	International Journal of Academic Research in Computer Engineering Volume 1, Number 1, Pages 05 - 15, September 2016 www.ijarce.org	***IJARCE***

Ant Colony based Metaheuristic Algorithms for Software Cost Estimation

Isa Maleki[1, *], Laya Ebrahimi[2], and Mitra Khanjari Japelaghi[3]

[1,2]Young Researchers and Elite Club, Urmia Branch, Islamic Azad University, Urmia, Iran.
[3]Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran.

Receive Date 15 May 2016; Accepted Date: 12 July 2016, Published Date: 15 September 2016

*Corresponding Author: maleki.misa@gmail.com (I.Maleki)

Abstract
In software development and software project management, Software Cost Estimation (SCE) will be considered a major step in the start of projects. SCE is one of the main activities at the decisions of software's time and expense management which has a special status in a software project. SCE in software development is considered as a key parameter in software project management. Therefore, to achieve the basic goals requires accurate and reliable cost estimate. Actual estimate in software development is based on effective factors that its accurate value should be recognized using algorithmic models and Artificial Intelligence (AI). Boehm used COCOMO model for SCE which is an algorithmic model in 1981. Algorithmic models such as COCOMO are based on criteria such as the number of lines of code or the Function Point (FP). In COCOMO model, project development and then the cost is calculated by such units. Therefore, the lower accuracy and unreliability of the algorithmic models creates a substantial risk in software projects, so, regularly estimating the cost throughout the project is necessary and it should be compared with other techniques. In the meantime, meta-heuristic algorithms in recent years have made good progress in the area of software and it has been used widely in SCE. Among meta-heuristic algorithms, Particle Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony (ABC), and Ant Colony Optimization (ACO) used to optimize the issues based on population and they have good effects in optimizing estimation factors. In this paper, a hybrid model DE-ACO, PSO-ACO and ABC-ACO based on ACO algorithm have been proposed for optimization based on effective factors in COCOMO model. Test results show that hybrid models have less magnitude of relative error (MRE) and Mean MRE (MMRE) in estimating software project cost in comparison with COCOMO model.

Keywords: Software Cost Estimation, COCOMO, Meta-Heuristic Algorithms, Optimization

1. Introduction

The main factor in completing the software project in accordance with planned cost and time schedule is the use of accurate estimation. SCE subject is an important issue in software development and management that should be considered over other things and software development projects should be based on it. Today, the problem of estimating the total cost of the project, including resource planning, estimating costs, human resources and cost control has become the major concern of software companies [1, 2]. Also, the number of software projects that after spending a lot of cost fails and basically the developer teams that cannot get the software projects to the primary distinguished purposes is growing. Problems and issues arising from the incomplete implementation or non-compliance of the project management tips can impose heavy cost to the software companies and even in some cases leads to failure of the software project [3, 4]. With a more extensive range of software project development and technologies very fast changes, accurate estimate is of the companies needs and software teams to achieve a performance. There are many projects that did not last long due to the lack of accurate estimates and therefore need to estimate the cost of software projects is very impressive. Due to fundamental differences between software projects such as being intangible, complex and changing needs, the cost effective factors should be identified to estimate.

Before the design and implementation of the software project, providing the estimation model for them is essential and it is the most difficult part of the software project development. In the process of the production of software projects, estimate must be taken to reduce cost and schedule risk in order to avoid project failure [5, 6]. Therefore, in the process of the production of software projects, cost estimation is of extra estimation. Also software project managers are eager to highly qualify their products by including factors like cost and time. So software project success depends on accurate identification of SCE. It can be said that without an accurate estimate, the cost of software projects will fail in design and in developmental phases of the program. For the first time, Boehm first model, COCOMO, which is an algorithmic model to estimate the cost software, was used in[7, 8].

In algorithmic models, software project development and cost and time estimates depend on their effort multipliers (EMs) [9]. Therefore, determining the value of multiplier in estimations is vital. In this paper, new models to SCE with a hybrid of meta-heuristic algorithms and ACO are provided. We used NASA60 dataset software projects to evaluate the proposed models. Meta-heuristic algorithms have been used for solving many problems [10-15]. In order to accurately assess software project's cost estimating, hybrid models with COCOMO algorithmic model have been compared and discussed.

The structure of this paper is as follows: in Section 2, we will introduce the previous works; in Section 3, we will introduce the meta-heuristic algorithm models; in Section 4, we will explain the hybrid models; in Section 5, we will explain the results and evaluation of hybrid models and finally in Section 6, we will explain the conclusions and future works.

2. Related Works

Estimating of software projects using COCOMO algorithm models the possibility of error is too high, with software development and as a result, SCE is very difficult. Research shows that implementation of the AI technique often performs better than algorithmic models.

Maleki et.al. proposed a new model for the SCE using a hybrid model of GA and ACO algorithms for 60 projects from NASA60 software projects dataset [16]. In their model, factors affecting the estimation have been tested using GA and have been trained using ACO. Experimental results show that the proposed model has better performance in compared with COCOMO model in cost estimation and has less MRE than the COCOMO model. Soft computing techniques to deal with ambiguity and factors affecting the SCE investigated [17]. The main goal of this paper is examining the role of fuzzy logic (FL) to improve the accuracy of the COCOMO II model with determining input parameters using the membership functions of Gaussian, Trapezoidal, and Triangular. The NASA93 dataset is used to evaluate the COCOMO II model and after the analysis of the results, it can be seen that the cost estimations using Gaussian membership function offered better results in comparison with other models. The results show that after applying the Gaussian membership functions and the fuzzification scale factors and all 17 cost factors in COCOMO II accuracy estimation is more highly optimized. And with fuzzification the MRE error is much more reduced. The main objective in SCE is software development in accordance with cost and time estimation. Hybrid model based on morphological-rank-linear (MRL) and GA is proposed for the SCE [18]. GA purpose is optimizing MRL parameters to achieve accurate estimates. Evaluation was done on 6 datasets and has been considered for comparison of two metrics MMRE and percentage relative error deviation (PRED) (%25). Comparisons of results show that hybrid model has less MMRE error value in comparison with SVR, SVR-RBF and MRL.

