www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 7 July 2015, Page No. 13292-13297

Review of Evolutionary Optimization Algorithms for Test Case Minimization

Aditya Vikram Sharma
adityasharma720@gmail.com

Abstract* - Multi-objective test suite minimization problem is to select a set of test cases from the available test suite while optimizing the multi objectives like code coverage, cost and fault history.[1] Regression Test suite optimization is an effective technique to reduce time and cost of testing. Many researchers have used computational intelligence techniques to enhance the effectiveness of test suite. These approaches optimize test suite for a single objective. Introduction of nature inspired algorithms like GA, PSO and BFO may be used to optimize test suite for multi-objective selection criteria. Main focus of our approach is to find a test suite that is optimal for multi-objective regression testing.[2]

***Keywords* – Regression testing, Test suite minimization, Bacterial Foraging Optimization Algorithm.**

I. INTRODUCTION

As any software system is developed changes are made to the software. Changes are done to introduce new features and functionalities. So after upgradation it is necessary to test the software to make sure that the system is working as intended. Hence, during regression testing the new test cases along with the old ones are executed to certify the functionality of the software. So, it becomes a tough task to carry out regression testing as size of test suite grows.[1]

In order to assist the software engineer in regression testing, test suite minimization techniques can be used

Test Suite Minimization Approach: Initially**,** a test suite T is given with all the possible test cases to test the software completely. Then use some algorithm to reduce T to get the test suite reduction T '.

T ' is not redundant, meaning that if any of the test cases is removed from T ' , the rest of the test case does not meet all the requirements.[3]

Test suite can be optimized based on fault detection, execution time and coverage given in eqn. (1),

$$Min(ET) \wedge Max(Cov) \wedge Max(FD) \wedge Min(S) \tag{1}$$

Where ET= Execution Time, Cov = Path Coverage, FD= Fault detection, S= Test Suite Size.[2]

The NP-completeness nature of test suite minimization problem inspired many researchers to experiment with different heuristics for its solution.

In recent years, biological intelligent heuristic optimization algorithms have become one of the mainstream methods to solve the non-linear, non-differential, multi-peak and complex problems. Many different bionic algorithms have been introduced by scholars from different countries, inspired from the foraging behaviors in the nature creature. Dorigo M et al. proposed the *Ant Colony Optimization (ACO)* [4] in 1991; Eberhart and Kennedy proposed the *Particle Swarm Optimization (PSO)* [5] in 1995; Passino et al proposed the *Bacterial Foraging Optimization (BFO)* [6] in 2002. Because of the advantages of parallel searching, jumping out of local minimum easily and so on, BFO is becoming a hot spot of bionic algorithm. Since the researching work of this algorithm in China is at the beginning stage and the randomness of bacterial chemotaxis in the algorithm, it leads to the slow chemotaxis speed and inefficient. So that, combining some common mechanisms

and principles of intelligent bionic algorithm with the differences in the internal operation mechanism becomes a natural way to optimize the algorithm.[7] The optimization algorithms are explained below :

A. *Genetic Algorithm*

Genetic Algorithms are population-based general purpose algorithms used to find accurate or estimated solutions to optimization and search problem.

They are stochastic search techniques based on the phenomenon of natural selection and genetics.GA begins with an initial population which is a random set of solutions. Each individual solution in the population is called a Chromosome. A chromosome can be a binary digit or any other data structure. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measure of fitness.

Selection, Crossover and Mutation are three basic operators responsible for GA and these are described below:

1. Selection : A new generation is formed by selecting those chromosomes that satisfy the fitness value criteria. Suitable chromosomes with higher probability are selected. Some parents and offsprings are retained while others are rejected so as to keep the population size constant. After several generations the algorithm converge to optimal or near optimal solution.

2. Crossover : the exchange of parents' information produces an offspring, as shown in figrure 1.

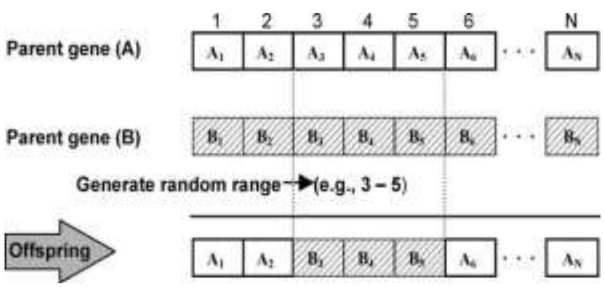

Fig 1: Crossover operation.[14]

