ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 9(21), DOI: 10.17485/ijst/2016/v9i21/95286, June 2016 ISSN (Online) : 0974-5645

An Evolutionary Computation Approach for Project Selection in Analogy based Software Effort Estimation

I. Thamarai* **and S. Murugavalli**

Sathyabama University, Chennai 600 119, Tamil Nadu, India;
ilango.thamarai@gmail.com, murugavalli26@rediffmail.com

Abstract

Objectives: Software effort estimation is a critical task in the software development process due to the intangible nature of software. A new model for software effort estimation using Differential Evolution Algorithm called DEAPS is proposed in this paper. **Method:** In this methodology, the complete set of historical project base is reduced to a set of similar projects using Euclidean distance metric. Then Differential Evolution Algorithm which is an Evolutionary Computation method is used for optimization and the most relevant project is retrieved. The proposed method is validated on Desharnais dataset. **Findings:** DE has a very effective mutation process which improves the ability of exploration. So we got promising results which indicate that the use of this model could significantly improve the efficiency of Analogy based Software Effort Estimation. The metrics used are MMRE, MdMRE and pred (25%). The results are compared with previous findings and the results clearly show that the proposed method is better than the existing methods. **Application:** This methodology can be used to minimize the errors in the software estimation so that financial loss and delay in the completion of project may be avoided.

Keywords: Algorithmic and Non-Algorithmic Models, Differential Evolution Algorithm, Evolutionary Computation, Software Effort Estimation

1. Introduction

Software estimation at early stage of software development is very important to minimise the risk in the process. It is a crucial task to find correlations between the resources that are mostly partial in nature and predict the effort required for the software project. The uncertainty level of effort prediction is high in the early stages of software development and tends to decrease as the process progresses towards the final stages[1]. So advanced techniques are used to obtain more accurate estimation. Software effort estimation remains a complex problem that needs considerable research attention. Several methods have been proposed by many researchers. Expert Judgement method is the most traditional method where estimation is done based on the experience of the experts in the field. Larry Putnam and Barry Boehm are the pioneers of software estimation methods. Boehm proposed a new method called COCOMO (Cost Constructive Model) that uses equations to calculate software development effort[2]. Larry Putnam developed SLIM (Software Life Cycle Management) model[3]. Albrecht introduced the concept of Function Point (FP) as a metric for software effort estimation. Analogy based estimation was proposed[4]. In this method, the parameter of the proposed new project is compared with the completed projects to give the effort or cost estimation. The rapid change in the software development methodologies has led to newer versions of COCOMO called COSEEKMO[5]. MRE errors are largely reduced in this model and the analysis is fully automatic. In[6] proposed a use case based model which works best in an interactive development process. In discussed about Use Point Method for estimating effort for a software project[7]. They also presented a Re-UCP

* Author for correspondence

(Revised Use Case Point Method) and evaluated their model[8]. In [9] presented a new approach that uses different estimation techniques for different states of software. In recent years, soft computing techniques are widely used to predict the software development effort to respond to the dynamic behaviour of software projects. Soft computing is a consortium of fuzzy logic, Artificial Neural Network (ANN) and Evolutionary Computation (EC). Artificial Neural Networks led to the concept of Evolutionary Computation Algorithm.

