International Journal of Innovative Computing, Information and Control
Volume **11**, Number **2**, April **2015**

ICIC International ©2015 ISSN 1349-4198
pp. 587–597

AN EFFICIENT SOFTWARE COST ESTIMATION TECHNIQUE USING FUZZY LOGIC WITH THE AID OF OPTIMIZATION ALGORITHM

Marappagounder Krishnamoorthy Shanker Ganesh[1]
and Keppanagounder Thanushkodi[2]

[1]Research Scholar, Anna University
Chennai-600025, Tamil Nadu, India
shankerganesh.0385@gmail.com

[2]Director, Akshaya College of Engineering and Technology
Coimbatore-642109, Tamilnadu, India

Received December 2013; revised April 2014

Abstract. *Software has, of late, established itself as an indispensable segment in the domain of the entire computer based systems. In a software development task, effective cost estimation surfaces as the most demanding task. Software cost estimation has become a hard nut to crack, these days. Cost estimation process encompasses an assortment of well-arranged steps which yield valuation with incredibly diminished error. Anyhow, the opportunities are galore for cost estimation in view of the ambiguity still afflicting the input data. Thus, we have invested time and effort to launch an innovative approach in accordance with fuzzy logic coupled with the optimization task to estimate effort, which is materialized through this document. In fact, fuzzy logic-based cost estimation models have proved its mettle as the most efficient model for the cost estimation in software engineering. We have also resorted to the inclusion of the optimization algorithm in fuzzy which goes a long way in further boosting the estimation process. The optimization algorithm we have deployed in our novel technique is the mighty Particle Swarm Optimization (PSO) algorithm. The effort of the software is deemed as the fitness for the optimization function. In accordance with these fitness values the optimization of the fuzzy rule is effectively carried. The execution is effected by means of JAVA program and the glistening outcomes recount the success stories of our epoch-making technique as the par-excellence method vis-à-vis its rival approaches.*

1. **Introduction.** In fact, thirty yester-years have enthusiastically witnessed the emergence of several amazing quantitative software cost estimation techniques. Marco vociferously argues that consistent forecast of dimension and effort in software development projects is an essential requirement for devising consistent cost and schedule evaluations [3]. The expenditure and delivery of software projects and the excellence of products are strongly influenced by the precision of software effort evaluation [4]. In the event of under-estimation of costs, the result fallout is erroneous authorization of the envisaged techniques by the less-enlightened managements, eventually leading to surpassing of budget estimated, followed by improperly designed functions and below-par quality, and failing to deliver the goods well within the stipulated time-frame [1].

Software cost development is invariably linked with the duration and the number of essential to successfully finish a software project [5]. Software has become the essence of any hi-tech service or product and hence it is all the more essential that, its excellence is held at its best. There is a flow of flak emanating from the educational circles and the public in general on the below-par software quality right from its inception [14]. Along with the surge in size and significance of software there has been a disturbing growth in its intricacy, making the forecast of software development cost really a hard nut to crack [2,11]. With a keen eye on realizing this target, software development procedure should be effectively streamlined from the prerequisite state to the performance stage [6]. Toeing a different path, the techniques of the chips and bytes era adopt a parasitic behavior on replication, neural network, genetic algorithm, soft computing, fuzzy logic modeling, etc. [8].

In fact, software evaluation precision has assumed alarming proportions as the thorniest issue for software promoters. It is, in essence, a crucial segment which is entrusted with the task of effective designing and tracing of software project in the contemporary worlds. Managing the outlay of software development is essentially significant in the aggressive atmosphere at present. The necessity for consistent and precise software development cost forecasts in software engineering is a daunting task in as much as it deals with significant financial and strategic planning [9]. Of late, the software projects have been multiplying in geometric proportions both in dimension and intricacy, causing heart-burns to the delivery manager as the customer deems the sky not even the limit in quenching his relentless thirst for superlative quality software with cost-effective price [13]. Software effort evaluation invariably directs the exact forecast of the probable quantity of effort, time, and staffing patters necessary to give shape to a novel software system in the childhood days of a mega project [21,25].

