Tuning of Use Case Point (UCP) Analysis Parameter using PSO

Poonam Kumari
Post Graduate Student
Dept. of Computer Science Engineering
Chandigarh University

Ishdeep Singla
Assistant Professor
Dept. of Computer Science Engineering
Chandigarh University

ABSTRACT

Test Effort Estimation is an important activity in software development. The test effort can be calculated on the basis of effort cost and time required for testing. Several studies have been done for developing test effort estimation models but to some extent only, most of these models result in erroneous results. So there is a strong need to optimize the efforts estimated. Meta heuristic techniques can be used for this purpose, to optimize a problem by iteratively trying to improve a solution, using some computational methods. In this paper, we have implemented meta-heuristic based search algorithm namely PSO. The particle swarm optimization algorithm is used for improving testing effort estimation. The particle swarm optimization algorithm (PSO) is applied on use case point (UCP) and results led us to the conclusion that test effort estimation can be optimized by applying PSO. The PSO optimization can also be applied for estimating efforts of software development. This implementation increases the accuracy of testing effort estimation.

Keywords

Software testing, particle swarm optimization (PSO) and use case point (UCP).

1. INTRODUCTION

Testing is the most important quality assurance measure of software. Software testing is an important and complex issue of software development process. Software testing is defined in the different activities of the test process. Testing is a process rather than activity. It involved the series of the activities. The defects find out without executing the code is static testing and the code is executed for manifest or presents the results. These testing are useful for cost effective way of testing. The test effort refers to expenses for test in the software development process. It is related to the effort cost and time required for testing. Test effort defines the relation with test effort and failure effort. To analyses the factor in test effort are difficult, because the factor act upon each other. The test effort estimation are used use case point methodology for estimation. To manage the complex nature of software, we use various meta-heuristic techniques like soft computing technique and tabu search [1]. The particle swarm optimization is an meta-heuristic technique which is widely used for the optimization of the software and other various field.

The PSO algorithm is based on social behavior of bird flocking, fish schooling and swarming theory in particular [3]. This technique is also related to the evolutionary computation techniques like genetic algorithm (GA) and evolutionary programming. This technique is used for optimization. It provides the position of each particle randomly and it aware of the global best and local best positions. The mathematical equation of PSO is based on the no. of particles and velocity of particles. Each particle in PSO flies through the search space with an adaptable velocity that is dynamically modified according to its own flying experience and also flying experience of other particles using the following equations.

$$v_{id}^{t+1} = w \times v_{id}^{t} + r_1 \times c_1 \times \left(p_d^{g} - x_{id}^{t}\right) + r_2 \times c_2 \times \left(p_d^{l} - x_{id}^{t}\right) \dots\dots\dots\dots\dots (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \dots\dots..(2)$$

where

- v_{id}^{t+1} is a velocity vector at t+1 time for i particle in d^{th} dimension
- x_{id}^{t} position vector at t+1 time for i^{th}particle in d dimension
- r_1, r_2 are random number generators.
- C_1 and $C_2$are are learning rates governing the cognition and social components.
- p_{id}^{l} represents the particle with best p-fitness.
- W is the inertia factors that dynamically adjust the velocities of particles gradually focusing the PSO into a local search.

Test Effort Estimation is an important activity in software development because on the basis of effort cost and time required for testing can be calculated [5]. Test effort estimation uses following techniques:

- Conversion of software size into person hours of effort directly using a conversion factor.
- Conversion of software size into testing project size such as Test Points or Software Test Units using a conversion factor and then convert testing project size into effort
- We can also derive software testing project size and effort using Delphi Technique or Analogy Based Estimation technique.

2. RELATED WORK

James Kennedy et al,[3]introduces a method for optimization of continuous nonlinear functions. The method was discovered through simulation of a simplified social model; thus the social metaphor is discussed, though the algorithm stands without metaphorical support. This paper describes the particle swarm optimization concept in terms of its precursors, briefly reviewing the stages of its development from social simulation to optimizer. Discussed next are a few paradigms that implement the concept. S. Aloka et al. has presents a Particle Swarm Optimization (PSO) algorithm. It was applied on two techniques: use case points (UCP) and test point analysis (TPA) and the results led us to the conclusion that test effort estimation can be optimized by applying PSO [1]. The results were compared with those obtained from existing methods, and were found to be closer to the actual effort. Suresh Nageswaran et al. [2**]** developed a new approach to the estimation of software testing efforts based on Use Case Points [UCP] as a fundamental project estimation measure. The V model must be in use and use of case generation must start becoming available right at the requirements gathering phase. The author conjectures that this could become a more robust method of estimation over a period of time. The availability of data from past projects will definitely contribute to the accuracy of these estimates. The estimation technique is not claimed to be rigorous, but the approach offers significant practical advantages over ad hoc techniques currently in use.

