IJCSN International Journal of Computer Science and Network, Volume 4, Issue 4, August 2015
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.417

A Hybrid Firefly-Water Wave Algorithm for Effort Estimation of Software Testing

[1] Poonam Kumari, [2] Gurjot Kaur

[1] Department of Computer Science Engineering, Chandigarh University
Mohali , Punjab, India

[2] Department of Computer Science Engineering, Chandigarh University
Mohali , Punjab, India

Abstract - Test effort estimation of software testing is the complex task. There are multiple factors affect the test effort estimation of software testing. The test effort can be calculated on the basis of effort cost and time required for testing. Multiple studies have been done for developing test effort estimation models but the most of these models provide inaccurate result after some time. . The multiple optimization techniques are used to optimize test effort estimation. The test effort estimation is optimizing multiple model, method and techniques of the test effort estimation. In this paper, we proposed a hybrid algorithm to improve the accuracy. The hybrid algorithm is the combination of firefly and water wave algorithm. The hybrid algorithm is applied on the test effort estimation techniques and the parameters are tuned for optimal performance in terms of minimum error in effort estimation. The results are found to be quite satisfactory both in terms of convergence and accuracy. Thus, it justifies our use of hybrid approach for test effort estimation.

Keywords - *Software Testing, Use Case Point (UCP), Test Point Analysis (TPA), Firefly Algorithm, Water Wave Algorithm.*

1. Introduction

Testing is a necessary measure of software and quality assurance measure of software. Software testing is the complex and important part of software development life cycle process [12]. Software testing is the process of evaluating the correctness of the program during the execution. Software testing can also confer a motive, independent view of the software to permit the business to applaud and understand the risks of software implementation [11]. The test effort refers to the expenses for tests. The test effort is related to the cost and time required for testing. It defines the relation between test cost and failure cost [1]. The test effort estimation are used various techniques such as COCOMO, use case point analysis (UCP), function point, test point analysis (TPA) etc. In this paper, we only implements only use case point (UCP) analysis and test point analysis (TPA). The use case point analysis is based on the use cases. In the software engineering, the use case is a list of steps; it defines the interaction between human (actor) and the system to achieve the goal. The use cases are used at higher level in system engineering than software engineering [2]. It represents the goals of stakeholders and their missions.

- It is the characteristics of the process models and framework.
- It is fit for iteration and evolutionary model and also goal for agile development.

Use case point are used for calculate for software estimation. The effort estimation are calculated on their factors like time, cost and quality of software projects. The use case point methodology is widely used in software industry for estimation [25].

Test point analysis is a top down approach for formulating all methods. Test point analysis is utilized for arriving at the estimating effort. Test point analysis is an estimation of test activities because it only covers the black box testing. The test point analysis can also be helpful in the case where the test hour allowance has been preset. Risks will clearly be identified by compressing the adjective TPA estimate with the assistance of the number of test hour that is predetermined [1]. To manage the complex nature of software, we use various meta-heuristic techniques like soft computing technique and tabu search [1]. The firefly algorithm is a meta-heuristic technique

which is widely used for the optimization of the software. The water wave optimization algorithm is also a nature inspired based algorithm. These algorithms are overcome the limitations of each other. The water wave algorithm is work on the combinatorial optimization and this is the application of firefly algorithm. Hence we merge these two algorithms and make a hybrid algorithm.

2. Related Work

This section presents related work regarding the use case point based effort estimation approach and test point analysis approach. In this section we discuss the previous work of the firefly and water wave algorithm Nazeeh Ghatasheh et al. [15] has discussed the efficiency of effort estimation and applying the Firefly Algorithm on the estimation models. The algorithm is used to optimize the parameters of multiple effort estimation models. The firefly algorithm is discussed in the detail for effort estimation of software. The software estimation models are defined in three variations of the Constructive Cost Model COCOMO. These are the basic COCOMO model and other two are the elongation of the basic COCOMO Model. In this paper the models optimized with the different evaluation criteria and compare the results with other meta-heuristic techniques like PSO, GA. S. Aloka et al. [1] presented a Particle Swarm Optimization (PSO) algorithm.

