29	10.69
51	98	300	77.86	38.02	29.60	27.90

52	150	324	76.37	52.19	32.89	18.26
53	66.6	352.8	17.66	14.31	6.53	9.15
54	100	360	13.52	7.46	4.86	3.81
55	100	360	16.17	5.38	7.54	6.91
56	50	370	21.98	12.33	9.93	5.44
57	79	400	30.04	17.04	14.80	9.12
58	60	409	5.31	2.14	6.22	1.86
59	190	420	4.03	7.75	3.80	6.35
60	24	430	65.94	40.12	28.31	32.11
61	151	432	42.47	23.71	19.03	13.91
62	90	444	18.81	9.89	6.13	5.03
63	339	444	492.88	325.22	268.46	218.46
64	70	458	2.99	5.11	3.83	1.65
65	16.3	480	41.14	23.18	20.23	15.80
66	53	480	27.66	11.49	7.58	4.55
67	78	571.4	3.98	7.53	2.46	4.25
68	144	576	42.55	16.74	18.52	13.34
69	41	599	53.51	34.01	27.50	18.21
70	111	600	27.33	15.21	10.71	7.59
71	137	636	14.63	10.92	6.49	3.63
72	7.25	648	83.99	43.26	36.22	28.61
73	100	703	1.70	2.23	1.89	0.85
74	350	720	67.99	53.12	43.91	23.14
75	101	750	19.64	7.06	5.13	3.45
76	162	756	43.76	26.02	21.62	17.41
77	150	882	1.11	4.52	1.23	2.41
78	284.7	973	39.13	21.50	16.11	14.62
79	227	1181	4.71	2.13	5.16	3.41
80	352	1200	115.24	62.40	42.83	28.11
81	177.9	1248	1.79	3.41	2.22	3.10
82	32	1350	10.21	6.98	4.67	3.68
83	282.1	1368	16.70	7.60	11.33	5.74
84	70	1645.9	32.83	17.44	10.90	12.16
85	65	1772.5	34.40	19.62	15.07	8.24
86	50	1924.5	55.90	25.17	28.37	21.23
87	219	2120	28.79	17.67	12.01	9.15
88	302	2400	31.60	19.41	12.65	10.16

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

89	423	2400	52.34	28.01	21.81	14.28
90	271	2460	2.68	5.20	4.15	3.85
91	165	4178.2	14.89	5.26	8.91	3.74
92	980	4560	442.25	358.02	246.31	174.95
93	233	8211	34.48	18.01	13.16	8.90

The results of the hybrid model and GA and FA algorithms are shown in Table (3). According to criteria MMRE, PRED, and EF it can be seen that in all the best performance in estimation belongs to hybrid model.

Table 3. Evaluation of Models

Models	MMRE	PRED (25)	EF
COCOMO	58.80	51.61	0.86
GA	38.31	77.41	1.96
FA	30.84	80.64	2.53
Hybrid Model	22.53	88.17	3.74

The results of criteria of Table (3) show that the hybrid model has better performance compared to GA and FA algorithms. MMRE criterion in the hybrid model has lower error compared to COCOMO model and GA and FA algorithms. As can be seen in Table (3) the amount of MMRE error in the hybrid model is 22.53 compared to 38.31 and 30.84 in GA and FA algorithms respectively. Also, FA algorithm has better accuracy than GA algorithm. PRED criterion in the hybrid model has better accuracy than GA and FA algorithms. EF criterion in the hybrid model has better accuracy compared to the other models. This shows that the hybrid model has lower MMRE error and higher PRED accuracy. Evaluation of Table (3) shows that the hybrid model had increased the accuracy of EF criterion by 2.88% compared to COCOMO model. Figure (3) shows the diagram of MMRE criterion for the models.