Effort evaluation with excellent exactness offers a helping hand in the organization of the overall budgeting and planning. However, it is sad that the precision of these evaluations tends to be incredibly infinitesimal and a Himalayan task to realize, in view of the paucity of data regarding the project when it is in the womb. Precise effort estimations empower software consultancies with the capacity to bid effectively. It may be noted that, while quoting for tenders with a smaller evaluation irrespective of the real results in a devastating damage whereas an exceedingly elevated estimate paves the way for the loss of the bid. Effort evaluations can be engaged as input to project plans, iteration plans, budgets, investment analyses, pricing processes and this makes it extremely obligatory to attain precise valuations [24]. In the course of the development procedure, the cost and time evaluations are fruitful for the primary uneven substantiation and supervision of the conclusion procedure of the project. Moreover, these valuations are found to be beneficial for project productivity appraisal stages. In this regard, software effort evaluations are broadly segregated two vital types such as algorithmic and non-algorithmic [22]. In fact, the constraints of algorithmic versions have effectively paved the way for the investigation of the non algorithmic methods which are soft computing techniques [16].

Skyrocketing requirements for the software in the chip era has elegantly elevated the cyberspace software industry as one of the incredibly coveted cynosure all through the cheering cosmos. It is anybody's guess that a well-knit pack of eager estimation, ever-vigilant eye on procedures and resources, unfathomable investigation, crucial inspection and effectual forecast is a sine-qua-non for the incredible triumph of any growth plan [15]. In this regard, software engineering cost (and schedule) models and evaluation methods like Budgeting, Trade-off and risk analysis, Project planning and control, Software improvement investment analysis are shining stars in the galaxy ever sought after for extensive deployment [12,23].

The residual part of the document takes the following route. Section 2 gallantly takes us to an in-depth discussion on the diverse investigations carried out having a significant bearing on our innovative method. Section 3 charismatically curries favor with us by enlightening us on the deft design approach and the novel technique. Section 4, on the other hand, delights us with the colourful outcomes and the consequential debate of our well-cherished system. Finally, Section 5 performs the customary chore of conclusion of our captivating technique for software cost evaluation toeing the line of soft computing approaches.

2. **Related Researches.**

Several intriguing investigators have invested their sweat and blood with special enthusiasm on software cost evaluation in accordance with the soft computing approaches. Reinvented below are some of the eye-catching systems significantly launched by several investigators.

Attarzadeh and Ow [7] have imaginatively brought to limelight a fuzzy logic pragmatic version to usher in utmost exactness in software effort evaluation. The core intent was the appraisal of the part played by fuzzy logic approach in perking up the effort estimation precision by representing inputs parameters by means of two-pronged Gaussian function which invariably ensured superlative metamorphosis from one period to another. Exploiting the merits of fuzzy logic like fuzzy sets, inputs constraints were effectively indicated by allocation of its feasible values and these fuzzy sets were characterized by membership functions. With a view to achieve a smoother transformation in the membership function for input constraints, its related linguistic values were signified by two-fangled Gaussian Membership Functions (2-D GMF) and rules.

Sehra et al. [10] have scientifically brought to spotlight a soft computing approach in the current version in order to furnish an indepth assessment of software and project evaluation methods doing the rounds in industrial and literary domains in accordance with the various test databases along with their pluses and minuses. The effort concentrated on a software project was perhaps one of the utmost significant and extensively-assessed variables in the procedure of project organization in the past few years. Soft computing was a conglomeration of techniques focused on fuzzy logic, artificial neural networks, and evolutionary computation. It is pertinent to note here, that these techniques showed more signs of harmonization and dynamism rather than aggressiveness.

Evaluating the work-effort and the schedule needed to design and/or sustain a software system was one of the most crucial tasks in the smooth organization of software projects. Software cost evaluation, in essence, was a daunting and burdensome procedure. Evaluation by comparison was one of the easiest approaches in software effort evaluation domain. Ziauddin et al. [17] have zealously conceived a soft computing method to boost the exactness of software effort evaluation. In this technique fuzzy logic was deployed along with particle swarm optimization to evaluate software development effort.

Software development effort evaluation was a challenging issue faced by software promoters because of the dearth of sufficient data on the software intended to be launched during the initial stages of development. The data to be collected for several traits of software have to be subjective; else it may result in inaccuracy and vagueness. Kad and Chopra [18] have systematically launched the employment of soft computing method to form an apt version which steps us the procedure of effort evaluation. To accomplish this end, several constraints of Constructive Cost Model (COCOMO) II are fuzzified thereby resulting in consistent and precise evaluations of effort.

