A Comparative Analysis of Optimization Techniques

Kanika Tyagi  
M.Tech Computer Science and Engineering  
Ajay Kumar Garg Engineering College  
Ghaziabad, India

Kirti Tyagi  
Department of Computer Science and Engg.  
Ajay Kumar Garg Engineering College  
Ghaziabad, India

ABSTRACT
Regression testing is an inescapable and very expensive task to be performed, often in a resource and time constrained environment. The goal is to minimize the time spent in the process of testing by reduction in the number of test cases to be used. Thus various techniques are being used for test case optimization, to select the less indistinguishable test cases while providing the best possible fault coverage. This paper presents a comparative analysis of the different test case optimization techniques. There are various optimization techniques available for the context. This review explains about the different optimization techniques on the basis of their evolution, methodology, performance and applications.

Keywords
Optimization techniques, evolution, applications, regression testing.

1. INTRODUCTION
Software maintenance is defined as activity performed on a software product which includes enhancements, optimization, error corrections, and deletion of inoperative functionalities. Regression testing is conducted during maintenance phase of Software Development Life Cycle and it is defined as “the process of re-testing the modified components of the software and ensuring that no new errors have been encountered in previously tested code”[1].

The approach for performing regression testing is to again execute the test cases. Test cases are set of conditions under which a tester will determine whether an application or its features are functioning as it was originally established to do. Regression testing is an expensive activity, running all test cases requires large amount of time and cost. So, it is required to optimize test cases to reduce the efforts and cost. Optimization tends to provide the best output with the least investments [2]. The goal of optimization techniques for test cases is to minimize the number of test cases without affecting the fault coverage of the testing process.

Component Based System (CBS) is the emerging technology in last few years. It is a more generalized approach for software development. CBS are mainly developed using reusable components and Commercial-On-The-Shelf components [3]. Due to this, CBSs are developed with minimum engineering efforts and resource costs and time. CBS provide a new approach to construct, design and implement software applications. Software applications are foregathered from variety of components that may be written in different languages and may run on different platforms. That is, components are heterogeneous.

There are various advantages of using CBS for developing applications. Some of them are listed below:

- Reusability: A code designed for a particular functionality may be used in a number of applications which need the same functionality, a programmer need not to write the same code multiple times.
- Interoperability: Components have the ability to communicate, execute and exchange data among themselves without any need of knowing about underlying structure of the system.
- Upgradable: The application can be upgraded easily if a new component has been introduced during the lifetime of the application.
- Time and cost effective: As in CBS, code for a component can be used again and again in similar type of applications. This makes it a time and cost efficient approach.

In this paper, a critical review of different test case optimization techniques is represented. Different techniques are compared with each other based on their evolution, methodology, performance and applications.

2. COMPARATIVE ANALYSIS
This section covers the comparative analysis of different optimization techniques based on the parameters like: Evolution, Methodology, Performance and Applications.

Evolution: It covers the evolution of the techniques all through their lifetime. It includes the growth of the technique from its development.

Methodology: It describes the way in which the algorithm works and identifies the methods used in it. It shows how a result is to be calculated.

Performance: This section covers the overall performance of the techniques. It also covers how performance may be achieved by improvements in the technique and hybridization with other techniques.

Applications: Application is defined as the utility of a technique, technology or a system. It covers where these techniques may be applied to give a better result.

2.1 Artificial Bee Colony Optimization Algorithm
Artificial Bee Colony Optimization (ABC) is an optimization technique which provides a search procedure based on population, where artificial bees modify the food positions [29]. The colony of bees is comprised of three classes of bees (1) employed bees, (2) onlookers and (3) scouts. A possible solution to the problem is represented by the position of food source and the nectar amount corresponds to the fitness or efficiency of the solution.

Evolution: Artificial Bee Colony Optimization algorithm was developed by Karaboga in year 2005, inspired from foraging and waggle dance behaviour of honey bee colonies [4]. Since 2005 various modifications have been done in bee colony algorithm which are given in the Table 1.
Methodology: ABC inhibits the behaviour of natural ants which is based on the random behaviour of ants as they wander randomly to find path [5]. ABC aims at generating the optimal number of test cases with fewer amounts of time and resources. Bees tend to find out the food sources with higher nectar amount. Depending on the experiences of themselves and their nest mates Employed and Onlookers bees select food sources [6].