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

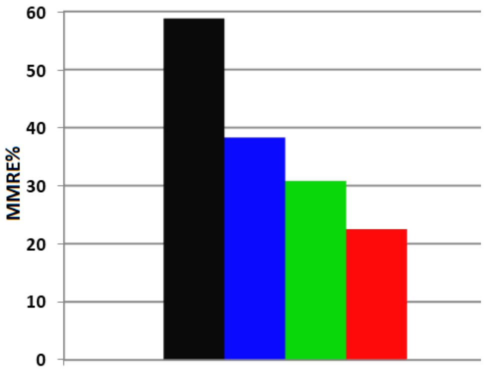

Figure. 3. Diagram of MMRE Criterion of the Models

6. Conclusion and Future work

The capability of precisely estimating the software costs is considered as an important activity for the project management team. Since cost factors as ambiguous and incomplete, software costs should be estimated especially in the primary steps of the life cycle of the development. Therefore providing precise cost estimation has a profound effect on economical processes including budgeting and contracting and decision making about the way the project is executed. A significant portion of SCE area consists of algorithm models. In these models mathematical formulas are used in estimating the cost and other input parameters. An alternative model for SCE is using meta-heuristic algorithms which use test and train in evaluating the estimation. In this paper hybrid model based on GA and FA algorithms is presented for SCE. Three criteria of MMRE, PRED (25) and EF were used to evaluate the performance. MMRE criterion showed that the hybrid model reduced the error from 58.80% to 22.53% and was capable of reducing the MMRE criterion of COCOMO model. The evaluation of the results on NASA93 dataset showed that the hybrid model had better efficiency according to performance of criteria. By presenting this paper we hope that in the future we can present other efficiency models for SCE by hybridizing other meta-heuristic algorithms.

REFERENCES

1. Singh B.K., Tiwari S., Mishra K.K., Misra A.K., “Tuning of Cost Drivers by Significance Occurrences and Their Calibration with Novel Software Effort Estimation Method”, Advances in Software Engineering, pp. 1-10, 2013.

2. Boehm B.W., “Software Engineering Economics”, Prentice-Hall, Englewood Cliffs, New Jersy, 1981.

3. Boehm B.W., “Software Cost Estimation with COCOMO II”, Prentice Hall PTR, Englewood Cliffs, New Jersy, 2000.

4. Albrecht A.J., Gaffney J., “Software Function, Source Lines of Code, and Development Effort Prediction: a software science validation”, IEEE Transactions on Software Engineering SE, 9(6), 639-648, 1983.

5. Attarzadeh I., Ow S.H., “A Novel Algorithmic Cost Estimation

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.50)

Model based on Soft Computing Technique," Journal of Computer Science, 6(2), 117-125, 2010.

6. Gharehchopogh F.S., Talebi A., Maleki I., Analysis of Use Case Points Models for Software Cost Estimation, International Journal of Academic Research, Part A, 6(3), 118-124, 2014.

7. Elish M.O., Helmy T., Hussain M.I., "Empirical Study of Homogeneous and Heterogeneous Ensemble Models for Software Development Effort Estimation", Mathematical Problems in Engineering, pp. 1-2, 2013.

8. Gharehchopogh. F.S, Ebrahimi. L, Maleki. I., Gourabi. S.J, A Novel PSO based Approach with Hybrid of Fuzzy C-Means and Learning Automata in Software Cost Estimation, Indian Journal of Science and Technology, 7(6), 795-803, 2014.

9. Wen J., Li S., Lin Z., Hu Y., Huang C., "Systematic Literature Review of Machine Learning based Software Development Effort Estimation Models," Information and Software Technology, 54(1), 41-59, 2012.

10.Kultur Y., Turhan B., Bener A., "Ensemble of Neural Networks with Associative Memory (ENNA) for Estimating Software Development Costs", Knowledge-Based Systems, 22(6), 395-402, 2009.

11.Elish M., "Assessment of voting ensemble for estimating software development effort," in Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM '13), 322-327, 2013.

12.Gharehchopogh F.S., Maleki I., Khaze S.R., "A Novel Particle Swarm Optimization Approach for Software Effort Estimation", International Journal of Academic Research, Part A, 6(2), 69-76, (2014).