3. PROPOSED WORK

Software testing forms an integral part of software development process and is said to consume maximum effort of the whole process, which is nearly 50% of the total effort. Test Effort Estimation is an important activity in software development because estimating the effort beforehand enables project managers to allocate resources i.e. budget, time and staff efficiently and avoid future inconvenience. Many effort estimation techniques are available resulting into satisfactory estimates. However, with increasingly tight schedules and market competition, more accurate estimates are needed. Use Case Point analysis one such technique which has proven to be successful over time in estimating testing effort based on the number of use cases. However, there are a number of parameters which are utilized in the analysis but their values have to be set by the tester on their previous experience. This increases the dependency of the estimation accuracy on human and can vary widely as per the tester's capabilities. This encourages the use of optimization techniques to be utilized to tune the parameters of the UCP analysis. In this project, a heuristic technique called Particle Swarm Optimization (PSO) has been utilized to do the same and its performance has been tested over a well-known data and shown to perform exceedingly well in terms of both accuracy and convergence rate.

4. METHODOLOGY

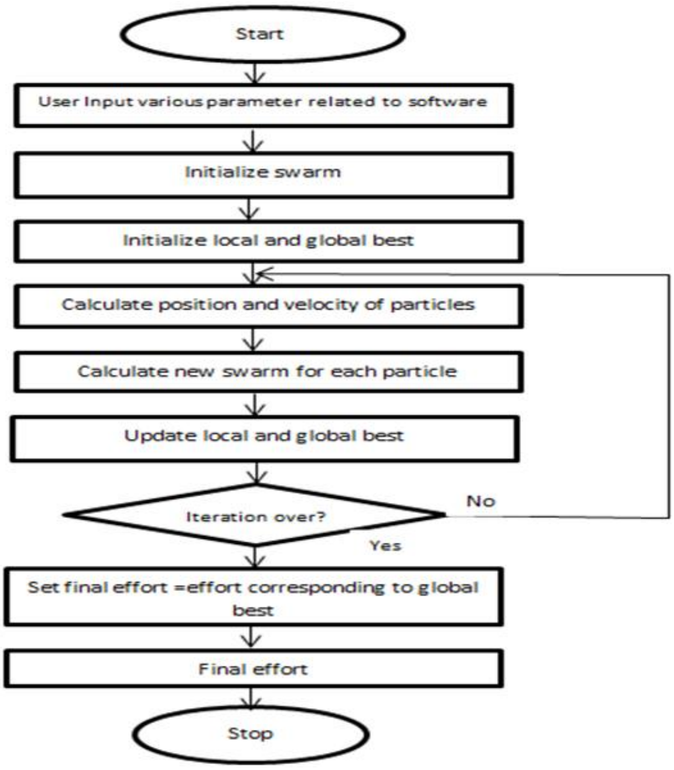

Start
User Input various parameter related to software
Initialize swarm
Initialize local and global best
Calculate position and velocity of particles
Calculate new swarm for each particle
Update local and global best
Iteration over?
No
Yes
Set final effort =effort corresponding to global best
Final effort
Stop

Fig 1: Flowchart of Methodology

5. WORK DONE

All experiments are performed in the MATLAB framework. This framework is used for the PSO algorithm implementation. In this experiment, main aim is to tune up the parameters of PSO algorithm to improve the test effort estimation. 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 Use Case Point Analysis in the search space randomly. The range of values for the parameters was bound in the intervals shown in Table 4.1.