A software effort evaluation technique following soft computing approaches offers a solution to adapt the doubtful and vague traits of software effort drivers. Singh and Misra [19] have brilliantly conceived COCOMO as algorithmic model and an effort was undertaken to authenticate the robustness of artificial neural network technique by means of NASA project data. The vital target of this investigation was to assess the impact of crisp inputs and soft computing approaches on the precision of the yield of the technique when the novel version was performed on the NASA dataset to effectively attain the software effort evaluations.

Software cost evaluation has been aptly deemed as the function of forecasting the effort needed to design a software technique. The fundamental inputs for the software cost evaluation include coding dimensions together with the set of cost drivers. And, the output is deemed as Effort in terms of Person-Months (PM's) method. Kumari and Pushkar [20] have proficiently proposed the evaluation of software project effort by means of support vector regression (SVR).

Evaluation of the development effort in software projects has been giving sleepless nights to the eager experimenter from times immemorial. Vague nature of software projects makes it extremely hard for the estimating function in the nascent phases of the project, thus even a stray thought on attaining accurate appraisals in software projects has been rather kept at bay. Khatibi et al. [26] have competently conceived a fuzzy clustering for placing the identical projects in diverse clusters. Neural Networks (NN) and Analogy Based Estimation (ABE) techniques have been extensively employed in this domain because of the adaptive nature viable with vibrant environs of software projects.

The major contribution in our work is the reduction of error rate when compared with the existing work. We have utilized fuzzy rule based soft-computing technique where the fuzzy rules are further optimized using the PSO algorithm. The usage of optimization algorithm helps in attaining better parameter values and hence the final result that we obtain proves to be more error free and better fitness values are attained. The usage of PSO algorithm along with the fuzzy process proves to be a novel method in the field of software cost estimation and the major advantage of the proposed work lies in the factor that relative error that we obtain after the process is well reduced than the existing works where the fuzzy logic alone is utilized for parameter selection.

3. **Proposed Methodology for Estimating Software Cost.** Cost estimation is capable of offering assistance in the entire and efficient design of related software. In our captivating technique we have effective followed fuzzy rule based software effort estimation. Moreover, we have taken care to duly include the optimization algorithm to the related fuzzy logic for optimizing the fuzzy rules.

3.1. **Cost estimation using fuzzy.** Fuzzy logic has the inimitable prowess of empowering the evaluation task to successfully tackle the abstraction of the data gathered in the primary stages of a software development procedure. It goes a long way in helping to tackle the anxiety on the accurate connotation of semantic values employed in the course of the evaluation function. The roadmap for the cost estimation in our amazing approach is illustrated in the flow chart described in Figure 1.

3.1.1. *Fuzzy logic.* Fuzzy logic is an innovative technique intended to find keys to the vexed dilemmas beyond the domain of quantitative comprehension in view of the inherent intricacies, and it invariably toes the line of fuzzy set theory. It passes through three vital stages such as 1) Fuzzification 2) Inference and 3) Defuzzification.

The fuzzifier modifies the input into linguistic expressions by means of membership functions which indicates the quantity by which a specified numerical value of a definite variable matches with the linguistic expression under consideration. The fuzzy inference engine is entrusted with the task of mapping between the input membership functions and the output membership functions using fuzzy rules which are easily accessible by means of superb awareness of the kinships being worked out. A defuzzifier effectively performs the Defuzzification function to blend the output into a solitary label or numerical value as desired.

