International Journal of Current Trends in Engineering & Technology
ISSN: 2395-3152
Volume: 02, Issue: 03 (MAY-JUNE, 2016)

Test-Case Prioritization by Using Binary Particle Swarm Optimization (PSO) Technique

Shruti Mishra[1], Narendra Sharma[2], Rishi Singh Kushwah[3]
Computer Science & Engineering Department, SSSIST, India
[1]shruti.mishra.766@gmail.com, [2]narendra_sharma88@yahoo.com, [3]rishisinghkushwah@gmail.com

***Abstract:* - One of the important phases in the development of the software system is the software testing. It evaluates the software and reveals the error in a given product. Therefore test case selection and prioritization are used to determine which test cases need to executed thereby making the results more effective. In the proposed approach for getting the next suitable test case an optimization technique is used called as Particle Swarm Optimization (PSO) technique. A fitness function is used for knowing the capability of the test case. With this fitness value test cases can be prioritized. Particle swarm optimization (PSO) method that is based on artificial intelligence (AI) technique. It is an optimization method that was developed in 1995 by Eberhart and Kennedy based on the social behaviors of fish schooling or birds flocking. In terms of software, it is a Model Based Technique which is helpful in finding the errors in the designing phase itself. Performance of PSO can be improved as suggested by researchers are by initialization of Swarm. Some of them introduced new parameters like constriction coefficient and inertia weight. On the other hand some of them defined different methods like inertia weight to improve the performance of PSO. This Paper gives an overview about Test-Case Prioritization and detailed functioning of Particle Swarm Optimization (PSO) Technique that can be very useful in the field of Software testing (Model Based Testing)**

***Keywords:* - Optimization, Prioritization, artificial intelligence, Swarm, inertia, constriction.**

I. INTODUCTION

Test case prioritization is the process of scheduling test cases to be executed in a particular order such that the test cases with a higher priority are executed earlier in the test sequence. Test cases are prioritized to increase a test suites rate of fault detection. It means that how quickly test suite detects faults in software to increase reliability. In PSO optimization includes finding "*best available*" values for some function given a defined domain or a set of constraints, which includes a variety of different types of objective functions and different types of domains and the term "*particles*" refers to population members which are mass-less and volume-less and are subject to accelerations and velocities towards a better mode of behavior. Particle swarm optimization (PSO) is an artificial intelligence (AI) technique that can be used to find approximate solutions to extremely difficult or impossible problems. In terms of computer science the particle swarm optimization (PSO) is a computational method which optimizes a problem by iteratively trying to improve a solution with a given measure of quality. By increasing the overall rate of fault detection, a greater number of errors can be found more rapidly in the code. As with the other optimization methods PSO has advantages and disadvantages too. Advantages of the basic particle swarm optimization algorithm are that PSO is based on the intelligence. It can be applied into both scientific research and engineering use. Then PSO have no and mutation calculation overlapping. On the other hand, disadvantages of the basic particle swarm optimization algorithm are the method easily suffers from the partial optimism, which causes the less exact at the regulation of its speed and the direction.

II. USED ELEMENTS IN PSO

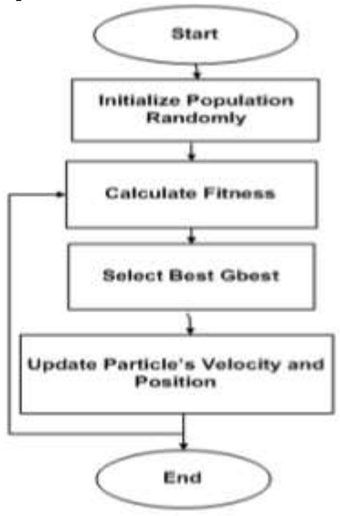
To know PSO, first of all we need to the elements of PSO in brief first on is Particle- We can define the particle as Pi for real numbers. Fitness Function- Fitness Function is the function used to find out the optimal solution. Usually it is an objective function. Local Best- It is the best position of the particle among its all positions visited so for. Global Best- The position where the best fitness is achieved among all the particles visited so for. Velocity Update- Velocity is a vector to determine the speed and direction of the particle. Position Update- All the particles try to move toward the best position for optimal fitness. Each particle in PSO updates their positions to find the global optima.