3. Mutation : Randomly change one or more digits in the string representing an individual.

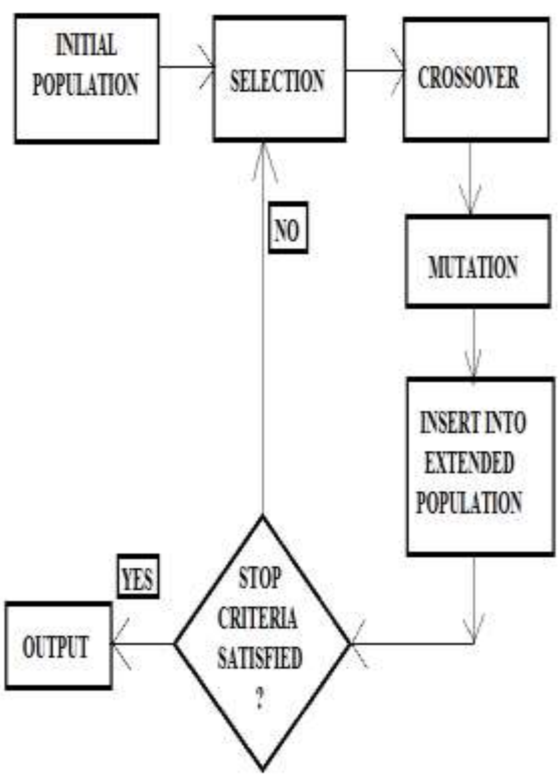

Fig 2: Genetic Algorithm Flow-Chart.[15]

B. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based meta-heuristic algorithm developed from the simulation of social models of bird flocking, fish schooling, and swarming able to find best possible solution(s) to the non-linear numeric problems. PSO was first introduced in 1995 by Eberhart and Kennedy. However, PSO can easily be trapped in local optimal point when dealing with some complex and multimodal functions.

PSO involves a number of particles, which are initialized randomly in the space of the design variables. These particles fly through the search space and their positions are updated based on the best positions of individual particles and the best position among all particles in the search space which in truss sizing problems corresponds to a particle with the smallest weight[5]. In PSO, a swarm consists of N particles moving around in a D-dimensional search space. The position of the jth particle at the kth iteration is used to evaluate the quality of the particle and represents candidate solution(s) for the search or optimization problems. The update moves a particle by adding a change velocity V_j^{k+1} to the current position X_j^k as follows:

$$V_j^{k+1} = wV_j^k + c_1 \times r_{1j}^k \otimes \left(P_j^k - X_j^k\right) + c_2 \times r_{2j}^k \otimes (P_g^k - X_j^k) \quad (2)$$

$$X_j^{k+1} = X_j^k + V_j^k \quad (3)$$

where w is an inertia weight to control the influence of the previous velocity; r_{1j}^k and r_{2j}^k are random numbers uniformly distributed in the range of (0,1); c_1 and c_2 are two acceleration constants called cognitive and social parameter, respectively; P_j^k is the best position of the jth particle up to iteration k; P_g^k is the best position among all particles in the swarm up to iteration k. In order to increase PSO's exploration ability, the inertia weight is now modified during the optimization process with the following equation:

$$w^{k+1} = w^k \times D_r \times rand \quad (4)$$

where D_r is the damping ratio which is a constant number in the interval (0,1); and rand is a uniformly distributed random number in the range of (0,1).[18]

Unlike GA there are no selection, crossover and variation operation in PSO. So its algorithm is very simple and has a high execution speed.

However if some particle finds a present optimal point then other particle will be closed to it rapidly. Hence the diversity of whole swarm and its global searching ability will be weakened obviously. [17]

C. *Ant Colony Optimization*

Ant Colony Optimization(ACO) algorithm was proposed by Marico Dorigo in 2005.[8] ACO is a probabilistic technique for solving computational problems which can be used for searching shortest paths.[9]

ACO deals with two important processes, namely: Pheromone deposition and trail pheromone evaporation. Pheromone deposition is the phenomenon of ants adding the pheromone on all paths they follow. Pheromone trail evaporation means decreasing the amount of pheromone deposited on every path with respect to time. Updating the trail is performed when ants either complete their search or get the shortest path to reach the food source.[16]

Basic Principle of the algorithm is that Ants always find a shortest path between the nest to food source, which mainly depends on a hormone-pheromones. The shorter path contains more pheromone, the probability of choosing that path by ants is greater and finally ants colony will find a shortest path.[3]

ACO technique has been already used in solving various combinatorial problem such as knapsack problem, travelling salesman problem, distributed network, telecommunication network, vehicle routing, test data generation.[9]

Though ACO is next generation technique for optimization problems but it is not providing good solutions of problems like multiple objectives optimization, Dynamic Optimization Problems, the Stochastic Optimization Problems, continuous optimization and Parallel Implementations of the constraints.[10]

Most ant colony optimization algorithms use this algorithm demonstrated below :[11]

Initiation of the parameters which determines the pheromone trail.
While (until result conditions supplied) **do**
Generate Solutions
Apply Local Search
Update Pheromone Trail
End