This paper deals with the PSO that algorithm was applied on the test effort estimation techniques. The two test effort estimation techniques are defined in this paper i.e. use case points (UCP) and test point analysis (TPA). In this the test effort estimation can be optimized for both technique of test effort estimation by applying particle swarm optimization (PSO). The results of existing method compare with the obtained method and results found closer to the actual effort. Lihong Guo et al. [9] presented hybrid meta-heuristic approaches by hybridizing harmony search (HS) and firefly algorithm (FA). In HS/FA, top fireflies scheme was introduced to reduce running time; the other was used to mutate between fireflies when updating fireflies. The new Harmony vector takes the place of the new firefly only if it was better than before, which generally outperforms HS and FA. This paper provides the much better results from the previous results by using firefly and harmony search algorithms Hamed Shah-Hosseini et al. [18] has discusses the Intelligent Water Drops (IWD) algorithm. This algorithm is used to find the optimal values of numerical functions with a mutation-based local search. This paper proposed an algorithm namely IWD-CO (IWD for continuous optimization). This algorithm was modified for the continuous optimization problems. In this paper, the algorithm was tested with six different benchmark functions. The water drop algorithm provides the good experimental results, which permute further researches in this respectfulness. Hamed Shah-Hosseini et al. [20] has proposed a new problem solving algorithm. This algorithm is based on the process of natural river system. This algorithm shows the behavior of the water waves and represents the reaction and action occurs during the water drops in the rivers. So the algorithm is called water drop or intelligent water drop algorithm. This algorithm was firstly proposed for solving the TSP (travelling salesman problem). This paper provides the fast coverage to optimum the solution and provides best solutions.

3. Firefly Algorithm and Water Wave Algorithm

The firefly light flashing behavior is an astonishing signal in the sky, usually found in tropical and temperature regions. There are about 2000 species of firefly algorithm and most fire beetle acquire unique and rhythmic scour. These flashes are used to fundamental function such as attract mating parts as well as potential prays. In additions, the flashing behavior may also save as a vindicatory admonition mechanism. These rhythmic flashes are different from each other on the basis of rate of flashing. Female firefly responds to peerless pattern of flashing of a male firefly which brings both sexes together. We know that, when a light source emits light intensity at a Euclidian distance r from the light source obeys the inverse square low. The intensity I is decreases with increased the Euclidian distance r is term of

$$I \infty {}^{1}/_{r^2} \ .$$

1) Fireflies are attracted toward each other's regardless of gender.
2) The attractiveness of the fireflies is correlative with the brightness of the fireflies, thus the less attractive firefly will move forward to the more attractive firefly.
3) The brightness of fireflies is depend on the objective function.

For two fireflies x_i and x_j ,they can be updated as follows:

$$x_j = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha\epsilon$$

Where α is represents the movement of firefly.

β_0 is represents the attractiveness of the firefly.

γ is represents the intensity of firefly in our work the intensity of firefly represents the cost function of software. and r represents the Euclidian distance of the firefly.

The water wave algorithm is also called intelligent water drop and water drop algorithm. The water wave optimization is based on swarm nature inspired optimization algorithm. This algorithm contains a few necessary elements of natural water drops and action and reaction that occurs between rivers bed & the water drops that flow within. It flows in two categories like meta-heuristic and swarm intelligence. Intrinsically, the IWD algorithm can be used for combinatorial optimization. It was firstly introduced for the travelling salesman problem in 2007. Since then, multitudes of researchers have focused on improving the algorithm for different problems.