III. TEST CASE PRIORITIZATION

We define the test case prioritization problem as follows: The Test Case Prioritization Problem: Given: T, a test suite; PT, the set of permutations of T; f, a function from PT to the real numbers. Problem: Find T0 2 PT such that (8T00) (T00 2 PT) (T00 6= T0) [f (T0) f (T00)]. In this definition, PT represents the set of all possible prioritizations (orderings) of T, and f is a function that, applied to any such ordering, yields an award value for that ordering. (For simplicity the definition assumes that higher award values are preferable to lower ones). PSO makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, Meta heuristics such as PSO does not guarantee an optimal solution. However, PSO does not use the gradient of the problem being optimized, which means that PSO does not require that the

International Journal of Current Trends in Engineering & Technology
ISSN: 2395-3152
Volume: 02, Issue: 03 (MAY-JUNE, 2016)

optimization problem be differentiable as is required by classic optimization methods such as gradient descent and quasi-newton methods.



3.1 Algorithm

A basic variant of the PSO algorithm on a population (called a swarm) of particles. These particles are moved around in the search-space by following few simple formulae. The movements of the particles are guided to the best known position in the search-space. When improved positions are discovered the movements of the swarm is guided. The process is repeated again and again until a satisfactory solution is discovered. Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is the cost function which is to be minimized. The function takes an argument in the form of a vector of real numbers and a real number is produced as output which indicates the objective function value of the given Particle. The gradient of f is not known. The aim is to find a solution $\mathbf{x}$ for which $f(\mathbf{x}) \leq f(\mathbf{y})$ for all $\mathbf{y}$ in the search-space, which would mean $\mathbf{x}$ is the global minimum. By considering the function $h = -f$ instead, Maximization can be performed. Let the number of particles in the swarm be S, each having a position $\mathbf{x}_i \in \mathbb{R}^n$ in the search-space and a velocity $\mathbf{v}_i \in \mathbb{R}^n$. Let the best known position of particle i is $\mathbf{p}_i$ and let $\mathbf{g}$ be the best known position of the entire swarm. A basic PSO algorithm is then:

- For each particle $i = 1, ..., S$ do:
  - Initialize the particle's position with a uniformly distributed random vector: $\mathbf{x}_i \sim U(\mathbf{b}_{low}, \mathbf{b}_{upp})$, where $\mathbf{b}_{low}$ and $\mathbf{b}_{upp}$ are the lower and upper boundaries of the search-space.
  - Initialize the particle's best known position to its initial position: $\mathbf{p}_i \leftarrow \mathbf{x}_i$
  - If $(f(\mathbf{p}_i) < f(\mathbf{g}))$ update the swarm's best known position: $\mathbf{g} \leftarrow \mathbf{p}_i$
  - Initialize the particle's velocity: $\mathbf{v}_i \sim U(-|\mathbf{b}_{upp}-\mathbf{b}_{low}|, |\mathbf{b}_{upp}-\mathbf{b}_{low}|)$
- Until a termination criterion is met (e.g. number of iterations performed, or a solution with adequate objective function value is found), repeat:
  - For each particle $i = 1, ..., S$ do:
    - For each dimension $d = 1, ..., n$ do:
      - Pick random numbers: $r_p, r_g \sim U(0,1)$
      - Update the particle's velocity: $\mathbf{v}_{i,d} \leftarrow \omega\, \mathbf{v}_{i,d} + \varphi_p\, r_p\, (\mathbf{p}_{i,d}-\mathbf{x}_{i,d}) + \varphi_g\, r_g\, (\mathbf{g}_d-\mathbf{x}_{i,d})$
    - Update the particle's position: $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$
    - If $(f(\mathbf{x}_i) < f(\mathbf{p}_i))$ do:
      - Update the particle's best known position: $\mathbf{p}_i \leftarrow \mathbf{x}_i$
      - If $(f(\mathbf{p}_i) < f(\mathbf{g}))$ update the swarm's best known position: $\mathbf{g} \leftarrow \mathbf{p}_i$
- Now $\mathbf{g}$ holds the best found solution.

Table 1: Basic Variant of PSO

Basic Variant	Function	Advantages	Disadvantages
Velocity Clamping	Control the global exploration of the particle Reduces the size of the step velocity, so that the particles remain in the search area, but it cannot change the search direction of the particle	VC reduces the size of the step velocity so it will control the movement of the particle	If all the velocity becomes equal to the particle will continue to conduct searches within a hypercube and will probably remain in the optima but will not converge in the local area
Inertia Weight	Controls the momentum of the particle by weighing the contribution of the previous velocity.	A larger inertia weight in the end of search will foster the convergence ability.	Achieve optimality convergence strongly influenced by the inertia weight
Constriction Coefficient	To ensure the stable convergence of the PSO algorithm [21]	Similar with inertia weight	when the algorithm converges, the fixed values of the parameters might cause the unnecessary fluctuation of particles
Synchronous and Asynchronous Updates	Optimization in parallel processing	Improved convergence rate	Higher throughput More sophisticated finite element formulations Higher accuracy (mesh densities)

3.2 Parameter selection

The choice of PSO parameters greatly affects the optimization performance. Therefore it is very important that the parameters must be selected as such that the maximum output can be obtained. By using another overlaying optimizer, a concept known as meta-optimization, the parameters can be tuned.