Techniques such as Artificial Neural Networks (ANNs), Liner Regression (LR), Multiple LR (MLR), Bayesian Network (BN), fuzzy decision trees (FDTs), FL, Neuro Fuzzy (NF) have been studied for SCE [19]. Weight is given to each of input factors using ANNS and their values have been estimated by hidden layers and their optimum value have been gain in output layer. In the MLR and BN models, estimation is done using analysis of interdependent variables. In FDTs, FL and NF models, for each of the estimated factors is considered a fuzzy membership domain. ANNs to estimate software project's cost is used [20]. In this paper, 11 projects from 60 projects in the NASA dataset have been tested and trained using ANNs and have been compared with COCOMO models and it has been shown that the error rate in COCOMO models is often more than ANN model. The results show that over 90% of the time ANN model provides a much better estimate than COCOMO model. Therefore, we can conclude that methods based on AI as a supplement can be an alternative to the algorithmic approach. Data mining techniques have good accuracy in SCE [21]. One of the most critical issues in software development is accurately estimating software cost. SCE have been modeled using technique like LR, ANNs, support vector regression (SVR) and K-Nearest-Neighbors (KNN). It can determine dependency effective properties using LR model in SCE. LR model will find the relationship between the independent factors and other dependent factors among the data. ANNs tries to SCE accurately with training and testing the data. SVR models are used for optimizing effective factors in SCE. KNN is a technique in data mining that is used for categorizing a set of data that has been classified before and their characteristics have been specified. Using KNN, the weight of effective attributes has been determined in SCE. Their results show the SVR model has less MRE than other models.

One of Common models in SCE is COCOMO model. Although it may not have enough accuracy with grow in size and complexity of software projects. In [22] a new COCOMO model based on Morphological-Rank (MR) has been proposed for the SCE. This model can estimate well incomplete and ambiguous input data and can be said to be reliable for SCE. For evaluating and the results, NASA dataset's software projects have been used. Experimental results show that the MR model has accurate predictions as well as less MMRE error value in comparison with the actual model. For estimating software projects' effort, SVR has been used [23]. NASA dataset software projects have been used to evaluate the results. Experimental results show that SVR model in comparison with the actual models are more accurate in prediction and also the amount of MMRE error has been less than them. In order that well-illustrated the performance of SVR model, the results are compared with Radial Basis Functions ANNs models and observed that in most cases, a SVR model is able to predict more accurately. For SCE application, FL has attracted the attention of the researchers [24]. A significant portion of the SCE is specified to the algorithmic models. In these models, mathematical formula is used for SCE based on criteria and other involved factors in the projects. FL rules are based on a series of if-then rules that examines the relationship between the variables. Thus, FL is an efficient method for showing approximate and ambiguous issues. In practice, FL is effective when the system for analysis by algorithmic techniques is not appropriate or the information that is available is incomplete and ambiguous. COCOMO81 dataset which contains 63 projects has been used in this study. Test results show that the precision of the estimates in FL is better than the COCOMO model. COCOMO model in three modes of Intermediate COCOMO, Detailed COCOMO and Basic COCOMO has been compared with FL. In models of the intermediate COCOMO, detailed COCOMO and the basic COCOMO average absolute error are %78.146, %99.498 and %464.659 respectively. While in FL is equal to %45.575. Also the average relative error (ARE) in model of Intermediate COCOMO, Detailed COCOMO and Basic COCOMO is %0.188, %0.188, and %0.602 respectively and in FL equal %0.137.

In [25] wavelet neural networks (WNNs) has been used for SCE. MORLET and GAUSSIAN functions are used as the objective functions for WNN and also is intended a threshold to train the network. Also WNN model is compared with radial basis function network (RBFN), multilayer perceptron (MLP), Dynamic evolving NF inference system, and MLR and support vector machine in terms of MMRE. Experiments have been conducted on the IBM dataset. WNN-Gaussian model has less MMRE error in comparison with other models, and generally it is said that WNN model is an effective method to improve the accuracy of the SCE. The results show that the performance of WNN for SCE is considered one of possible method to improve the accuracy of estimates. One of the common models in SCE is the COCOMO model. Although it may not be accurate enough with grow in size and complexity of software projects. COCOMO model based on feed-forward neural network is proposed for the SCE [26]. It can evaluate well the incomplete and ambiguous input data and is quite reliable in SCE. In this study, the proposed model will be tested and trained hidden layers of input data. To evaluate the proposed model in this paper, COCOMO II dataset used which contains 63 projects. MMRE factor is used to check. Proposed model is able to reduce MMRE value in COCOMO II model from 0.85 to 0.41. Conducted survey showed that ANNs has high accuracy in SCE and the result of estimated cost is very close to actual costs. There are various models for SCE.