The IWD algorithms update the soil for the edges by

$$soil\ (k) = 1.1.soil\ (k) - 0.01.\Delta soil(k)$$
$$soil^{IWD} = soil^{IWD} + \Delta soil(k)$$

Where $\Delta soil(k)$ is calculated by

$$\Delta soil(k) = 0.001$$

```
Step 1: Objective function of f(x), where x=(x1,.........,xd)
Step 2: Generate initial population of fireflies;
Step 3: Formulate light intensity I;
Step 4: Define absorption coefficient γ;
Step 5:  While (t<MaxGeneration)
                For i = 1 to n (all n fireflies);
                   For j=1 to n (all n fireflies)
                      If (Ij > Ii),
                         move firefly i towards j;
                     end if
                     Evaluate new solutions and update light intensity;
                  End for j;
                End for i;
          Rank the fireflies and find the current best;
Step 6:  End while;
      Post process results and visualization;
Step 7: End procedure;
```

Algorithm 1: Firefly Algorithm

Step 1: Static parameter initialization

- Problem representation in the form of a graph
- Setting values for static parameters

Step 2: Dynamic parameter initialization: soil and velocity of IWDs

Step 3: Distribution of IWDs on the problem's graph

Step 4: Solution construction by IWDs along with soil and velocity updating

- Local soil updating on the graph
- Soil and velocity updating on the IWDs

Step 5: Local search over each IWD's solution (optional)

Step 6: Global soil updating

Step 7: Total-best solution updating

Step 8: Go to step 2 unless termination condition is satisfied

Algorithm 2: Water Wave Optimization

4. Proposed Algorithm

Based on the introduction of firefly and water wave algorithm in the previous section, the combination of the two approaches is described and hybrid firefly-water wave algorithm is proposed, which updates the poor solutions to accelerate its convergence speed. The hybrid firefly-water wave algorithm is improving the accuracy of the effort estimation of software testing and reduces the magnitude relative errors. In this algorithm the firefly algorithm are tuned the parameters of test effort estimation techniques i.e. UCP (use case Point) and TPA (test point analysis). But there is an issue with the firefly algorithm. The problem is that the firefly algorithm has itself some tuning parameters which need to be optimized. To solve this problem, another algorithm known as Water Wave Optimization is being utilized for tuning the parameters of the firefly algorithm.

Begin
Initialize parameters of fireflies: alpha, Beta and Gamma

Calculate river velocity

If (number of paths>1)

Select the minimum soil path

Run loop of Water wave algorithm for all alpha, Beta, Gamma

Initialize parameters of TPA and UCP

J:position x of fireflies
MaxGen: the maximal number of generations
γ: the light absorption coefficient
r: the particular distance from the light source

d: the domain space

f(x):objective function as a combination of throughput, efficiency and average waiting time
Define the objective function of f(x)=Calculated Effort, where $x=(x_1,.....,x_d)$
Generate the initial population of fireflies or x_i (i=1,2 ,..., n)
Determine the light intensity of I_i at x_i via $f(x_i)$
While (t<MaxGen)
    For i = 1 to n (all n fireflies);
        For j=1 to n (n fireflies)
            If ($I_j > I_i$),
                move firefly i towards j by
                end if
    Attractiveness varies with distance r via Exp [$-\gamma r^2$];
    Evaluate new solutions and update light intensity;
        End for j;
    End for i;
Rank the fireflies and find the current best;
End while;
Calculate the best parameters and store.

Update river velocities and select optimal path

Go to water wave loop again

Calculate and store the estimated effort

End Procedure

Algorithm 3: Hybrid Firefly-Water Wave Algorithm

5. Work Done

All experiments are performed in the MATLAB framework. This framework is used for the meta-heuristic algorithms implementation. In this experiment, main aim is to make a hybrid algorithm to improve the test effort estimation techniques like use case point (UCP) and test point analysis (TPA). The firefly algorithm is implemented for tune the values of different parameters. The tuning of the firefly algorithm parameters with the water wave optimization and make a hybrid algorithm for both technique of test effort estimation.

In the use case point analysis, we have taken 10 particles, with 16 parameter values corresponding to actors (3 for actor weights), use cases (4 for use cases), and technical factors (9 for technical factors) in the search space randomly. The range of values for the parameters was bound in the intervals shown in Table 5.1.

