An Optimizer using Particle Swarm Theory and its Application in Software Cost Estimation

Hina Agarwal[1], Zaved Akhtar[2]

[1]M.Tech. Student, VIET, G.B Nagar

[2]Department of Computer Science, VIET, G.B Nagar

Abstract: ***This paper is focused on the study of the shortcomings of PSO and to generate a new variant of PSO. In this new MPSO variant, the value of parameters has been tuned. Also a new parameter has been added in it. MPSO has been applied on real life problem, i.e., software cost estimation.***

Keywords: COCOMO Model, MPSO, PSO, SI

1. Introduction

Optimization is defined as the act of obtaining the best result under the given circumstances. It can also be defined as the process of finding the conditions that give the mini-mum or maximum value of a function, where the function represents the effort required or the desired benefit.

Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems. Particle Swarm Optimization (PSO) incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the idea has emerged. PSO is a population-based optimization tool, which could be implemented and applied easily to solve various function optimization problems.

A variant of PSO named as Modified Particle Swarm Optimization (MPSO) has been proposed to solve optimization problems. MPSO is designed to remove the shortcomings associated with the operators of PSO algorithm.

To improve the convergence rate of existing PSO, a change has been made in the velocity vector, inertia weight and learning factors of PSO. In PSO, velocity helps to move particle in the right direction to gain optimal position. A new factor is added in the velocity vector that helps particle to decide how much improvement is required to find new position with best fitness value. Inertia weight has been tuned enough to maintain balance between exploration and exploitation habit of the search space. Acceleration constants c1 and c2 are also tuned enough to control the movement of the particle in the search space.

The performance of this algorithm has been checked on a real life optimization problem. This real life problem is related to software effort estimation and its main objective is to minimize the difference between actual effort and predicted effort.

2. Objective

a) Able to perform better in higher dimensions: - It is required to capture global optimal solution in higher dimensions. This can be done by introducing a new algorithm which can capture global optimal solution in higher dimensions.
b) Improved convergence rate: - This newly developed algorithm must be as fast as compared to other state-of-the-art algorithms. It should require minimum number of generations to capture global optimal solution.
c) Able to capture optimal solutions with diversity: - It can be done by proper tuning of random parameters that helps to capture diverse optimal solutions for unimodal and multimodal problems.
d) Avoid Local Optima: - It must be able to escape from local optima. This can be done by proper tuning of random parameters, inertia weight and learning factors.

3. Parameter Tuning in PSO Algorithm

The challenge of identifying appropriate parameters for efficient performance of particle swarm optimization algorithms has been studied for many years. In this approach, the parameters of the PSO algorithm (inertia weight, cognitive acceleration coefficient and social acceleration coefficient) are tuned for improving the optimal solution in the search space. Parameter tuning strategy is needed because the basic version of the PSO algorithm was not giving efficient performance on the benchmarks functions. Proper and fine tuning of the parameters may result in faster convergence of the algorithm, and alleviation of the local minima. Initially, the values of the parameters of PSO algorithm were constant. However, experimental results proved that it is better to initially set the parameters to a large value, in order to promote global exploration of the search space, and gradually decrease it to get more refined optimal solutions. A large parameter’s value facilitates global exploration (searching new areas), while a small one tends to facilitate local exploration (fine tuning of the current search area). A suitable value for the parameters usually provides balance between global and local exploration abilities and consequently results in a reduction of the number of iterations required to locate the optimum solution.

4. Algorithm for Modified Particle Swarm Optimization (MPSO)

Step 1: Let initialize the iterative number for generation i = 0. Initialize the population size as POPsize. Maximum generation size for termination of the algorithm is Maxgen. Generate initial population of the particles of size POPsize.

Step 2: Calculate the fitness of each particle of the initialized population and for the first generation let Pid be the initialized particles. Select the particle with best fitness value among all particles as Gbest (gbest).

Step 3: Give birth to the particles (obtaining new positions) for the next generation using equation (4) and (5) as Xid. Calculate the fitness of the particles and compare with the best fitness value P_{id}^{i} i.e. pbesti in the history. Select the particles with better fitness value and take them as new P_{id}^{i}. Select the particle with best fitness value among all particles as G_{best}^{i}. Check if number of generation "i" is equal to maximum number of generation Maxgen or optimal position is obtained which comes first then go to Step 4 or else i = i + 1 go to Step 2.

Step 4: The global best position of the swarm is the optimal solution i.e. Gbest.

5. Software Cost Estimation

Software cost estimation is the most significant and challenging task for the development process of any software and software project management. It includes many factors like the cost of labour, hardware, tools etc. But effort is the most significant and dominating factor among all factors for any software project. Here effort is the amount of labour (persons) requisite for completing a software project. We have applied the proposed optimization algorithms to find out the optimal estimation of the effort required for a software project.

6. Solving Approach

It is highly expected to get the precise estimate of the cost, but there is as such no technique that can accurately predict the software development cost because of uncertainties, and imprecision associated with the software development process. This is a kind of optimization problem and due to this most of the researchers have explored the domain of natural phenomena based optimization techniques such as evolutionary computation, swarm intelligence, differential evolution etc. to form better estimation model. These techniques have tremendous exploration capabilities, and power to handle impression which motivated us to improve the performance of existing model. MPSO is a swarm based optimization technique that generates solutions with accuracy.