D. *BFO Algorithm*

Bacteria Foraging Optimization Algorithm (BFOA), given by Passino in 2002 , belongs to nature-inspired optimization algorithms. Main idea behind the algorithm is group foraging strategy of E.Coli. bacteria in order to maximize energy obtained per unit time. Communication also occurs between individual bacterium to improve the searching strategy. This algorithm consists of four prime steps[13] :

1. Chemotaxis**:** Here, swimming and tumbling are the two prime ways which define the manner in which bacteria search for food. Swimming means moving in a pre-specified direction. Tumbling means moving in a completely new direction. Mathematically, tumble of any bacterium can be given by multiplication of $\phi(j)$ and $C(i)$,where $\phi(j)$ is unit length in random direction and $C(i)$ is step length. In case of swimming, $C(i)$ is constant.

2. Swarming: For the algorithm to converge at the optimal solution, it is required that the optimum bacteria attract other bacteria so that together they converge at the solution point quickly. To achieve this, a penalty function is added to the original cost function on the basis of relative distances of each bacterium from the fittest one. Penalty function becomes zero when all the bacteria have reached to the solution point.

3. Reproduction: Here, the fittest bacteria are divided into two groups. The weaker set of bacteria are replaced by other more fit set of bacteria. This keeps the population of bacteria constant throughout the evolution process.

4. Elimination and Dispersal: Because of changes in environment some bacteria may be killed or may be dispersed to a new place. In BFOA, this phenomenon is simulated by liquidating some bacteria and initialing new replacements randomly in the search space. It helps in reducing the probability of being trapped in pre-mature solution point.[12]

Table 1: Existing Approaches to optimize Regression Test Suite

Genetic Algorithm	**Particle Swarm Optimization[17]**	**Ant Colony Optimization Algorithm**	**Bacterial Foraging Optimization Algorithm [13]**
Genetic Algorithms can be applied to virtually any problem that has a large search space.[19]	Simple Mathematical Model	ACO technique has been already used in solving various combinatorial problems such as knapsack problem, travelling salesman problem.	Widely accepted as a global optimization algorithm of current interest for distributed optimization and control.

Genetic algorithm are computationally slow.[20]	Unlike GA, there are no selection, crossover and mutation operation in PSO. So, its algorithm has high execution speed.	Not providing good solutions of problems like multiple objectives optimization, Dynamic Optimization Problems, the Stochastic Optimization Problems	Multi-optimal function optimization is the key idea of the new algorithm.
There is a problem of local optimum points but mutation and crossover help get out of this.	Standard PSO often falls into local optimal points.	Increasing the number of ants used to tackle a large problem almost yield to a worse algorithm performance.[11]	Ability to escape from local optimal point due to elimination and dispersal step.

Table 2: Existing Approaches to optimize Regression Test Suite.[2]

Authors	Algorithm used	Objective	Limitation	Year
Krishnamoorthi et al	Genetic Algorithm	Maximized code coverage.	Time constrained execution environment.	2009
Nachiyappan et al	Genetic Algorithm	To find a test suite with minimum test size.	To find test cases that have maximum coverage.	2010
Suri et al	Ant Colony Optimization	Test suites covering the paths with minimum time were selected for final testing	Time constrained regression testing.	2011
Subramanian et al	Mutant gene algorithm	To find test suite with greater fitness based on mutation score.	Branch-type coverage measures are chosen as the test adequacy criteria.Path-coverage can also be considered.	2011
Kaur et al	Hybrid algorithm of Particle swarm optimization and mutation	To find a test suite that finds maximum faults in minimum time.	The HPSO depends on randomly generating a mutant that makes the execution time quite long.	2011a
Kaur et al	Genetic Algorithm	To optimize test cases that cover all independent paths with minimum number of test cases.	Time constrained regression testing.	2011b
Kaur et al	Particle Swarm Optimization (PSO) with cross over	Test cases that cover the entire path or find all faults are selected as global best.	Time constrained regression testing.	2011c
Kaur et al	Genetic Algorithm	Test suite optimization in Minimum time.	To find test cases that have maximum coverage.	2011d
Kaur et al	Bee Colony Optimization	The primary objective is to uncover all faults and time minimization is second objective.	Requires manual interface for the input test suite data making the technique restricted to small sized test suite.	2011e

Sehrawat et al	Neuro Genetic Algorithm	Whole Path Coverage.	Time constrained regression testing.	2012
Suri et al	Bee Colony Optimization + Genetic Algorithm	Primary Objective is finding “all faults”. Secondary is “time minimization”.	Approach is not repeatable	2012
Vivekanandan et al	Ant Colony Optimization	To find the faults earlier or “in minimum time span”.	Time constrained regression testing.	2012
Maia et al	Weighted sum approach	Select optimal test path.	Will not be successful to find an aggregate weightage of each test.	2012
Suri et al	Swarm optimization and GA hybrid approach.	Reduction in Test-Suite.	The technique can provide different results in each run.	2012
Ming Chn et al[17]	PSO combined with mutative scale chaos method.	To mitigate the slow convergence and local optimum points in Standard PSO algorithm.	Results are not repeatable.	2012

ACKNOWLEDGMENT

I wish to express my deep gratitude to Guide, Assistant Professor, Computer Science & engineering department, Deenbandhu Chhotu Ram University of Science & Technology for providing his uncanny guidance and support throughout the paper.