A new model based on FL which called FECSCE has been proposed for the SCE [27]. COCOMOII model only considers projects' factors while, specification of development team are also is considered as important factors. In FECSCE model in addition to the characteristic of the project, experiences of individuals, skills of individuals, and the extent of the development of the group was also examined. In this model, the Multi Agent System (MAS) is used for modeling communication of the individual in team. Existence of the experienced team is considered an important element in the success of a complex software project. FL and MAS are used for modeling personal characteristics and interactions of people in the group. The main purpose of the FECSCE model is the review of characteristics of the team in order to improve accuracy of the model in COCOMO II for SCE. Experimental results show that FECSCE model has better performance in estimating PM than COCOMO II model.

The model base on FL and system Takagi-Sugeno for SCE proposed [28]. Experiments have been done on software projects of NASA dataset. The results show that Takagi-Sugeno model better estimation than the other models. In the past several years, algorithmic models such as COCOMO were used widely for SCE. New approach based on regression based classifier proposed [29]. Used for the categorizing of the training and testing samples of the software projects of NASA dataset. The results of the comparisons of COCOMO and regression models show that in most cases training and testing regression provides better results in a more accurate estimate. The accuracy of implementation in training set and testing set are 95% and 85 % respectively and relative absolute error for training data was equal to 84 % and for testing data was equal to 24.18 % respectively. Results show that the regression model decreases the total relative error and is considered a suitable method for measuring COCOMO parameters. Estimates from the regression model for the initial steps of the project are closer to the actual amount of effort. A new model based on Multi-Objective PSO (MOPSO) algorithm is proposed for SCE [30]. To test and evaluate MOPSO model, COCOMO model has been used. Experimental results show that the MOPSO models perform better in estimation than the COCOMO model and its mean absolute relative error is less.

Based on done works, you can say that investigated models are reliable and valid in comparison with algorithmic models. Also you can conclude that all of these models cannot reach to sufficient accuracy against evaluation and accuracy data. It can conclude that the machine learning techniques and AI are appropriate for the accuracy of prediction and yet there is no proof that all the predictions of the estimation model are always right.

3. Meta-Heuristic Algorithms

The meta-heuristic algorithm searches the total space for solving optimization problems in group and in parallel mood. These algorithms are accurate and complete, to wit, if the problems contain answers, the algorithm ensures to find the answer. This algorithm starts from an empty answer and leads the answers step by step towards an optimal or near-optimal answer. The major benefit of these algorithms is that if a wrong choice happens in the beginning of the search operation, this wrong action detected by algorithms operators and optimized and as a result, the time of algorithm's execution decreases and its performance increases. In applications where time is critical and it is important to find some answers to the problems, these algorithms have good performance. Meta-heuristic algorithms with the benefit of both the diversity and cooperation search optimization problems space to achieve optimal mode. So, the more the power of the algorithm to control these two parameters the more the algorithm ability to find near-optimal solutions to problems will grow. Meta-heuristic algorithms that solve optimization problems based on population can be tools for finding near-optimal solutions. In this section, we introduce the algorithms ACO [31], DE [32], PSO [34] and ABC [35] which are the most important population algorithms.

ACO algorithm is one of meta-heuristic algorithms which were presented in 1996 [31]. ACO algorithm is inspired by ant's natural life. Ants are capable of producing an adorable substance called pheromones. Although it is evaporation soon but it remains on earth as ant footstep for a short time. Ants are capable to find the shortest path to food with producing pheromones. Ants that choose the shortest path than those who choose a longer path are creating the more pheromone. Since most pheromones, attract more ants, more and more ants choose a shorter route, so all ants find the shortest path and travel from that path. To further investigate this issue, we assume for example, there are two paths to food sources that vary in length. Ants select both directions with equal probability. Ants that selected the shorter route and returned and are produced most pheromones returned earlier than others.

DE algorithm is a probabilistic search algorithm based on a population that tries to solve the problems [32]. This algorithm uses the distance and direction information from the current population to carry the search operation. The advantages of this algorithm is speed, setting the parameters, its effectiveness in finding optimal solutions, paralleling, high accuracy and it does not require sorting or matrix multiplication. In order to find the optimal solutions, DE algorithm has the ability to efficiently coordinate axes variables and also change the coordinate axes are in the right direction. DE algorithm starts evolutionary search process from a random initial population. Three operators of mutation, selection and crossover and three control parameters including the number population, scale factor and the possibility of crossover in the DE algorithm are very important.

PSO algorithms inspired by social behavior of birds that live in large and small groups were introduced in 1995 [34]. PSO algorithm is a simulation of the social behavior of a group of birds will follow food in an environment. None of the birds have any information about the location of food, but in each phase they know how much distance they have from the food. Accordingly, the best approach to find food is to follow the closest bird from food. PSO algorithm is a population algorithm in which the numbers of particles that are solutions of a function or an issue form a crowd. Population of particles moving in the space based on their individual experiences and collective experiences trying to find the optimal solution in search space. The PSO algorithm as an optimization algorithm provides a search based on population where each particle may vary its position over time. In PSO algorithm, the particles in a multi-dimensional search space move from possible solutions of problem. In this space an assessment criteria are defined and measurement of quality of solutions to the problem done through it. Change of the mode of each particle in a group influenced by their experience or knowledge of neighbors and behavior of a particle search is affected by other particles. This simple behavior causes to find the optimal areas of the search space. So in PSO algorithm, each particle so as to find the optimal position, properly inform it to the other particles, and each particle based on the obtained values for cost function with a certain probability will decide to adhere to other particles and search space using prior knowledge of the particles. This action causes all the particles do not be too close to each other and to effectively solve continuous optimization problems.

ABC algorithm inspired by social insects was invented in 2005 for the solution of optimization problems [35]. The algorithm simulates the behavior of a group of bees searching for food. Honey bees can distribute in far distances and diverse directions to exploit food sources. In the ABC algorithm, artificial bees are divided into three categories 1) Employed bees; 2) Onlooker bees; 3) Scout bees.