Table 4.1. Range Taken For Weights of Parameters

Parameters	Range
Actor	1-3

Use Case	5-20
Technical Factor	1-5

We have experimented on the 10 particles, with 16 parameters of the UCP analysis. Initially, effort of all 10 particles is calculated by taking the effort equation as the objective function. This will be their initial local bests, and from these local bests, the best one is taken to be the initial global best. The 16 parameters are actor, use cases and technical factor. In each iteration, these parameters for each particle are updated using the position and velocity equations for PSO. And then, objective function is again computed to find the local best and global best. Thus in each iterations, local and global best get updated, and the swarm moves towards optimality. In this problem 100 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**.**

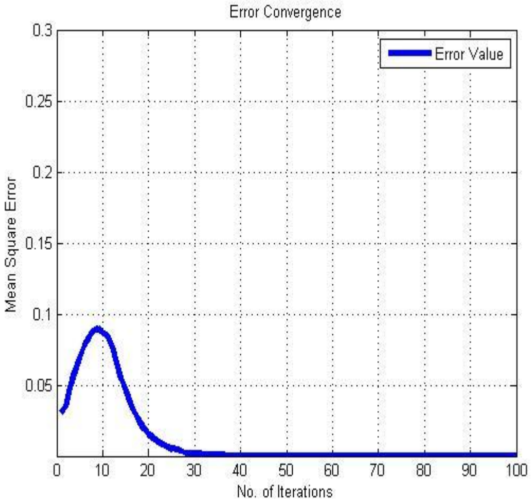


Fig.2 Shows Error Convergence of particles

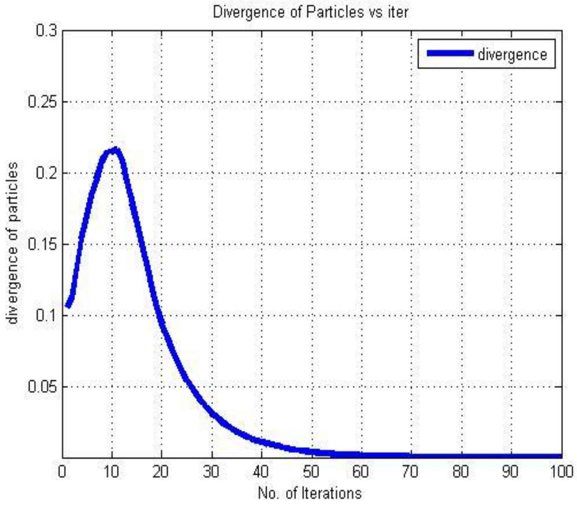


Fig.3 Shows divergence of particles

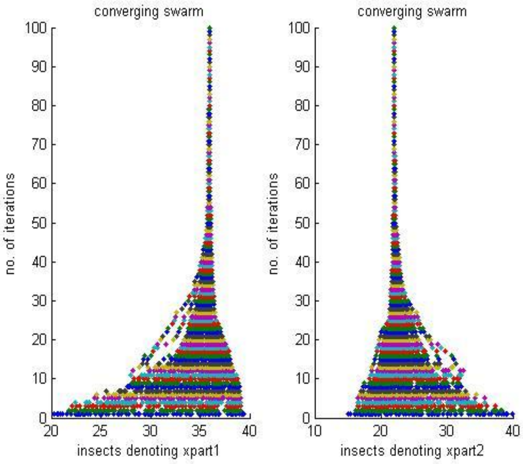
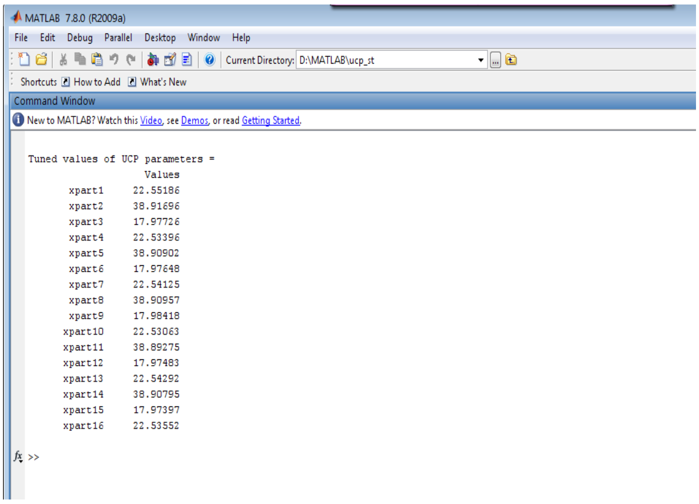