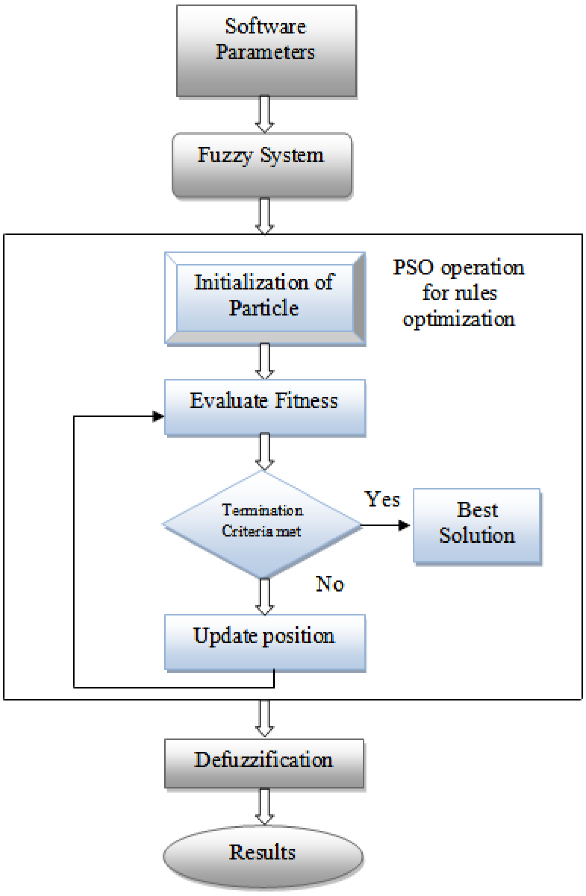

Figure 1. Proposed software cost estimation model

Fuzzy Triangular Membership Function

The attributes with numerical values in the XML database is modified into the fuzzy by means of the triangular membership function. Membership functions are either selected by the client discretely or be devised with the help of machine learning techniques such as artificial neural networks, genetic algorithms and the like. The main shapes of membership functions include triangular, trapezoidal, piecewise-linear and Gaussian, bell-shaped. Now, we have selected the Triangular membership function in which p, q and r characterize the x coordinates of the three vertices of $f(x)$ in a fuzzy set where p represents the lower boundary and r signifies the upper boundary where membership degree is zero, and q stands for the centre where membership degree is 1. One of the main challenges in all fuzzy sets is the manner in which fuzzy membership functions are evaluated.

- The membership function completely shells out the fuzzy set.
- A membership function furnishes a measure of the degree of resemblance of an element to a fuzzy set.
- Membership functions may be of any shape, but there must be certain general instances emerging in authentic applications.

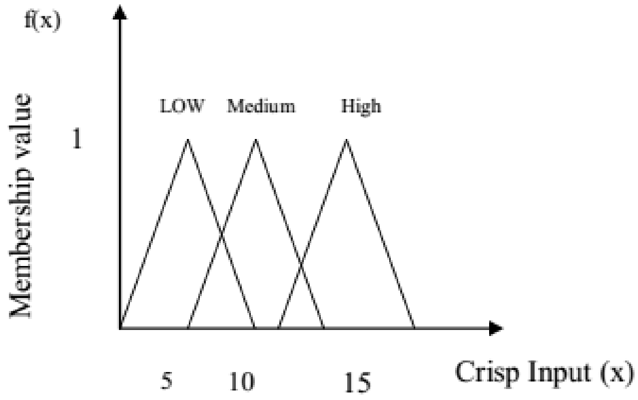

FIGURE 2. Triangular membership function with defined parameters and their values

The formula employed to estimate the membership values is well-explained as follows:

$$f(x) = \begin{cases} 0 & \text{if } x \le p \\ \frac{(x-p)}{(q-p)} & \text{if } p \le x \le q \\ \frac{(r-x)}{(r-q)} & \text{if } q \le x \le r \\ 0 & \text{if } x \ge r \end{cases} \tag{1}$$

Figure 2 effectively exhibits the plot after taking into account all the three membership functions with overlapping values. In Figure 2 the curves for low, medium and high are depicted for the software effort.

By employing the fuzzy membership equation, we have adapted the numerical attributes into the fuzzy.

3.1.2. *Fuzzy rules.* A set of rules have been demarcated for a fuzzy arbiter where it produces the software efforts in accordance with the attributes fuzzy value. The software efforts are clustered into diverse rules based on the parameters and in our technique we have utilized optimization procedure with a view to optimize the fuzzy rules. We have utilized particle swarm optimization for optimizing the rules. The diverse constraints employed for the fuzzy rule generation are *1) TeamExp, 2) Length, 3) Transactions, 4) Entities, 5) Effort.* They are effectively employed for the fuzzy rule creation. The optimization of these rules is performed by means of the PSO.

Fuzzy Rule Optimization Using PSO

PSO owes its origin to the recreation of community character of birds in a cluster. In this technique, every particle flies in the probe space with a velocity influenced by its own flying memory and the flying practice of its companion. Each particle is endowed with its impartial function value which is evaluated by a fitness function. PSO is one of the outstanding evolutionary computing techniques akin the Genetic Algorithm wherein a definite system is activated by a population of discrete solutions.