4. Hybrid Models

SCE algorithmic models work based primarily on cost factors and scale factors. Also estimation in algorithmic models depends on the size of the project and change in the size of the project will lead to numerous changes in the rate of cost and time. Incorrect calculations and estimations in cost factors will result in major changes in final results. Cost estimation and how to implement software projects in the early rounds is very challenging and often very difficult to model. This problem is due to the fact that important information on cost estimates is vague and imprecise and sometimes variable. COCOMO model is not able to use vague and imprecise information. This inefficiency prevents extraction of useful information that can dramatically improve the cost estimates by the model. Hybrid models try to minimize estimation error with optimizing effective factors in software projects. We test effective factors in the cost estimates in the hybrid models that are very important in COCOMO models based on the size of Kilo Source Line Of Code (KSLOC) projects and effective estimation factors and train parameters optimization operation using the optimized colonies of ants. Dataset NASA60 have includes 15 EMs [9] factors such RELY, CPLX, STOR, TOOL, SCED and etc. are which very effective in NASA dataset software projects and will be tested and assessed using hybrid models.

4.1. DE-ACO Hybrid Model

The DE algorithm randomly generates a number of primitive answers. In DE algorithm selecting the optimal vector algorithm has a significant impact on the final answer, from obtained vectors only the vectors that have acceptable fitness to be chosen to evaluate the optimal solution. Vectors will be selected randomly based on the probability of fitness for generating the next generation. In the next generation's production each of the operators applied randomly to selected vectors during the production of each generation and the best vector is selected. In the hybrid model, ACO algorithm increases the performance of DE algorithm in terms of the speed of convergence to the optimal solution. Flowchart of DE-ACO hybrid model is shown in Figure (1).

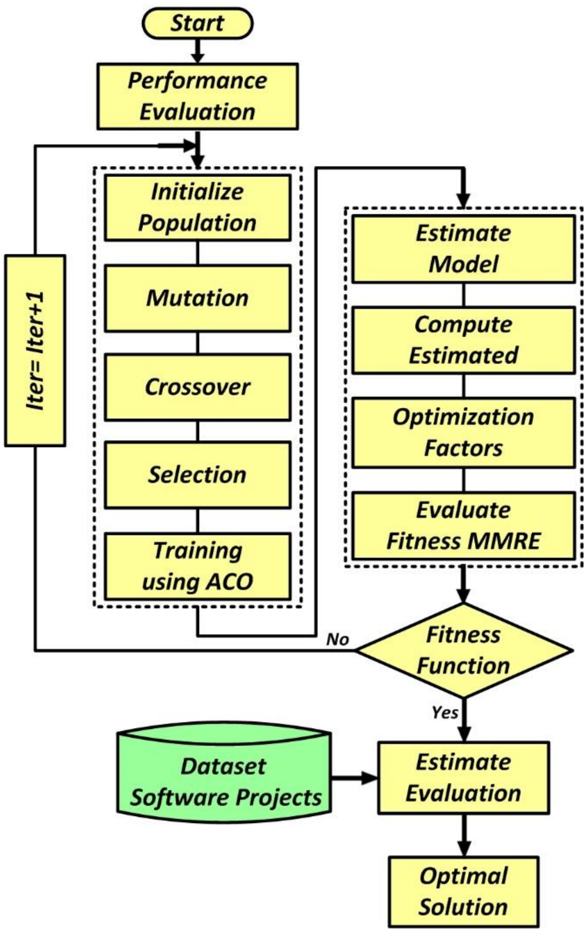

Figure 1. Flowchart of the DE-ACO Hybrid Model

In Figure (2) quasi code of DE-ACO hybrid model is shown.

1. Start 2. Initialize the Parameters 3. Evaluation the Population using Fitness Function 4. Loop (Mutation, Crossover, and Selection) 5. Testing DE 6. Training ACO 7. Compute Estimate 8. Evaluate Fitness (MRE=|Estimate – Actual| / Actual; MMRE=∑ (MRE) / N;) 9. While (a termination condition is reached) 10. Estimate Evaluation 11. Results Estimate

Figure 2. Quasi Code of the DE-ACO Hybrid Model

4.2. PSO-ACO Hybrid Model

In PSO algorithm, at first a number of random responses in the interval of effective factors in estimating are created. In PSO algorithm to select the optimal particle has a significant impact on the algorithm's final response, from the obtained particles only the particles are selected to evaluate the optimal solution which has an acceptable fitness. Therefore, we used updating pheromone in ACO algorithm to change the position of the particle toward the best answer in order to escape from local optimum solution. Flowchart of PSO-ACO hybrid model is shown in Figure (3).