A lot of researches have been made to use Evolutionary Computational algorithms in Software effort estimation. The EC algorithms solve the problem by using techniques that are inspired by biological evolution strategies. This is a subset of soft computing methods. The main idea is the survival of the fittest which arises from the natural biological evolution where survival is achieved through reproduction. The initial population of individuals are referred as chromosomes and these individuals compete to produce off springs. The chromosomes with the best survival capabilities have the better chance to reproduce. Crossover is the process of generating new off spring. The characteristics of an individual can also be altered to produce a better off spring. This process is called mutation. The survival strength of an individual is measured by a value called Fitness function which depends on the nature of the process or problem. The EC Algorithms are population based search algorithms. The complexity of the problem depends on the size of the population. The crossover and mutation operations are performed iteratively until a stopping criteria is satisfied, which may be to terminate the process when an acceptable solution is found or when there is no improvement in the process of iteration. There are many popular E.C. Algorithms such as Genetic Algorithm, Genetic Programming, Particle Swarm Optimization, Cultural Evolution, Differential Evolution, etc. Following are the few literatures that have applied E.C. techniques on the estimation of software effort. In [10] proposed the use of soft computing method to improve the software effort estimation accuracy. In this approach, fuzzy logic has been used with Particle Swarm Optimization (PSO). NASA dataset is used and the results are compared with COCOMO data set. Sehra et al analyzes different soft computing techniques in the existing models[11]. In [12] proposed to use Genetic Algorithm for the selection of relevant project in Analogy for software effort estimation. Albrecht data set and Desharnais data set are used to implement the method. GA is used to optimize the nearest neighbours and feature weights simultaneously. In [13] proposed a hybrid method which is a combination of fuzzy clustering, ABE and ANN to increase the accuracy of effort estimation. In this method the projects are clustered and the effect of irrelevant and inconsistent project is decreased by designing a specific framework. In [14] integrated ABE with fuzzy logic to improve the estimation at early stages of a software development life cycle. The datasets used are Desharnais, ISBSG, Kemerer, Albrecht and COCOMO. Results were compared with models that used Case Based Reasoning and Stepwise Regression. In [15] presented a model using soft computing technique to improve software effort estimation process. In this method, the parameters of COCOMO-II model are fuzzified to get more reliable effort estimation. A new method is presented based on Radial Basis Function (RBF), Artificial Neural Network and Genetic Algorithm (GA) to estimate software project cost[16]. In [17] used Particle Swarm Optimization (PSO) algorithm hybrid with Fuzzy C-Means (FCM) and Learning Automata (LA) algorithms for Software Cost Estimation (SCE) and validated the result using NASA dataset.

A model was proposed that uses ANN method in NASA dataset and results show smaller Mean Magnitude of Relative Error (MMRE) [18]. In [19] proposed a Fuzzy Analogy software cost estimation based on linguistic quantifiers. The model was designed for COCOMO dataset .The similarity between two software projects are calculated based on fuzzy aggregation operators. But this method did not perform well with other data sets and also it is not suitable for estimation at early stages. An innovative method namely FUZANN (Fuzzy Analogy Neuroticism) is proposed by S.Malathy et al which encompass the Fuzzy logic based analogy methods with neuroticism characters to explore the effort estimation[20]. The model is analysed with Desharnais, NASA 60 and NASA 93 datasets. In [21] used Genetic Programming (GP) in the software effort estimation. Desharnais dataset was used and it is proved that use of GP has increased the accuracy but data interpretation is difficult. A new methodology based on Fuzzy logic and Particle Swarm Optimization (PSO) Algorithm was proposed for the software estimation[22]. The uncertainties in the size of the projects are controlled by Fuzzy Logic Controller using the particles of PSO Algorithm. In recent years the use of Differential Evolution Algorithm is dealt in various fields of research. DE is an Evolutionary Computation Algorithm that is more simple and straight forward than most other EC Algorithms. It has only three control parameters namely Size of Population (NP), Mutation Scale Factor (F) and Crossover Constant (CR). Though it is similar to GA, it differs in the technique to guide the search process. Distance and direction information from the current population is used for searching. This paper is intended to analyze the use of EC algorithms in prediction of software estimation and propose a new method called DEAPS (Differential Evolution in Analogy for Project Selection) which is better than the other comparable methods which are in vogue. The rest of this paper is organised as follows. In the section 2, we propose a new model, DEAPS that uses Differential Evolution Algorithm in Estimation by Analogy method. In section 3, the results of the experiment with the Desharnais dataset have been dealt in detail. Section 4 discusses the conclusion and future work in this approach.