Fig. 4 Converging Swarm


MATLAB 7.8.0 (R2009a)
File
Edit
Debug
Parallel
Desktop
Window
Help
Current Directory: D:\MATLAB\ucp_st
Shortcuts
How to Add
What's New
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
Tuned values of UCP parameters =
                  Values
      xpart1     22.55186
      xpart2     38.91696
      xpart3     17.97726
      xpart4     22.53396
      xpart5     38.90902
      xpart6     17.97648
      xpart7     22.54125
      xpart8     38.90957
      xpart9     17.98418
     xpart10     22.53063
     xpart11     38.89275
     xpart12     17.97483
     xpart13     22.54292
     xpart14     38.90795
     xpart15     17.97397
     xpart16     22.53552

fx >>
```

Fig.5 Tuned UCP Parameter Values

6. CONCLUSION AND FUTURE WORK

In this paper, we have implemented meta-heuristic based search algorithm namely PSO. The particle swarm optimization algorithm is used for improving testing effort estimation. The particle swarm optimization algorithm (PSO) is applied on use case point (UCP) and results led us to the conclusion that test effort estimation can be optimized by applying PSO. The PSO optimization can also be applied for estimating effort of software development. This implementation increases the accuracy of testing effort estimation. In future, we will implement the other algorithms of meta-heuristic technique such as ant colony optimization, firefly algorithm etc. and generate a hybrid algorithm for increase the accuracy of test effort estimation.

7. REFERENCES

[1] S. Aloka, Peenu Singh, Geetanjali Rakshit, and Praveen Ranjan Srivastava. 2011, "Test Effort Estimation-Particle Swarm Optimization Based Approach", Springer-Verlag Berlin Heidelberg, CCIS.

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

[3] James Kennedy and Russell Eberhart, 1995, "Particle Swarm Optimization", IEEE Conference on Neural Networks, Piscataway, NJ.

[4] Vahid Khatibi Bardsiri, Dayang Norhayati Abang Jawawi ,Siti Zaiton Mohd Hashim and Elham Khatibi, 2013, "A PSO-based model to increase the accuracy of software development effort estimation", Springer Science+Business Media, Software Qual J.

[5] Jin-Cherng Lin, Yueh-Ting Lin, Han-Yuan Tzeng and Yan-Chin Wang, January 2013, "Using Computing Intelligence Techniques To Estimate Software Effort", International Journal of Software Engineering & Applications (IJSEA).

[6] Narmada Nayak and Durga Prasad Mohapatra, 2010, "Automatic Test Data Generation for Data Flow Testing Using Particle Swarm Optimization", Springer-Verlag Berlin Heidelberg, IC3 2010, Part II, CCIS.

[7] Aiguo Li, Yanli Zhang, 2009, "Automatic Generating All-Path Test Data of a Program Based on PSO", World Congress on Software Engineering, IEEE.

[8] Xiaochun Zhu, Bo Zhou, Li Hou, Junbo Chen, Lu Chen. 2008,"An Experience-Based Approach for Test Execution Effort Estimation", 9th International Conference for Young Computer Scientists, IEEE.

[9] Khaled Hamdan, Hazem El Khatib, Khaled Shuaib. 2010,"Practical Software Project Total Cost Estimation Methods", MCIT 10, IEEE.

[10] Srivastava, P. R., Varshney, A., Nama, P., & Yang, X. S.,2012, "Software test effort estimation: a model based on cuckoo search". International Journal of Bio-Inspired Computation, 4(5), 278-285.

[11] Priya Chaudhary and C.S. Yadav, 2012,"An Approach for Calculating the Effort Needed on Testing Projects". International Journal of Advanced Research in Computer Engineering & Technology.

[12] Srivastava, P. R., Bidwai, A., Khan, A., Rathore, K., Sharma, R., & Yang, X. S. , 2014,"An empirical study of test effort estimation based on bat algorithm", International Journal of Bio-Inspired Computation, 6(1), 57-70.

[13] M. Jorgensen and M. Shepperd,2006, "A systematic review of software development cost estimation studies",IEEE Transactions on Software Engineering.

[14] Eduardo Aranha, Paulo Borba,2007, "An Estimation Model for Test Execution Effort", 1st International Symposium on Empirical Software Engineering and Measurement.