REFERENCES

[1] A.Charan Kumari, K.Srinivas, M.P.Gupta (2012). Multi-objective test suite minimization using quantum-inspired multi-objective differential evolution algorithm. IEEE, 978-1-4673-1344-5/12.

[2] Aftab Ali Haider, Shahzad Rafiq, Aamer Nadeem (2012). Test Suite Optimization using Fuzzy Logic. IEEE, 978-1-4673-4451-7/12.

[3] Cui Donghua, Yin Wenjie (2011). The Research of Test-Suite Reduction Technique, IEEE, 978-1-61284-459-6/11.

[4] Colorni A, Dorigo M, Maniezzo V (1991). Distributed optimization by ant colonies. Proceedings of ECAL’91, European Conference on Artificial Life. Paris, France: Elsevier Publishing (134 – 142).

[5] Eberhart R C, Kennedy J (1995). A New Optimizer Using Particle Swarm Theory. Proc. The Sixth Int. Symposium on Micro Machine and Human Science, Nagoya Japan, IEEE Robotics and Automation Society (39-43).

[6] Passino K M (2002). Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine (52-67).

[7] Liu Xiao Long, Li RongJun, YangPing (2010). A Bacterial Foraging Global Optimization Algorithm Based On the Particle Swarm Optimization, IEEE, 978-1-4244-6585-9/10.

[8] M.Dorigo (2006). Ant Colony Optimization: Artificial Ants as Computational Intelligence Technique, IEEE Computaional Intelligence Magazine.

[9] Bharti Suri, Isha Mangal (April 2012). Analyzing Test Case Selection using Proposed Hybrid Technique based on BCO and Genetic Algorithm and a Comparison with ACO, IJARCSSE, Volume 2, Issue 4, ISSN: 2277 128X.

[10] Manoj Kumar, Arun Sharma, Rajesh Kumar (November 2011). Optimization of Test Cases using Soft Computing Techniques: A Critical Review. WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, Issue 11, Volume 8, ISSN: 1790-0832.

[11] Teerapun Saeheaw, Nivit Charoenchai, and Wichai Chattinnawat (December 2009). Application of Ant colony optimization for Multi-objective Production Problems, World Academy of Science, Engineering and Technology Vol 36.

[12] Swagatam Das, Arijit Biswas, Sambarta Dasgupta, and Ajith Abraham. Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications.

[13] Nikhil Kushwaha, Vimal Singh Bisht
Gautam Shah (2012). Genetic Algorithm based Bacterial Foraging Approach for Optimization. International Journal of Computer Applications (IJCA).

[14] Emad Elbeltagi, Tarek Hegazy Donald Grierson (2005). Comparison among five evolutionary-based optimization algorithms. Advanced Engineering Informatics 19 (43–53).

[15] Siba Prasada Tripathy, Debananda Kanhar (February 2013). Optimization of Software Testing for Discrete Test suite using Genetic Algorithm and Sampling Technique. International Journal of Computer Applications (0975 – 8887) Volume 63– No.

[16] Priyanka Bansal (2013). A Critical Review on Test Case Prioritization and Optimization using Soft Computing Techniques. 2nd International Conference on Role of Technology in Nation Building (ICRTNB) ISBN: 97881925922-1-3.

[17] MING CHEN, TAO WANG, JIAN FENG, YONG-YONG TANG1, LI-XIN ZHAO (2012). A Hybrid Particle Swarm Optimization Improved by Mutative Scale Chaos Algorithm. 2012 Fourth International Conference on Computational and Information Sciences, 978-0-7695-4789-3/12 IEEE.

[18] A. Kaveh, R. Sheikholeslami, S. Talatahari, M. Keshvari-Ilkhichi (2014). Chaotic swarming of particles: A new method for size optimization of truss structures. Advances in Engineering Software 67, 136–147.

[19] Muhannad Harrim. Genetic-Algorithms. Retrieved from https://www.cs.wmich.edu/~elise/courses/cs6800/ Genetic-Algorithms.ppt

[20] Soft Computing Paradigm. Lecture Module 23. Retrieved from http://www.cse.iitd.ac.in/~saroj/AI/ai2013/L23.ppt