2. Differential Evolution in ABE

The basic idea of Analogy Based Estimation (ABE) is to compare the parameters of the new project with the parameters of the previous projects whose effort is already known. Euclidean distance metric is most commonly used to find the similarity measure between the features of the testing projects and historical projects. The Analogy method increases the accuracy and reliability of estimation. In our proposed method, DE is used to select the relevant project from the available set of projects in ABE. DE has very effective mutation process that ensures diversity in searching as it improves exploration ability[23]. Many variants of DE are available namely Simple DE, Compact DE, Hybrid DE, Population based DE, etc. The main steps of DE are Selection, Initialization, Mutation and Crossover A simple DE Algorithm creates an initial population by a random set of individual. For each generation, three individuals say x_1, x_2 and x_3 are selected. The offspring x_{off} is generated by the process of mutation.

$$x_{off} = x_1 + F\,(x_2 - x_3)$$

F is a scale factor. There are several techniques for mutation of individuals in DE. The parent vector is mixed with the mutated offspring to produce a new vector based on crossover rate. The child vector produced is compared with parent vector based on fitness function and the best is retained in the population. The process continues until the stopping criterion is achieved. The frame work for this model is shown in Figure 1.

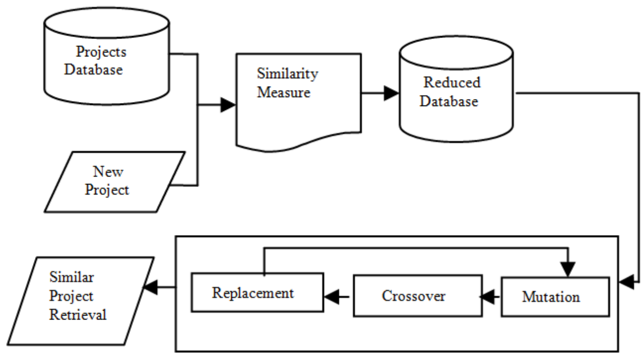

Figure 1. Differential Evolution in the selection of relevant project.

In the first stage, the new project is compared with a database of available historical projects and similar projects are retrieved using the similarity measure. In the second stage, the most relevant project is got after the application of differential evolution algorithm.

3. Experimental Results

3.1 Datasets

To evaluate and compare the performance of the software effort estimation models, data sets are used. The most commonly used Datasets are Desharnais Dataset, Albrecht Dataset and COCOMO 81 dataset. To analyze the performance of our proposed model, we have used Desharnais dataset. The Desharnais dataset consists of 81 projects developed by a Canadian software house in 1989. Each project has twelve attributes which are described in Table 1. Out of the twelve attributes, 11 attributes are independent attributes and one is a dependent attribute. The projects 38, 44, 65 and 75 have missing attributes and so only 77 projects are used.

Table 1. Description of features of Desharnais dataset

Sl. no	Feature	Variable Type	Description
1	Project	Numeric	Project Id
2	TeamExp	Numeric	Team experience in years
3	ManagerExp	Numeric	Manager experience in years
4	YearEnd	Numeric	Year in which the project ended
5	Length	Numeric	Project Duration in months
6	Effort	Numeric	Actual effort measured in person-hours
7	Transactions	Numeric	Number of the logical transactions
8	Entities	Numeric	Number of the entities
9	PointsAdjust	Numeric	Size of the project measured in adjusted function points
10	Envergure	Numeric	Function point complexity adjustment factor
11	PointsNonAjust	Numeric	Size of the project measured in unadjusted function points
12	Language	Categorical(1,2,3)	Type of language in the project expressed as 1, 2 or 3

3.2 Metrics

Metrics are important to validate the performance of Software Effort Estimation models. The most commonly used metrics are given in the Table 2.

Table 2. Metrics for Evaluation

METRICS	DESCRIPTION
$\text{MRE}=\frac{\lvert\text{Acti-Esti}\rvert}{\lvert\text{Acti}\rvert}$	Magnitude of Relative Error
$MMRE=\frac{1}{n}\sum_{i=1}^{n}MREi$	Mean Magnitude of Relative Error
MdMRE = Median (MRE)	Median of Magnitude of Relative Error
$pred(x)=\frac{1}{n}\sum_{i=1}^{n}\begin{cases}1\ if\ MRE\leq x\\ 0\ otherwise\end{cases}$	Percentage of relative error deviation

Act_i is the Actual Effort required for a $project_i$ and Est_i is the Estimated Effort. The model with a lower MMRE is better than the model which has higher MMRE. Also the model with a higher pred(x) value is better than the model with a lower pred(x).

3.3 Results

The results of the proposed model on selected projects of Desharnais dataset are summarized in the Table 3.