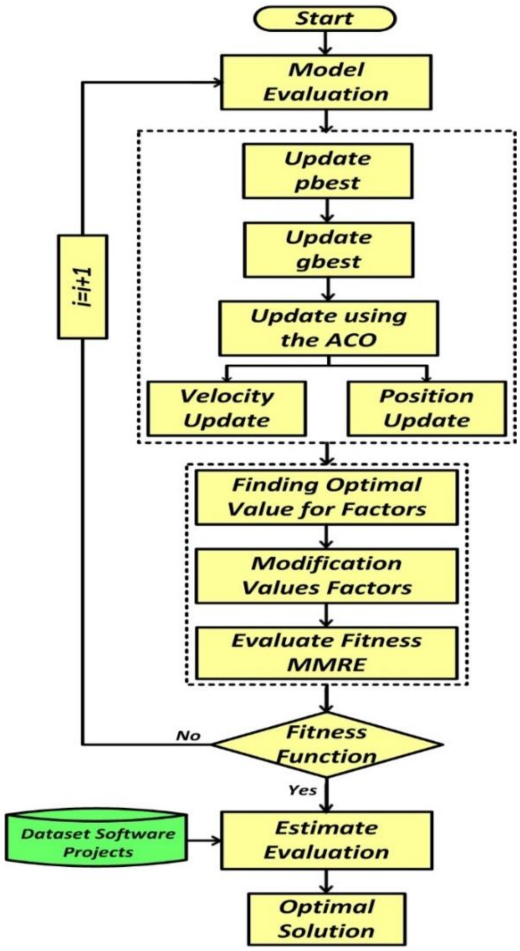

Figure 3. Flowchart of the PSO-ACO Hybrid Models

In Figure (4) quasi code of PSO-ACO hybrid model is shown.

1. Start 2. Initialize Parameters 3. Performance Evaluation 4. Loop (Update pbest and Update gbest) 5. Update each Particle's Velocity and Position using the ACO 6. Testing PSO 7. Finding Optimal Value for Factors 8. Modification Value Parameters Factors 9. Evaluate Fitness (MRE=|Estimate – Actual| / Actual and MMRE=∑ (MRE) / N;) 10. While (stop criteria not met) 11. Estimate Evaluation 12. Results Estimate

Figure 4. Quasi Code of the PSO-ACO Hybrid Model

4.3. ABC-ACO Hybrid Model

In ABC algorithm to escape from local optimum and to coverage wide areas of the solution space, an indicator is introduced as indicators of convergence. This index is equals to the ratio of the fitness function of the best solution in the population to the mean fitness functions in the general population. If the rate of the index increases, it is reflecting the risk of exposure in a local optimum. Thus, the hybrid model of the ACO algorithm is used to train well the ABC algorithm. Flowchart of ABC-ACO hybrid model is shown in Figure (5).

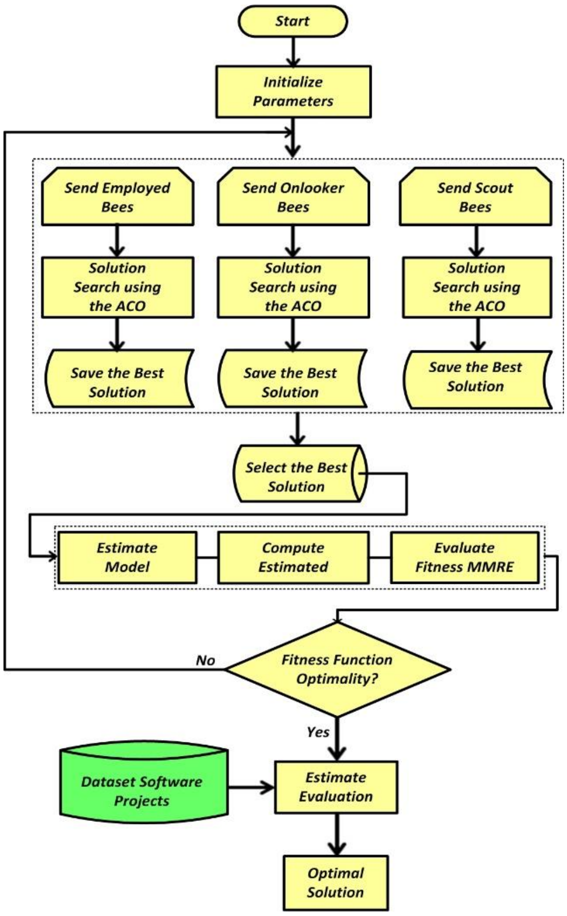

Figure 5. Flowchart of the ABC-ACO Hybrid Models

In Figure (6) quasi code ABC-ACO hybrid model is shown.

1. Start
2. Initialize Parameters
3. While (a termination condition is reached)
4. Train ACO
    4.1. For all bees do (Finding Best Solution and Create Best Solution by using Information Sharing)
    4.2. The employed bees phase
    4.3. The onlooker bees phase
    4.4. The scout bees phase
    4.5. Save the Best Solution
5. End While
6. Evaluate Fitness (MRE=|Estimate – Actual| / Actual; and MMRE=∑ (MRE) / N;)
7. Estimate Evaluation
8. Results Estimate

Figure 6. Quasi Code of the ABC-ACO Hybrid Models

In hybrid models, MMRE has been considered as the fitness function. The purpose of the fitness function in hybrid models is to minimize MMRE value in comparison with COCOMO algorithmic model and hybrid models iterated until MMRE value is decreased to an optimum level. MMRE be defined according to equation (2) [24].

$$MRE_i = \frac{|Actual_i - Estimate_i|}{Actual_i} \times 100 \qquad (1)$$

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i \qquad (2)$$

Using equation (2) can compare the sum of error between hybrid models and COCOMO model. Also PRED is an important criterion in careful prediction. The most common methods of accurate review of prediction are MMRE and PRED. PRED (x) is defined according to equation (3) [24].

$$PRED(x) = \frac{1}{n} \times \sum_{i=1}^{n} \begin{cases} 1, & if \quad MRE \le x \\ 0, & otherwise \end{cases} \qquad (3)$$

5. Evaluation and Results

In this section, the obtained results of the hybrid models and COCOMO model has been evaluated. In meta-heuristic algorithms for the evaluation of results, setting the initial parameters is very important. Meta-heuristic algorithms in comparison with their parameters are very sensitive and setting the parameters can have a significant impact on their performance. So, setting the parameters will result in their more flexibility and performance of hybrid models. In meta-heuristic algorithm, selection of initial populations is very important. If population be small, the problem leads to earlier convergence and we will not reach to intended solution or global near optimized solution and if the population be big, much time is needed so that algorithm reach to convergence. Therefore, we must limit the number of issues is appropriate and proportionate in order to achieve the optimal solution. Table (1) shows the parameters that have most effect on the performance of the hybrid model.