Steps in particle swarm optimization

The various phases through which the implementation of PSO flows are illustrated as follows.

i. At the outset, activate a population of particles (solutions) together position and velocity chosen discretely for n-variable in the problem space.
ii. For each such particle, evaluate the optimization health qualities in n-variables.

iii. Subsequently assess this fitness value with the particles *pbest* value. If the current fitness value exceeds the *pbest* then choose it as the *pbest* for the additional dispensation.
iv. These fitness values are compared with the entire best prior values and if the current value is better, then refresh the *gbest* for the current particles array index and deem it as the new *gbest*.
v. Revise the velocity and the position of the particle and repeat the steps until attainment of the typical of superior fitness. The velocity and the position of the particle are modified by means of the equations shown hereunder:

$$v_j(n+1) = v_j(n) + a_1 d_1(g_j(n) - h_j(n)) + a_2 d_2(g_j^{'}(n) - h_j(n)) \tag{2}$$

$$h_j(n+1) = h_j(n) + v_j(n+1) \tag{3}$$

vi. The process is run repeatedly till receipt of the key with superior fitness value.

In the equations mentioned above, a_1 and a_2 represent the acceleration constants indispensable for integrating every particle with the *pbest* and *gbest*. The best position of the particle is refreshed as shown in the following equations.

$$g_j(n+1) = \begin{cases} g_j(n), & k(h_j(n+1)) \geq k(g_j(n)) \\ h_j(n+1), & k(h_j(n+1)) < k(g_j(n)) \end{cases} \tag{4}$$

The particle velocity at every dimension is limited to the interval $[\pm V_{\max}]$ and is then estimated and compared with the $V_{\max}$. The $V_{\max}$ represents a crucial parameter. The $V_{\max}$ is entrusted with the task of assessing the resolution with which the region between existing position and the target position are traced. The $V_{\max}$ values furnish a foundation to ascertain whether the particles emerge in the long run as a superior solution or not. By means of the equations furnished above, we are capable of ascertaining the fitness of the solution and select the optimal solution in line with these fitness values.

It furnishes ample evidences to the effect that PSOs are competent tools which can be deployed to ascertain solutions to the optimization hassles like parallel evolutionary

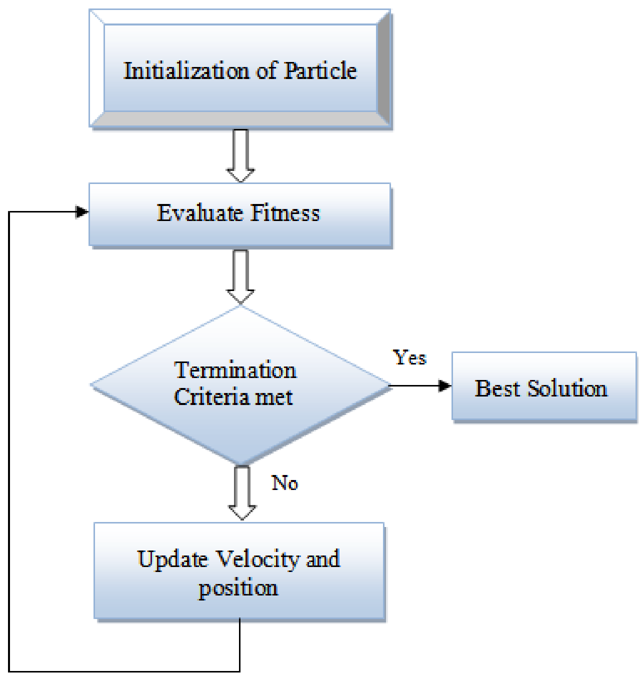

Figure 3. General flow diagram for particle swarm optimization

techniques. It is capable of being exceedingly employed in several spheres like signal processing, robotics, and simulations applications, to name a few. In the case of our well-envisaged technique, we have resorted to the employment of PSO with the intention of selecting the fuzzy rule for appraisal of software outlay. A hodgepodge of trial cases are performed in the innovative technique intended to attain superb yields in respect of software structural design.

3.2. **Defuzzification.** Long before the defuzzification was launched, aggregation procedure is performed in which fuzzy sets characterizing the outputs of each rule are blended together into a solitary fuzzy set. The result of the amalgamation task is a singular fuzzy set for each output variable. The input for the defuzzification function is a fuzzy set (the aggregate output fuzzy set) and the output is a solitary number. In accordance with the assistance provided by fuzziness for the rule appraisal in the course of the transitional steps, the ultimate preferred yield for each variable is invariably a solitary number. Nevertheless, the aggregate of a fuzzy set is home to a domain of output values, and hence it has to be de-fuzzified with an eye on resolving a solitary output value from the set. The defuzzification technique extensively found employed is the weighted average approach. The effort evaluated by means of the novel technique is lower as against those of parallel approaches and thus our ambitious approach is undoubtedly established to be an effective technique in software cost evaluation.