Table 5.1. Range Taken For Weights of UCP Parameters

Parameters	Range
Actor	15-40
Use Case	1-35
Technical Factor	1-45

In the test point analysis, we have taken 11 particles or module, with 16 parameter values corresponding to function dependent factor (5 factors), quality factor (4 factor), and technical factors (7 for technical factors) in the search space randomly. The range of values for the parameters was bound in the intervals shown in Table 5.2.

Table 5.2. Range Taken For Weights of TPA Parameters

Parameters	Range
Dependent Function	1-20
Quality Factor	1-10
Technical Factor	10-40

In this problem 50 iterations are considered because results converge to an optimal solution within this limit. More iteration may be required, depending on the problem. The actual effort is 390 man days for use case point analysis and 55 man days for test point analysis.

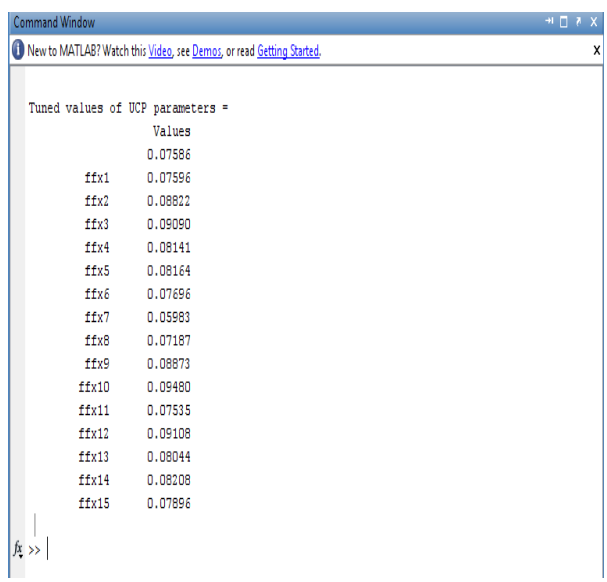

Fig.1 Tuned Values of UCP Parameters with Firefly

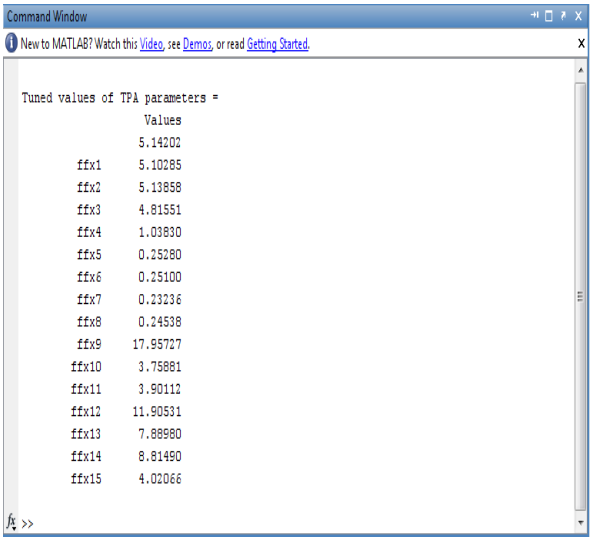

Fig.2 Tuned Values of TPA Parameters with Firefly

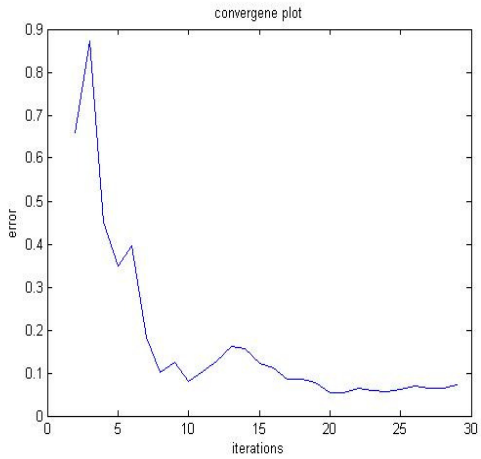

Fig.3 Convergence Graph of UCP with Firefly

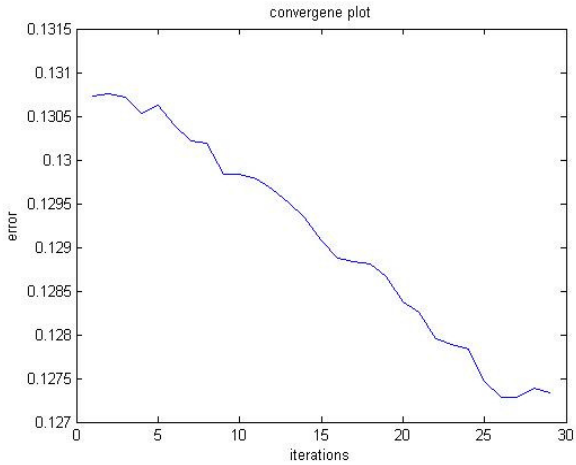

Fig.4 Convergence Graph of TPA with Firefly

Fig.5 Tuned Parameters of FA for UCP with Hybrid Algorithm

Fig.6 Tuned Parameters of FA for TPA with Hybrid Algorithm

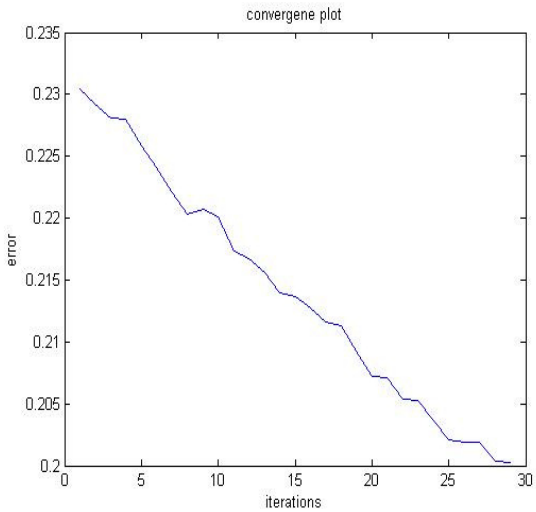

Fig.7 Convergence Graph of Hybrid Algorithm for UCP

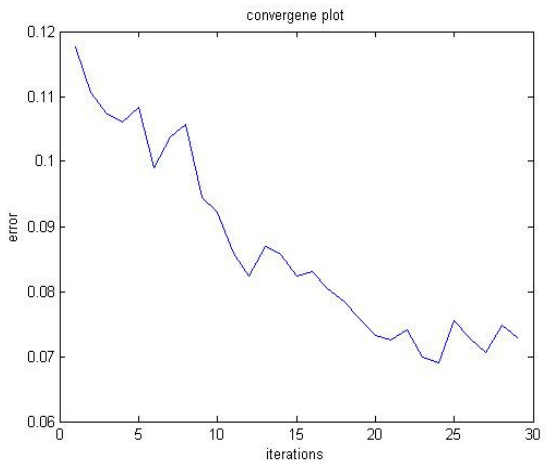

Fig.8 Convergence Graph of Hybrid Algorithm for TPA