Table 1. Parameter Values

Hybrid Models	Parameters	Values
DE-ACO	No. Population	100
	F	0.5
	Pm	0.3
	Pc	0.3
	Pheromone Rate	0.9
	No. Generation	10
PSO-ACO	No. Population	100
	C1	1.5
	C2	1.5
	W	0.4
	Pheromone Rate	0.9
ABC-ACO	No. Population	100
	Φ	0.1
	Pheromone Rate	0.9

In Table (1), the parameter F is a fixed constant positive that is used to integrate the mutation rate. Parameters Pm and Pc are the mutation rate and the crossover rate respectively. C1 and C2 parameters will assist to find the optimal learning particle. W parameter is inertial weight in order to balance a particle's speed. Φ parameter on ABC-ACO hybrid model is used to balance position among vectors. The initial population in DE-ACO, PSO-ACO and ABC-ACO hybrid models is equal to 100. Also in all of the models combined rates of pheromones is considered to be 0.9.

In Table (2), a total of 60 projects of NASA dataset software project have been evaluated and compared. The results in Table (2) show that the hybrid models compared to COCOMO model are lowered MRE criterion. Thus, DE-ACO, PSO-ACO and ABC-ACO hybrid models are useful for estimating the error and have less estimation error than the COCOMO model.

Table 2. Comparison of MRE of Hybrid Models with COCOMO Model

No.	KSLOC	Actual Effort	MRE COCOMO	MRE DE-ACO	MRE PSO-ACO	MRE ABC-ACO
1	2.2	8.4	24.15	8.04	5.05	3.47
2	3.5	10.8	3.95	5.66	7.44	1.34
3	5.5	18	7.36	3.60	1.03	2.35
4	6	24	58.88	17.52	23.89	7.28
5	9.7	25.2	20.05	8.72	2.19	3.95
6	7.7	31.2	23.91	11.70	5.09	2.84
7	11.3	36	30.83	9.78	11.30	6.37
8	8.2	36	29.55	5.28	2.45	3.19
9	6.5	42	28.22	16.69	9.10	5.32
10	8	42	22.22	5.55	3.99	7.65
11	20	48	27.21	8.39	12.33	4.52
12	10	48	41.66	11.06	5.29	3.67
13	15	48	46.19	16.78	9.06	8.54
14	10.4	50	34.90	7.33	11.29	5.98
15	13	60	9.36	7.60	6.04	3.10
16	14	60	25.88	8.74	10.50	7.80
17	19.7	60	6.10	3.10	8.76	2.30
18	32.5	60	93.91	23.12	17.63	12.39
19	31.5	60	3.81	4.54	1.08	5.42
20	12.8	62	27.96	11.48	5.08	9.65
21	15.4	70	22.51	8.89	10.77	5.95
22	20	72	60.76	19.79	7.90	4.02
23	7.5	72	41.75	9.73	11.53	5.24
24	16.3	82	29.79	11.05	5.10	1.62
25	15	90	39.54	13.07	12.82	7.92
26	11.4	98.8	42.04	6.08	15.33	10.67
27	21	107	36.75	7.79	5.29	4.85
28	16	114	34.48	15.94	9.74	6.32
29	25.9	117.6	27.85	5.68	14.01	2.70
30	24.6	117.6	31.65	3.13	7.05	8.63
31	29.5	120	18.94	7.72	3.44	9.52
32	19.3	155	35.78	7.11	5.03	13.11
33	32.6	170	29.88	4.11	9.89	2.38
34	35.5	192	32.10	6.64	4.19	10.44
35	38	210	28.46	11.32	8.09	5.75
36	48.5	239	24.31	6.04	10.30	10.39
37	47.5	252	37.81	13.66	5.45	1.20
38	70	278	21.28	6.60	9.10	6.22
39	66.6	300	23.76	4.52	2.99	7.32
40	66.6	352.8	35.17	7.72	5.33	11.15
41	50	370	36.90	6.70	5.29	9.65
42	79	400	45.74	19.78	9.06	15.24
43	90	450	38.29	12.28	17.29	7.50
44	78	571.4	24.50	2.69	7.04	5.21
45	100	215	120.66	34.55	20.50	13.59
46	150	324	49.50	15.39	10.76	9.78
47	100	360	44.97	8.06	17.63	11.20
48	100	360	15.85	3.78	6.08	5.98
49	190	420	1.89	6.33	2.08	0.69
50	115.8	480	11.37	7.60	5.77	3.14

51	101	750	19.87	9.74	1.90	7.65
52	161.1	815	4.76	5.10	7.53	1.24
53	284.7	973	38.36	8.12	15.10	8.29
54	227	1181	3.93	5.04	1.82	4.78
55	177.9	1228	3.64	2.66	5.33	1.23
56	282.1	1368	17.21	3.35	8.10	5.14
57	219	2120	29.00	5.03	9.54	1.64
58	423	2300	25.78	11.58	7.98	7.56
59	302	2400	0.46	1.68	5.54	3.69
60	370	3240	25.21	5.35	3.02	1.25

Figure (7) shows the chart of comparison of MRE of hybrid models and intermediate COCOMO model for 60 projects of NASA dataset software projects.