3.3 Inner workings

There are several different opinion how the PSO algorithm can perform optimization. A common belief is that the behavior of the swarm varies between exploratory behaviors means searching a broader region of the search-space, and exploitative behavior, that is, a locally searched so as to get the best optimum result. Another school of thought is that the behavior of a PSO swarm is not well understood in terms of how it affects actual optimization performance, especially for higher-dimensional search-spaces and optimization problems that may be discontinuous, noisy, and time-varying. This school of thought rarely tries to find PSO algorithms and parameters that cause good. Such studies have led to the simplification of the PSO algorithm.

IV. CONCLUSION

Several parameters have been found to improve the performance of PSO. These include: initialization of population- If these particles be initialized correctly then the performance can be enhanced to a great extent, inertia weights- balance the exploration-exploitation trade off, mutation operators- the global best particle the local best particle with different techniques can be mutated velocity clamping. If the velocity of a particle exceeds the maximum allowed speed limit, it will set a maximum value of the velocity. Some environments are proposed which can improve the performance of PSO. These include: multi-objective optimization with PSO and dynamic environment with PSO. Many different variants of PSO have also been proposed here.

REFERENCES

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", Proc. IEEE Int. Conf. Neural Networks., (1995) November-December, pp. 1942–1948.

[2] S. Vijayalakshmi, D. Sudha, S. M. Sigamani and K. K. Devi, "Particle Swarm Optimization with Aging Leader and Challenges for Multwaswarm Optimization", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) vol. 3, no. 3, (2014) March.

[3] Clerc, M. (2004). Discrete Particle Swarm Optimization, illustrated by the Traveling Salesman Problem, New Optimization Techniques in Engineering, Springer, pp. 219-239

[4] K.El-Far and J.A. Whittaker, "Model-Based Software Testing", In John J. Marciniak, "Encyclopedia of Software Engineering", vol. 1, pp. 825837, Wiley-Inter Science, 2002.

[5] M. Imran, R. Hashima and N. E. A. Khalid, "An Overview of Particle Swarm Optimization Variants", Malaysian Technical Universities Conference on Engineering & Technology 209, MUCET 209, Procedia Engineering, vol. 53, (2013), pp. 491– 496.

[6] P. Umapathy, C. Venkataseshaiah and M. S. Arumugam, "Particle Swarm Optimization with Various Inertia Weight Variants for Optimal Power Flow Solution", Discrete Dynamics in Nature and Society vol. (2010).

[7] M. B. Ghalia, "Particle Swarm Optimization with an Improved Exploration-Exploitation Balance," IEEE, vol. 978-1-4244-2167, 2008.

[8] Q. Bai, "Analysis of Particle Swarm Optimization Algorithm," Computer and Information Science, vol. volume 3 No 1, February 2010.

[9] J. C. Bansal and K. Deep, "A modified binary particle swarm optimization for knapsack problems," Applied Mathematics and Computation, vol. 218, no. 22, pp. 11042–11061, 2012.

[10] Shi, Y.; Eberhart, R.C. (1998). "Parameter selection in particle swarm optimization". Proceedings of Evolutionary Programming VII (EP98). pp. 591–600.

[11] Trelea, I.C. (2003). "The Particle Swarm Optimization Algorithm: convergence analysis and parameter selection". Information Processing Letters 85 (6): 317–325.

[12] Engel Brecht, "particle Swarm Optimization: Pitfalls and convergent aspect "O. Schutze, et al., "A Memetic PSO Algorithm for Scalar Optimization Problems," in Swarm Intelligence Symposium, 2007. SIS 2007. IEEE, 2007, pp. 128-134.

[13] International Journal of Computer Applications (0975 – 8887) Volume 14– No.1, January 2011

[13] Hindawi Publishing Corporation, "Computational and Mathematical Methods in Medicine", Volume 2015, Article ID 713898, 12 pages

[14] M. A. Arasomwan and A. O. Adewumi, "On the performance of linear decreasing inertia weight particle swarm optimization for global optimization," The Scientific World Journal, vol. 2013, Article ID 860289, 12 pages, 2013.

[15] Y. H. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98), pp.69–73,Anchorage, Alaska, USA, May 1998.