Table of Comparison:

Comparison of Firefly algorithm and Hybrid firefly-water wave algorithm

Table 5.3 Comparative Results

Technique	**Firefly(MRE)**	**Hybrid F-WW(MRE)**
Use Case Point	1.3830	0.6110
Test Point Analysis	0.9986	0.4004

6. Conclusion and Future Scope

Test effort estimation which has become of paramount importance considering the amount of money and resources spent on the software testing field by any company. Test Effort estimation was calculated using two methodologies namely- Use Case Point Analysis and Test Point Analysis. This paper proposes a novel hybrid algorithm comprising of firefly algorithm and water wave algorithm to tune the parameters of effort estimation techniques. The firefly algorithm is utilized for tuning the parameters of use case point analysis and test point analysis separately. But there is an issue with the firefly algorithm. The problem is that the firefly algorithm has itself some tuning parameters which need to be optimized. The performance of the algorithm varies on the values of these parameters. These parameters determine the convergence and accuracy. To solve this problem, another novel algorithm known as Water Wave Optimization is being utilized for tuning the parameters of the firefly algorithm. . The results are found to be quite satisfactory both in terms of convergence and accuracy. In future, various other algorithms can be tried and tested for the same. Use of a hybrid of estimation algorithm i.e. Use case point analysis and test point analysis can also be attempted.