4. **Result and Discussion.** The datasets employed in the investigation is the Desharnais dataset. The original version of the Desharnais dataset is home to 81 projects, of which 4 have been omitted on account of deficiency in values. The dataset embraces as many as 9 independent variables and one dependent variable. We employ Actual Effort in person hours as the 10^{th} variable for the matrix B. The MARE measure for the Desharnais dataset on effort estimation process is evaluated and observed as 0.14634.

The major assumption that we consider in our proposed method is the fitness values for the parameter estimation. Based on the fitness value we have selected the various parameters that are needed for implementing the proposed approach. Also we have estimated the MRE and MARE values which are required to calculate the mean magnitude of relative error (MMRE). The various parameters that are utilized in our proposed method are TeamExp, Length, Transactions, Entities and Effort. These parameters help in providing the required measures for effectively calculating the cost for any particular software.

Many an experimenter has employed diverse error measurements, of which the most accepted error measure is Mean Absolute Relative Error (MARE).

$$MARE = \sum\nolimits_{i=1}^{n} \left(\left|\left(E_i - A_i\right)/A_i\right|\right)/n \tag{5}$$

where, E_i – Estimated effort and A_i – Actual effort.

4.1. **Experimental results.** Table 1 appearing below demonstrates the test outcomes achieved by our ambitious approach. The actual effort together with the evaluated effort is determined in respect of diverse dimensions. The Magnitude of Relative Error (MRE) for each and every entry is ascertained. It is observed that the actual effort has always been considerably lesser as against the estimated effort. The MMRE for diverse efforts are estimated by means of the following Equation (7) appearing in the performance appraisal segment.

TABLE 1. Effort estimates and MRE

No.	Actual effort	Estimated effort	MRE
1	52.0	65.9432	0.2681398531948195
2	124.0	138.3309	0.11557240503182262
3	60.0	64.5942	0.07657145153747227
4	119.0	133.1731	0.11910186417687778
5	94.0	103.9609	0.10596778071185663
6	89.0	97.03521	0.09028332481758743
7	42.0	49.1576	0.17042080408984633
8	52.0	63.0460	0.2124247062834734
9	88.0	99.4889	0.13055616859855187
10	38.0	43.6095	0.14761961722488037

4.2. **Performance analysis.** The MRE and MMRE are effectively evaluated by means of Equation (6) shown below:

$$MRE = |(A_i - E_i)| / A_i \tag{6}$$

where, E_i – Estimated effort and A_i – Actual effort.

The MMRE for the estimated effort is ascertained by means of the following Equation (7). It is found that the MMRE for our innovative approaches fares amazingly superior to identical works employing fuzzy technique.

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i \tag{7}$$

Figure 4 vividly draws a graphical demonstration of the effort value attained by our well-conceived technique. In fact, the actual effort is found to be incredibly superior to the evaluated effort values. The novel technique finds itself subjected to evaluation and contrast with certain rival approaches with an eye on ascertaining its superlative efficiency and commanding edge over its contemporary methods. Table 2 is rich in content with