Table 3. Result of the Proposed Model

S.No.	Project Id	Estimated Effort	Actual Effort	MRE	Pred(0.25)
1	1	5152	7124.3	0.2768	0
2	4	3829	3154.8	0.2137	1
3	5	2149	1915.4	0.1220	1
4	6	2821	4181.8	0.3254	0
5	7	2569	2977.1	0.1371	1
6	15	4977	3536.1	0.4075	0
7	17	3192	3360.4	0.0501	1
8	19	4494	5799	0.2250	1
9	20	840	850.1	0.0119	1
10	23	5775	5697.1	0.0137	1
11	27	3542	4326.9	0.1814	1
12	28	4277	7035	0.3920	0
13	32	710	484.9	0.4642	0
14	33	2429	3043.5	0.2019	1
15	36	9135	6357.4	0.4369	0
16	39	847	640.9	0.3216	0
17	40	8050	7477	0.0766	1
18	43	2174	2129.7	0.0208	1
19	45	6699	5479.6	0.2225	1
20	47	4004	3028.3	0.3222	0
21	52	3136	1708.1	0.8360	0
22	54	2583	2122.3	0.2171	1
23	56	8232	6743.6	0.2207	1
24	57	3276	2290.9	0.4300	0
25	58	2723	1774.8	0.5343	0
26	61	2926	3101	0.0564	1
27	64	1603	2050.5	0.2182	1
28	68	1267	930.8	0.3612	0
29	70	1155	829.8	0.3919	0
30	71	546	689.8	0.2085	1

Table 4 summarizes the results of various cost estimation methods on Desharnais dataset. The corresponding test results are compared with the previous research results[24].

Table 4. Comparison of result with previous Models

S.No.	Methods	MMRE	PRED(0.25)	MdMRE
1	ABE	0.62	0.22	0.50
2	FWABE	0.46	0.22	0.39
3	PSABE	0.41	0.30	0.38
4	FWPSABE	0.32	0.44	0.29
5	ANN	0.57	0.22	0.43
6	PROPOSED MODEL	0.26	0.57	0.22

The Figure 2 shows a combined illustration of the test results.

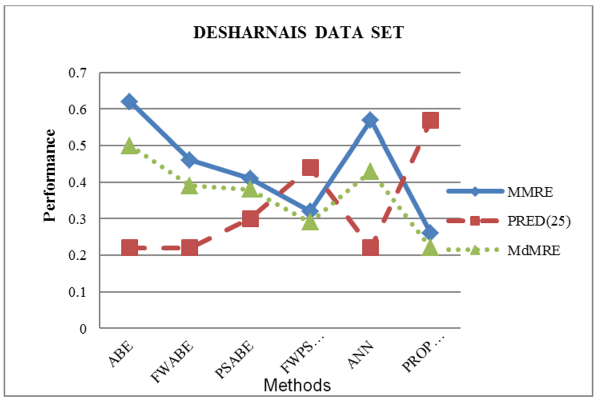

Figure 2. Results of proposed model DEAPS on Desharnais dataset.

The results shows that the application of the proposed Model DEAPS for the selection of relevant project has the best performance among all methods (0.26 for MMRE, 57% for Pred (0.25)and 0.22 for MdMRE). The MMRE and MdMRE are lesser than the other methods and also the probability of a project having MRE<=0.25 is also very high when compared with other models.

4. Conclusion and Future Work

This paper gives a detailed study of how Evolutionary Computation Algorithm has been used in the Software Effort Estimation models. Also a new approach has been proposed to simplify the Analogy based estimation. In our proposed model DEAPS, Differential Evolution Algorithm is used to select the most relevant project from set of historical projects that matches with the new project. The proposed method is implemented in JAVA platform. The experimental results are given and the observation of results clearly indicates that this model is better than existing methods. The metrics used are MMRE, MdMRE and pred (25%). As the search space is big, the Evolutionary Computation method is used which has been proved to be useful. Future work is to analyse the performance of the model with few more real datasets and to prove efficiency of this method.