References

[1] S. Aloka, Peenu Singh, Geetanjali Rakshit, and Praveen Ranjan Srivastava. “Test Effort Estimation-Particle Swarm Optimization Based Approach.” Springer-Verlag Berlin Heidelberg, CCIS 168, pp. 463–474. 2011.

[2] Suresh Nageswaran. “Test Effort Estimation Using Use Case Points.” San Francisco, California, USA. June 2001.

[3] Kusumoto, Shinji, Fumikazu Matukawa, Katsuro Inoue, Shigeo Hanabusa, and Yuusuke MAEGAWA. "Effort estimation tool based on use case points method." In Proc. of Software Metrics, 10th International Symposium, pp. 292-299. 2004.

[4] James Kennedy and Russell Eberhart. “Particle Swarm Optimization.” IEEE Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995).

[5] Vahid Khatibi Bardsiri, Dayang Norhayati Abang Jawawi ,Siti Zaiton Mohd Hashim and Elham Khatibi. “A PSO-based model to increase the accuracy of software development effort estimation.” Springer Science+Business Media, Software Qual J, 21:501–526.2013.

[6] Lin, Jin-Cherng, Yueh-Ting Lin, Han-Yuan Tzeng, and Yan-Chin Wang. "Using Computing Intelligence Techniques to Estimate Software Effort." International Journal of Software Engineering & Applications (IJSEA) 4, no. 1 (2013): 43-53.

[7] Narmada Nayak and Durga Prasad Mohapatra. “Automatic Test Data Generation for Data Flow Testing Using Particle Swarm Optimization.” Springer-Verlag Berlin Heidelberg, IC3 2010, Part II, CCIS 95, pp. 1–12.2010.

[8] Aiguo Li, Yanli Zhang. “Automatic Generating All-Path Test Data of a Program Based on PSO.” World Congress on Software Engineering, 978-0-7695-3570-8/09, IEEE 2009.

[9] Lihong Guo, Gai-Ge Wang, Heqi Wang, and Dinan Wang. “An Effective Hybrid Firefly Algorithm with Harmony Search for Global Numerical Optimization.” Hindawi Publishing Corporation,The ScientificWorld Journal , 2013.

[10] Yang, Xin-She. "Firefly algorithms for multimodal optimization." In Stochastic algorithms: foundations and applications, pp. 169-178. Springer Berlin Heidelberg, 2009.

[11] Arora, Sankalap, Sarbjeet Singh, Satvir Singh, and Bhanu Sharma. "Mutated firefly algorithm." In Parallel, Distributed and Grid Computing (PDGC),

2014 International Conference on, pp. 33-38. IEEE, 2014.

[12] Beizer, Boris. Software testing techniques. Dreamtech Press, 2002.

[13] Wang, Bin, Dong-Xu Li, Jian-Ping Jiang, and Yi-Huan Liao. "A modified firefly algorithm based on light intensity difference." Journal of Combinatorial Optimization (2014): 1-16.

[14] Abdullah, Afnizanfaizal, Safaai Deris, Mohd Saberi Mohamad, and Siti Zaiton Mohd Hashim. "A new hybrid firefly algorithm for complex and nonlinear problem." In Distributed Computing and Artificial Intelligence, pp. 673-680. Springer Berlin Heidelberg, 2012.

[15] Ghatasheh, Nazeeh, Hossam Faris, Ibrahim Aljarah, and Rizik MH Al-Sayyed. "Optimizing Software Effort Estimation Models Using Firefly Algorithm." Journal of Software Engineering and Applications 8, no. 3 (2015): 133.