13.Maleki I., Ghaffari A., Masdari M., "A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization", International Journal of Innovation and Applied Studies, 5(1), 72-81, 2014.

14.Gharehchopogh F.S., "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, 15-18 June 2011.

15.Khalifelu Z.A., Gharehchopogh F.S., "Comparison and Evaluation Data Mining Techniques with Algorithmic Models in Software Cost Estimation", Elsevier, Procedia-Technology Journal, Vol. 1, pp. 65-71, 2012.

16.de A. Araújo Ricardo, Soares S., Oliveira A.L.I., "Hybrid Morphological Methodology for Software Development Cost Estimation", Expert Systems with Applications, Vol. 39, pp. 6129-6139, 2012.

17.Oliveira A.L.I., Braga P.L., Lima R.M.F., Cornélio M.L., “GA-based Method for Feature Selection and Parameters Optimization for Machine Learning Regression Applied to Software Effort Estimation”, Information and Software Technology, Vol. 52, pp. 1155-1166, 2010.

18.Sheta A.F., Ayesh A., Rine D., “Evaluating Software Cost Estimation Models using Particle Swarm Optimization and Fuzzy Logic for NASA Projects: a Comparative Study”, International Journal Bio-Inspired Computation, 2(6), 365-373, 2010.

19.Sun-Jen Huang, Nan-Hsing Chiu, “Optimization of Analogy Weights by Genetic Algorithm for Software Effort Estimation”, Information and Software Technology, 48, pp. 1034-1045, 2006.

20.Benala T.R., Dehuri S., “Genetic Algorithm for Optimizing Functional Link Artificial Neural Network Based Software Cost Estimation”, Proceedings of the InConINDIA, pp. 75-82, Springer-Verlag, 2012.

21.Gharehchopogh F.S., Maleki I., Farahmandian M., “New Approach for Solving Dynamic Traveling Salesman Problem with Hybrid Genetic Algorithms and Ant Colony Optimization”, International Journal of Computer Applications (IJCA), 53(1), 39-44, 2012.

22.Khaze S.R., Maleki I., Hojjatkhah S., Bagherinia A., “Evaluation the Efficiency of Artificial Bee Colony and the Firefly Algorithm in Solving the Continuous Optimization Problem”, International Journal on Computational Science & Applications (IJCSA), 3(4), 23-35, 2013.

23.Gharehchopogh F.S., Maleki I., Zebardast B., “A New Solutions for Continuous Optimization Functions by using Bacterial Foraging Optimization and Particle Swarm Optimization Algorithms”, Elixir International Journal Computer Science and Engineering (Elixir Comp. Sci. & Engg.), 61, pp. 16655-16661, 2013.

24.Yang X.S., “Nature-Inspired Meta-heuristic Algorithms”, Luniver Press, 2008.

25.Srivatsava P.R., Mallikarjun B., Yang X.S., “Optimal Test Sequence Generation using Firefly Algorithm”, Swarm and Evolutionary Computation, Vol. 8, pp. 44-53, Elsevier B.V, 2013.

26.Holland J., “Adaptation in Natural and Artificial Systems”, University of Michigan, Michigan, USA, 1975.

27.Menzies T., Port D., Chen Z., Hihn J., “Validation Methods for Calibrating Software Effort Models”, ICSE ACM, USA, (2005).

28.http://promise.site.uottawa.ca/SERepository/datasets/cocomonasa_2.arff. [Last Available: 2014.08.31]

29.Mac Donell S.G., Gray A.R., “A

Comparison of Modeling Techniques for Software Development Effort Prediction", in Proceedings of International Conference on Neural Information Processing and Intelligent Information Systems, pp. 869-872, (1997).

30. Ara´ujo R.D.A., de Oliveira A.L.I., Soares S.C.B., "A Morphological-rank-linear Approach for Software Development Cost Estimation," in Proceedings of the 21st IEEE International Conference on Tools with Artificial Intelligence (ICTAI '09), pp. 630-636, 2009.