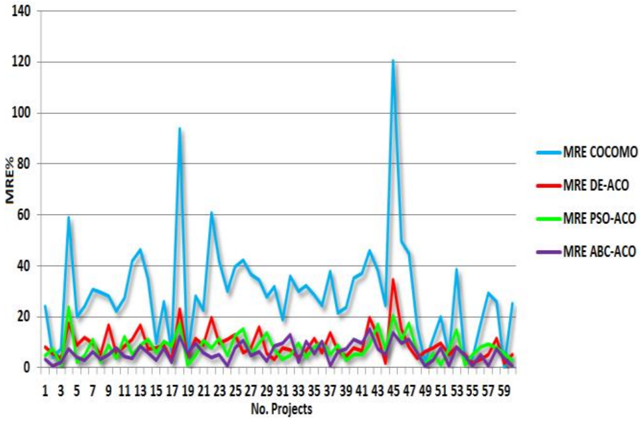

Figure 7. Comparison of MRE in Hybrid Models with COCOMO Model

As it is observed in Figure (7), accuracy of estimation in hybrid models is better than COCOMO model. Table (3) shows comparison of MMRE and PRED hybrid models and intermediate COCOMO models for 60 projects of NASA dataset project software. In Figure (8) comparison of MMRE in hybrid models and intermediate COCOMO models is shown for 60 projects of NASA dataset's software projects.

Table 3. Comparison of MMRE and PRED in Hybrid Models with COCOMO model

Evaluation Criteria	Models			
	COCOMO	DE-ACO	PSO-ACO	ABC-ACO
MMRE	29.64	9.10	8.15	6.11
PRED (10)	16.66	70	73.33	85

As it is shown in Table (3) in all cases, hybrid models have a higher accuracy of PRED in comparison with COCOMO model. As it is observed in Table (3) between the hybrid models, ABC-ACO hybrid model have higher PRED in comparison with other hybrid models. So we can say, hybrid models also have supremacy to each other and their convergence have enormous effect on accuracy and prediction. MMRE error value in COCOMO model is equal to %29.64 and in DE-ACO, PSO-ACO and ABC-ACO hybrid models are %9.10, %8.15 and %6.11 respectively. In Figure (8), MMRE comparison chart of hybrid models and COCOMO is shown for 60 projects dataset of software projects from NASA. As it is observed in Figure (8) hybrid models have the lower MMRE in comparison with COCOMO model. Also, ABC-ACO hybrid model in comparison with other models has the lower MMRE error.

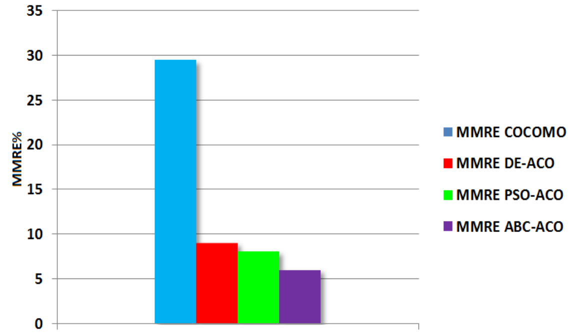

Figure 8. Comparison of MMRE of Hybrid Models with COCOMO model

5. Conclusions and Future Works

In this paper, by using hybrid DE, PSO and ABC algorithms and ACO algorithm to estimate the cost in software projects. Models based on mathematic and algorithms more are used on the needs of software, data and operational characteristics such as performance and reliability in software development projects and software quality other factors do not consider. In the process of software development using algorithm models may fail cost and time feature. Therefore, to estimate and accurate examination of the software project's cost requires models with high estimation accuracy. In algorithmic models, software project's development based on cost and time factors depends on the project EM. Then, determining the value of these multipliers is vital. Therefore, the exact value of the EMs for hybrid models that are based on repeat is very important. To develop software projects, it is required lots of skilled human resources, time and effort during the creation, design, building, testing, evaluation, and maintenance of software projects. Cost, schedule, adding new features of software, heterogeneity and complexity are change software development into an activity that requires SCE. Algorithmic models in terms of estimation have not good accuracy that in this paper for improving the accuracy of the algorithm models from hybrid of meta-heuristic algorithms and ACO are used as a tool to estimation. With presenting this paper, we hope on the future with other meta-heuristic algorithms would improve error in SCE.

References

[1] R. Puri, I. Kaur, Human Opinion Dynamics for Software Cost Estimation, Advances in Intelligent Systems and Computing, Vol. 437, pp. 375-386, 2016.

[2] F.S.Gharehchopogh, A. Talebi, and I. Maleki, Analysis of Use Case Points Models for Software Cost Estimation, International Journal of Academic Research, Part A, Vol. 6, No.3, pp.118-124, 2014.

[3] F.S. Gharehchopogh, I. Maleki, A. Talebi, Using Hybrid Model of Artificial Bee Colony and Genetic Algorithms in Software Cost Estimation, 9th International Conference on Application of Information and Communication Technologies (AICT), IEEE, Rostov on Don, pp. 102-106, 2015.

[4] J. Heidrich, Software Effort Estimation and Risk Management, Lecture Notes in Computer Science, Vol. 7983, pp. 370-371, 2013.

[5] A. Trendowicz, Why Software Effort Estimation? The Fraunhofer IESE Series on Software and Systems Engineering, pp. 3-7, 2012.

[6] E.E. Miandoab, F.S. Gharehchopogh, A Novel Hybrid Algorithm for Software Cost Estimation Based on Cuckoo Optimization and K-Nearest Neighbors Algorithms, Engineering, Technology & Applied Science Research, Vol. 6, No. 3, pp. 1018-1022, 2016.