[16] Hashmi, Adil, Nishant Goel, Shruti Goel, and Divya Gupta. "Firefly algorithm for unconstrained optimization." IOSR Journal of Computer Engineering 11, no. 1 (2013): 75-78.

[17] Arora, Sankalap, and Satvir Singh. "The firefly optimization algorithm: convergence analysis and parameter selection." International Journal of Computer Applications 69, no. 3 (2013): 48-52.

[18] Shah-Hosseini, Hamed. "An approach to continuous optimization by the intelligent water drops algorithm." Procedia-Social and Behavioral Sciences 32 (2012): 224-229.

[19] Xing, Bo, and Wen-Jing Gao. "Intelligent Water Drops Algorithm." In Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms, pp. 365-373. Springer International Publishing, 2014.

[20] Shah-Hosseini, Hamed. "Problem solving by intelligent water drops." In Evolutionary Computation. IEEE Congress on, pp. 3226-3231. IEEE, 2007.

[21] Zheng, Yu-Jun. "Water wave optimization: A new nature-inspired metaheuristic." Computers & Operations Research 55 (2015): 1-11.

[22] URL:http://istqbexamcertification.com/

[23] Kushwaha, Dharmender Singh, and Arun Kumar Misra. "Software test effort estimation." ACM SIGSOFT Software Engineering Notes 33, no. 3 (2008): 6.

[24] Aranha, Eduardo, and Paulo Borba. "An estimation model for test execution effort." In Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on, pp. 107-116. IEEE, 2007.

[25] Zhu, Xiaochun, Bo Zhou, Li Hou, Junbo Chen, and Lu Chen. "An experience-based approach for test execution effort estimation." In Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for, pp. 1193-1198. IEEE, 2008.

[26] Gharehchopogh, Farhad Soleimanian, Isa Maleki, and Seyyed Reza Khaze. "A Novel Particle Swarm Optimization Approach for Software Effort Estimation." International Journal of Academic Research, Part A 6, no. 2 (2014): 69-76.

[27] Subutic, M., Milan Tuba, and Nadezda Stanarevic. "Parallelization of the firefly algorithm for unconstrained optimization problems." Latest Advances in Information Science and Applications (2012): 264-269.

[28] Rayapudi, S. Rao. "An intelligent water drop algorithm for solving economic load dispatch problem." International Journal of Electrical and Electronics Engineering 5, no. 2 (2011): 43-49.

[29] Binitha, S., and S. Siva Sathya. "A survey of bio inspired optimization algorithms." International Journal of Soft Computing and Engineering 2, no. 2 (2012): 137-151.

[30] Thilagavathi, D., and Antony Selvadoss Thanamani. "Intelligent Water Drop Algorithm Based Particle Swarm Optimization (IWDPSO) Towards Multi Objective Job Scheduling for Grid Computing." (2015).

[31] Brajevic, Ivona, Milan Tuba, and Nebojsa Bacanin. "Firefly Algorithm with a Feasibility-Based Rule for Constrained Optimization." In Proceedings of the 6th WSEAS European Computing Conference (ECC'12), ISBN, pp. 978-1.

[32] Bacanin, Nebojsa, Ivona Brajevic, and M. Tuba. "Firefly algorithm applied to Integer Programming Problems." (2013).

[33] Kumari Poonam, Nikita Bakshi, and Yamini Pathania. "Test Effort Estimation and Its Techniques." International Journal For Technological Research In Engineering, 2015.

[34] S. X. Yang, "Firefly Algorithm", Engineering Optimization. Hoboken, New Jersey: Wiley, pp. 221-230, 2010.

[35] Alijla, Basem O., Li-Pei Wong, Chee Peng Lim, Ahamad Tajudin Khader, and Mohammed Azmi Al-Betar. "A modified Intelligent Water Drops algorithm and its application to optimization problems." Expert Systems with Applications 41, no. 15 (2014): 6555-6569.