7. Model for Optimization

The model proposed by Sheta which is based on Boehm Constructive Cost Model (COCOMO) model has been tuned in this work. Sheta did some modifications in the Boehm's basic model for software cost estimation to provide generalized model for the estimation of effort for all types of projects. He used Genetic Algorithm to optimize the parameters of the proposed model to predict the exact estimation of effort. Sheta included methodology in the COCOMO model to gain the prediction accuracy, a bias term 'b' and another parameter 'd' is added in the basic model to improve the prediction accuracy as shown in equation (5.8).

The Sheta Model is as given below:

$$EE = a\,(DLOC)\,b + c\,(M) + d \qquad (5.8)$$

Here EE denote estimated effort. The objective is to find the generalized optimal value of all parameters using this proposed algorithms named MPSO and to provide most accuracy in the estimation of the effort required for all type of software projects. In the proposed work this model has been optimized using MPSO.

The goal of proposed algorithm is to generate the global optimal value of a, b, c and d which can minimize the difference between actual software effort (measured effort) and estimated software effort and applicable to all types of software. It can also minimize Mean Magnitude of Relative Error (MMRE).

```
Step 1: Initially generate random population POP including
        different parameters such  as a, b, c and d within the
        search domain shown in table (5.2). POP is generated
        in different dimensions as POPsize = POP * Dim,
        where Dim denotes the number of dimension and it
        size is equal to the number of parameters.
Step 2: While (number of generations (Numgen) < Maxgen)
        do
         // Repeat the following steps 3 to 10 until the
        specified Maxgen has reached.
Step 3: For i = 1, 2………. POPsize
          // perform the following operations
          Calculate EEi //Estimated Effort
          Calculate MREi // Magnitude of Relative   Error
        end for
Step 4: MMREi is calculated// Mean Magnitude of Relative
        Error
Step5:  Store and replace minimum MMRE and
        corresponding four values of parameters.
Step 6:  Apply MPSO algorithm to optimize the value of a, b,
        c and d.
Step 7: Sort the merged population in history with optimized
        value of a, b, c and d. Select the best population
        (POP) for the next generation.
Step 8:  Numgen ← Numgen + 1.
Step 9:  End while
Step 10: Select the minimum value MMRE and also select
         the corresponding value of four parameters where
         the optimum minimum value of MMRE has
         obtained.
```

Step 11: Print the final value of a, b, c, d and MMRE according to step10.

8. Conclusion

MPSO has been developed which is a variant of basic PSO algorithm. It is developed to remove the shortcomings of PSO. In MPSO, the value of parameter has been fine-tuned such as inertia weight (w), acceleration coefficient (learning factors) i.e. c1 and c2. Also, a new parameter 'e' is added in velocity vector to manage the velocity of particles.

MPSO performs well with both unimodal and multimodal problems and provides better results than other algorithms

MPSO has been applied on real life problem i.e. software cost estimation. In the application, four parameters of the cost/effort estimation models proposed by Alla F. Sheta have been tuned to improve the accuracy in the estimated effort. The performance of the proposed algorithm is checked over the fitness functions MMRE. We have also calculated MRE, VAR and PRED for MPSO and compared these values with other proposed methods.

9. Future Work

Apart from the software cost/effort estimation the algorithm namely MPSO can be applied to other real world optimization problems such as image processing, protein folding problem, test case generation etc.

The performance MPSO can be further improved by mapping the natural phenomenon with this algorithm. Searching technique can be further improved to guide the search in a precise direction.

References

[1] J. Kennedy and R. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers, 2001.
[2] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in Proceedings of the sixth international symposium on micro machine and human science, vol. 1. New York, NY, 1995, pp. 39–43.
[3] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", Proc. IEEE Int. Conf. Neural Networks, pp. 1942-1948, 1995.
[4] Shi, Yuhui, and Russell Eberhart. "A modified particle swam optimizer."Evolutionary Computation Proceedings, 1998. IEEE World Congress on Cmputational Intelligence, the 1998 IEEE International Conference on.IEEE, 1998.
[5] Pederson, M. E. (2010). Good Parameters for Particle Swarm Optimization. Technical Report, Hvass Laboratories.
[6] S. Tewolde, D. M. Hanna, and R. E. Haskell, "Enhancing performance of PSO with automatic parameter tuning technique", IEEE Swarm intelligence Symposium, 2009.
[7] J.E. Matson, B.E Barrett, J. M. Mellichamp, "Software Development Cost Estimation Using Function Points", IEEE Trans. on Software Engineering, 20(4) (1994) 275-287.
[8] Sheta, Alaa F. "Estimation of the COCOMO model parameters using genetic algorithms for NASA software projects." Journal of Computer Science 2.2 (2006): 118.

Author Profile

Hina Agarwal received B.Tech degree in Computer Science and Engineering from Uttar Pradesh Technical University in 2011. Now She is doing M.Tech from VGI G.B. Nagar, AKTU Lucknow She is doing her Dissertation in Swarm Intelligence.

Zaved Akhtar received his M.Tech in Computer Science and Engineering. He is pursuing his P.hd. from Dr. A.P.J. Abdul Kalam Technical University, Lucknow. He is working as an Associate Professor (CSE Department) in Vishveshwaraya Group of Institutions, Dadri, G.B. Nagar and he has abourt 13 years Experience. His area of interest is Software Engineering, Data mining and Algorithms.He has published more than 15 research papers in reputed International/National journals.