[7] B.W. Boehm, Software Engineering Economics, Prentice-Hall, Englewood Cliffs, New Jersy, 1981.

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

[9] T. Menzies, D. Port, Zh. Chen, J. Hihn, Validation Methods for Calibrating Software Effort Models, ICSE ACM, USA, pp. 587-595, 2005.

[10] F.S. Gharehchopogh, I. Maleki, S.R. Khaze, A New Optimization Method for Dynamic Travelling Salesman Problem with Hybrid Ant Colony Optimization Algorithm and Particle Swarm Optimization, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 2, No. 2, pp. 352-358, 2013.

[11] S.R. Khaze, I. Maleki, S. Hojjatkhah, A. Bagherinia, Evaluation the Efficiency of Artificial Bee Colony and the Firefly Algorithm in Solving the Continuous Optimization Problem, International Journal on Computational Science & Applications (IJCSA), Vol. 3, No. 4, pp. 23-35, 2013.

[12] F.S. Gharehchopogh, I. Maleki, S.R. Khaze, A New Approach in Dynamic Traveling Salesman Solution: A Hybrid of Ant Colony Optimization and Descending Gradients, International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT), Vol. 3, No. 2, pp. 1-9, 2012.

[13] I. Maleki, S.R. Khaze, F.S. Gharehchopogh, A New Solution for Dynamic Travelling Salesman Problem with Hybrid Ant Colony Optimization Algorithm and Chaos Theory, International Journal of Advanced Research in Computer Science (IJARCS), Vol. 3, No. 7, pp. 39-44, 2012.

[14] F.S. Gharehchopogh, I. Maleki, M. Farahmandian, New Approach for Solving Dynamic Traveling Salesman Problem with Hybrid Genetic Algorithms and Ant Colony Optimization, International Journal of Computer Applications (IJCA), Vol . 53, No. 1, pp. 39-44, 2012.

[15] F.S. Gharehchopogh, I. Maleki, B. Zebardast, 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.), Vol. 61, pp. 16655-16661, 2013.

[16] I. Maleki, A. Ghaffari, M. Masdari, A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization, International Journal of Innovation and Applied Studies, Vol. 5, No. 1, pp. 72-81, 2014.

[17] A. Malik, V. Pandey, A. Kaushik, An Analysis of Fuzzy Approaches for COCOMO

II, International Journal Intelligent Systems and Applications, Vol. 5, pp. 68-75, 2013.

[18] Ricardo de A. Araujo, S. Soares, Adriano L.I.O, Hybrid morphological methodology for software development cost estimation, Expert Systems with Applications, Vol. 39, pp. 6129-6139, 2012.

[19] Z.A. Khalifelu, F.S. Gharehchopogh, A Survey of Data Mining Techniques in Software Cost Estimation, AWER Procedia Information Technology & Computer Science Journal, Vol. 1, pp. 331-342, 2012.

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

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

[22] Ricardo de A. Araujo, Adriano L.I. Oliveira, S. Soares, A Shift-invariant Morphological System for Software Development Cost Estimation, Expert Systems with Applications, Vol. 38, pp. 4162-4168, 2011.

[23] Adriano L.I. Oliveira, Estimation of Software Project Effort with Support Vector Regression, Neuro Computing, Vol. 69, pp. 1749-1753, 2006.

[24] Z. Xua and T.M. Khoshgoftaarb, Identification of Fuzzy Models of Software Cost Estimation, Fuzzy Sets and Systems, Vol. 145, pp. 141-163, 2004.

[25] K.V. Kumar, V. Ravi, M. Carr, N.R. Kiran, Software Development Cost Estimation using Wavelet Neural Networks, The Journal of Systems and Software, Vol. 81, pp. 1853-1867, 2008.

[26] I. Attarzadeh, S.H. Ow, Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks, 2nd International Conference on Computer Engineering and Technology, Vol. 3, pp. 487-491, 2010.

[27] M. Kazemifard, A. Zaeri, N.G Aghaee, M.A. Nematbakhsh, F. Mardukhi, Fuzzy Emotional COCOMO II Software Cost Estimation (FECSCE) using Multi Agent Systems, Applied Soft Computing, Vol. 11 pp. 2260–2270, 2011.

[28] P.V.G.DPrasad Reddy, CH.V.M.K.Hari, M.Jagadeesh, Takagi-Sugeno Fuzzy Logic for Software Cost Estimation Using Fuzzy Operator, International Journal Software Engineering, Vol. 4, No. 1, pp. 53-62, 2011.

[29] Z.A. Khalifelu, F.S. Gharehchopogh, A New Approach in Software Cost Estimation Using Regression Based Classifier, AWER Procedia Information Technology & Computer Science Journal, Vol. 2, pp. 252-256, 2012.

[30] Prasad 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.

[31] M. Dorigo, V. Maniezzo, A. Colorni, Ant System: Optimization by a Colony of Cooperating Agents, IEEE Trans. on Systems, Man and Cybernetics, Part B, Vol.26, No.1, pp.29-41, 1996.

[32] R. Storn, K. Price, Minimizing the Real Functions of the ICEC'96 Contest by Differential Evolution, International Conference on Evolutionary Computation, Nagoya, Japan, pp. 842-844, 1996.

[33] J.Holland, Adaptation in Natural and Artificial Systems, University of Michigan, Michigan, USA, 1975.

[34] J.Kennedy, R.C. Eberhart, Particle Swarm Optimization, In Proceedings of the IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.

[35] D. Karaboga, An Idea Based on Honeybee Swarm for Numerical Optimization, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.