5. References

1. Kitchenham B, Linkman S. Uncertainty, estimates and risk. IEEE Software; 1997 May –Jun. p. 1–6.
2. Boehm BW, Valerdi R. Achievements and challenges in COCOMO - based software recourse estimation. IEEE Software. 2008 Sep–Oct; 25(5):74–83.
3. Putnam LH. A general empirical solution to the macro software sizing and estimating problems. IEEE Transactions on Software Engineering. 1978 Jul:345–61.
4. Shepperd M, Schofield C. Estimating software project effort using analogies. IEEE Transaction on Software Engineering. 1997 Nov; 23(11):736–43.
5. Menzies T, Chen Z, Hihn J, Lum K. Selecting best practices for Effort Estimation. IEEE Transactions on Software Engineering. 2006 Nov; 32(11):883–95.
6. Ashman R. Project estimation: a simple use case based Model. IT Pro; 2004 Dec. p. 40–4.
7. Kirmani MM, Wahid A. Use case point method of software effort estimation : a review. International Journal of Computer Applications. 2015 Apr; 116(15):1–5.
8. Kirmani MM, Wahid A. Impact of modification made in Re-UCP on software effort estimation. Journal of Software Engineering and Applications. 2015 Jun; 8:276–89.
9. Basri S, Kama N, Ibrahim R. A novel effort estimation approach for requirement changes during software development phase. International Journal of Software Engineering and its Applications. 2015; 9(1):1–16.
10. Ziauddin, Tipu SK, Zman K, Zia S. Software cost estimation using soft computing techniques. Advances in Information Technology and Management. 2012; 2(1):233–38.
11. Sehra SK, Brar YS, Kaur N. Soft computing techniques for software project effort estimation. International Journal of Advanced Computer and Mathematical Sciences. 2011; 2(3):160–67.
12. LI YF, Xie M, Goh TN. A study of genetic algorithm for project selection for analogy based software cost estimation. Proceedings of IEEE IEEM; 2007 Dec. p. 1256–60.
13. Bardsiri VK, Jawawi DNA, Hashim SZM, Khatibi E. Increasing the accuracy of software development effort estimation using projects clustering. IEEE Transactions on IET Software. 2012 Dec; 6(6):461–73.
14. Azzeh M, Neagu D, Cowling PI. Analogy-based software effort estimation using Fuzzy numbers. The Journal of Systems and Software. 2011 Feb; 84(2):2701–84.
15. Kad S, Chopra V. Software development effort estimation using soft computing. International Journal of Machine Learning and Computing. 2012 Oct; 2(5):1–4.
16. Molani M, Ghaffari A, Jafarian A. A new approach to soft-

ware project cost estimation using a hybrid model of radial basis function neural network and genetic algorithm. Indian Journal of Science and Technology. 2014 Jun; 7(6):838–43. DOI: 10.17485/ijst/2014/v7i6/46537.
17. Gharehchopogh FS, Ebrahimi L, Maleki I, Gourabi SJ. A novel PSO based approach with hybrid of fuzzy c-means and learning automata in software cost estimation. Indian Journal of Science and Technology. 2014 Jun; 7(6):795–803. DOI: 10.17485/ijst/2014/v7i6/46481.
18. Singh BK, Misra AK. An alternate soft computing approach for efforts estimation by enhancing constructive cost model in evaluation method. International Journal of Innovation, Management and Technology. 2012 Apr; 3(3):272–75.
19. Idri A, Abran A, Khoshgoftaar T. Fuzzy analogy: a new approach for software effort estimation. International Workshop in Software Measurements. 2001 Aug:1–9.
20. Malathi S, Sridhar S. A novel approach to estimate the software effort based on fuzann technique. European Journal of Scientific Research. 2012; 81(4):563–74.
21. Burgess CJ, Lefley M. Can genetic programming improves software effort estimation? a comparative evaluation. Information and Software Technology. 2001 Dec; 43(14):863–73. DOI: 10.1016/S0950-5849(01)00192-6.
22. Krishna AB, Krishna TKR. Fuzzy and swarm intelligence for software effort estimation. Advances in Information Technology and Management. 2012; 2(1):60–4 .
23. Das S, Suganthan PN. Differential evolution – a survey of art. IEEE Transactions on Evolutionary Computation. 2011 Feb; 15(1):4–31.
24. Li YF, Xie M, Goh TN. A study of project selection and feature weighting for analogy based software cost estimation. The Journal of Systems and Software. 2009 Feb; 82(2):241–52.