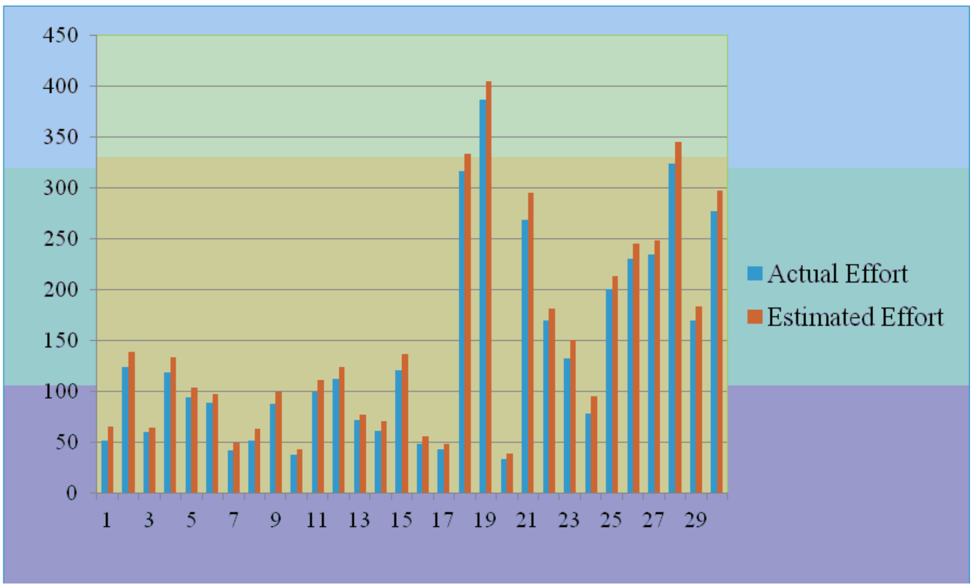

FIGURE 4. Graphical representation for effort estimates

Table 2. MMRE measurements for the proposed and existing methods

METHODS	MMRE (%)
Proposed Method	1.1588
Fuzzy Method	32.651

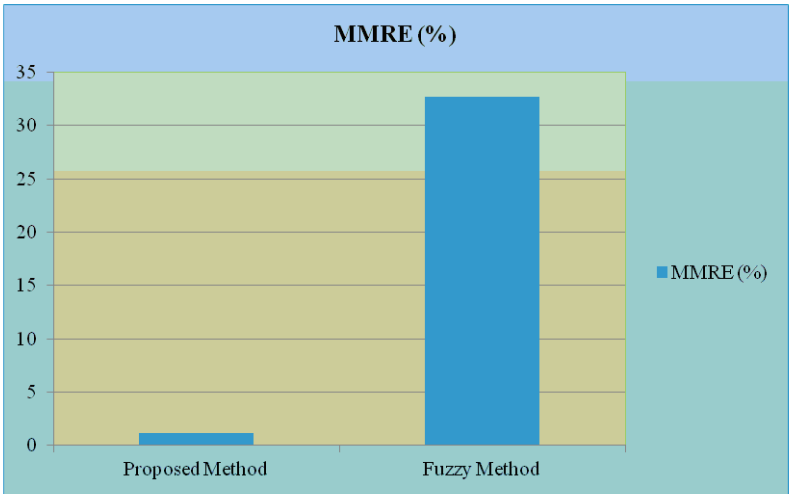

Figure 5. The comparison of MMRE measure for the proposed method

the MMRE measurements for our novel approach together with those of other modern methods. In this regard, the MMRE measure is evaluated in percentage values.

The graphical illustration for evaluation of the epoch-making technique with the modern approaches rooted on MRE and MMRE measures is beautifully pictured in Figure 5 appearing above. Now, the MMRE measure for the dataset of the new-fangled technique is assessed and contrasted with the modern approach [27]. The tables and graphs illustrated are ample credentials highlighting the superb excellence of our masterpiece method vis-à-vis the peer techniques.

5. **Conclusion.** We are very much pleased to launch, through this document, an innovative scheme to evaluate the software project effort, which is mainly based on fuzzy logic and optimization process. In the fuzzy analogy approach, both categorical and numerical data are characterized by fuzzy sets and hence while generating rules we have resorted to the deployment of the optimization by means of PSO with an eye on optimizing the rules for superior functioning. In fact, optimization of the fuzzy rules ushers in superb outcomes for fuzzy thereby paving the way for enhanced estimation process for software.

REFERENCES

[1] P. S. Sandhu, P. Bassi and A. S. Brar, Software effort estimation using soft computing techniques, *World Academy of Science, Engineering and Technology*, vol.46, pp.488-491, 2008.

[2] S. Aljahdali, Development of a software effort estimation model using differential evolution, *Journal of Electronics and Computer Science*, vol.10, no.1, 2008.

[3] J. Kaur, S. Singh, K. S. Kahlon and P. Bassi, Neural network – A novel technique for software effort estimation, *International Journal of Computer Theory and Engineering*, vol.2, no.1, pp.17-19, 2010.

[4] W. L. Du, D. Ho and L. F. Capretz, Improving software effort estimation using neuro-fuzzy model with SEER-SEM, *Global Journal of Computer Science and Technology*, vol.10, no.12, pp.51-63, 2010.

[5] J. S. Pahariyaa, V. Ravia, M. Carra and M. Vasua, Computational intelligence hybrids applied to software cost estimation, *International Journal of Computer Information Systems and Industrial Management Applications*, vol.2, pp.104-112, 2010.

[6] A. B. Nassif, L. F. Capretz and D. Ho, Enhancing use case points estimation method using soft computing techniques, *Journal of Global Research in Computer Science*, vol.1, no.4, pp.12-21, 2010.
[7] I. Attarzadeh and S. H. Ow, A novel algorithmic cost estimation model based on soft computing technique, *Journal of Computer Science*, vol.6, no.2, pp.117-125, 2010.
[8] K. Choudhary, Parametric estimation of software systems, *International Journal of Soft Computing and Engineering*, vol.1, no.2, pp.17-20, 2011.
[9] S. Kad and V. Chopra, Fuzzy logic based framework for software development effort estimation, *International Journal of Engineering Sciences*, vol.1, pp.330-342, 2011.
[10] S. K. Sehra, Y. S. Brar and N. Kaur, Soft computing techniques for software project effort estimation, *International Journal of Advanced Computer and Mathematical Sciences*, vol.2, no.3, pp.160-167, 2011.
[11] S. A. R. Kumar and S. SivaSubramanian, An approach towards developing an efficient software cost estimation system using fuzzy and analogy methods, *International Journal of Engineering Inventions*, vol.1, no.10, pp.29-34, 2012.
[12] D. Kashyap, A. Tripathi and A. K. Misra, Software development effort and cost estimation: Neuro-fuzzy model, *Journal of Computer Engineering*, vol.2, no.4, pp.12-14, 2012.
[13] T. R. Benala, S. Dehuri and R. Mall, Computational intelligence in software cost estimation: A perspective, *CSI Communications*, pp.15-18, 2012.
[14] Y. Alkouni and M. A. Alam, Software cost estimation analysis using Co-Como Ii for the improvement of the quality of software process and cost reduction, *International Journal of Advanced Technology and Engineering Research*, vol.2, no.2, pp.80-85, 2012.
[15] S. A. Abbas, X. Liao, A. Azam and A. K. Khattak, Neural net back propagation and software effort estimation: A comparison based perspective, *ARPN Journal of Systems and Software*, vol.2, no.6, pp.195-207, 2012.
[16] R. R. R. Merugu and V. R. K. Dammu, Effort estimation of software project, *International Journal of Advanced Research in Computer Engineering and Technology*, vol.1, no.10, pp.34-41, 2012.
[17] Ziauddin, S. K. Tipu, K. Zaman and S. Zia, Software cost estimation using soft computing techniques, *Advances in Information Technology and Management*, vol.2, no.1, pp.233-238, 2012.
[18] S. Kad and V. Chopra, Software development effort estimation using soft computing, *International Journal of Machine Learning and Computing*, vol.2, no.5, pp.548-551, 2012.
[19] B. K. Singh and A. K. Misra, An alternate soft computing approach for efforts estimation by enhancing constructive cost model in evaluation method, *International Journal of Innovation, Management and Technology*, vol.3, no.3, pp.272-275, 2012.
[20] S. Kumari and S. Pushkar, Comparison and analysis of different software cost estimation methods, *International Journal of Advanced Computer Science and Applications*, vol.4, no.1, pp.154-157, 2013.
[21] I. R. Khan and M. A. Alam, Software cost estimation using a neuro-fuzzy algorithmic approach, *International Journal of Computer Science and Management Research*, vol.2, no.7, pp.3140-3147, 2013.
[22] R. K. Yadav and S. Niranjan, Software effort estimation using fuzzy logic: A review, *International Journal of Engineering Research and Technology*, vol.2, no.5, pp.1377-1384, 2013.
[23] A. Malik, V. Pandey and A. Kaushik, An analysis of fuzzy approaches for COCOMO II, *International Journal for Intelligent Systems and Applications*, vol.5, pp.68-75, 2013.
[24] K. Ramesh and P. Karunanidhi, Literature survey on algorithmic and non-algorithmic models for software development effort estimation, *International Journal of Engineering and Computer Science*, vol.2, no.3, pp.623-632, 2013.
[25] S. Kumar and V. Chopra, Neural network and fuzzy logic based framework for software development effort estimation, *International Journal of Advanced Research in Computer Science and Software Engineering*, vol.3, no.5, pp.759-763, 2013.
[26] B. V. Khatibi, D. N. A. Jawawi and E. Khatibi, Increasing the accuracy of analogy based software development effort estimation using neural networks, *International Journal of Computer and Communication Engineering*, vol.2, no.1, pp.78-81, 2013.
[27] C. S. Reddy and K. Raju, Improving the accuracy of effort estimation through fuzzy set representation of size, *Journal of Computer Science*, vol.5, no.6, pp.451-455, 2009.