Performance: ABC is a simple approach, yet it deals with complex problems efficiently. This technique uses fewer parameters as compared to other search algorithms [7]. It is efficient to hybridize ABC with other techniques to increase its performance. Some hybridized ABC are GABC [8], CABC [9], P-DABC [10], hybrid ABC [11], ABC with neural networks [12] and many more.

Applications: There are many applications of ABC algorithm to real world and benchmark optimization problems. It was applied to integer programming problems [13]. Travelling Salesman problem [14], Bio-Informatics applications [15], scheduling applications, clustering, image processing and many more.

2.2. Particle Swarm Intelligence Optimization

Particle Swarm Intelligence Optimization (PSO) simulates swarming behaviour observed in herds of animals, flocks of birds etc., where social sharing of information takes place and individuals gain from the discoveries and experiences of other companions while searching for food. System is initialized with a random population, known as particles. During optimization particles explore a D-dimensional space. Each particle maintains its own current position, current velocity and best position so far [16]. The iterative update of rules leads to a stochastic manipulation of velocities.

Evolution: Kennedy and Eberhart introduced a heuristic global optimization method known as PSO in year 1995 in a research paper named “Particle Swarm Optimization” [17]. Since then, various modifications and improvements are done. Some of them are listed in Table 2.

Methodology: PSO is a global optimization algorithm which deals with problems which tends to find the best solution as a point in a D-dimensional space. In PSO, particles fly in the respective problem space by following the current optimum flying particles. Each of the particle keeps a track of its position in terms of coordinates in the problem space which are contributes to the best solution so far. Particles then move in solution space and they are evaluated after each iteration through some fitness functions.

Performance: PSO is a population based algorithm based on the cooperation of each particle. The convergence ability of PSO is faster than other optimization techniques [18]. It requires fewer parameters for the calculation of optimizing value. Number of particles may be decreased to increase the performance [19]. It may be hybridized to further increase its performance.

Table 1: Evolution of ABC Algorithm

<table>
<thead>
<tr>
<th>Year</th>
<th>Author</th>
<th>Name of algorithm</th>
<th>Problem constrained</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>Karaboga et al.</td>
<td>Modified artificial bee colony optimization (MABC)</td>
<td>Constrained optimization</td>
</tr>
<tr>
<td>2009</td>
<td>P.W.Tsai et al.</td>
<td>Interactive artificial bee colony optimization (IABC)</td>
<td>Numerical optimization</td>
</tr>
<tr>
<td>2010</td>
<td>M. Sonmez.</td>
<td>Artificial bee colony with adaptive penalty function (ABC-AP)</td>
<td>Weight of truss structures</td>
</tr>
<tr>
<td>2010</td>
<td>A. Aderhold et al.</td>
<td>ABCgBest and ABCgBestDist</td>
<td>Benchmark functions</td>
</tr>
<tr>
<td>2011</td>
<td>N. Taspnar</td>
<td>Partial transmit sequences (PTSs) based on ABC (ABC-PTS)</td>
<td>Peak-to-average power ratio</td>
</tr>
<tr>
<td>2011</td>
<td>M. El-Abd.</td>
<td>ABC with the concept of opposition number based optimization (OABC)</td>
<td>Black box optimization</td>
</tr>
<tr>
<td>2014</td>
<td>Deepak Rai et al.</td>
<td>Honey bee mating optimization algorithm (HBMO)</td>
<td>Test case optimization</td>
</tr>
</tbody>
</table>

Applications: PSO have a vast range of applications. It may be applied to optimize multi-objective problems [20], T-way software testing [21], mini-max problems [22], image classification, job shop scheduling, artificial neural networks, gesture recognition and many more.

2.3. Ant Colony Optimization Algorithm:

Ant Colony Optimization (ACO) is an intuitive algorithm influenced by the behaviour of natural ants. Ants are blind and have the tendency to find the shortest path from their initial position to the food source. Pheromone, the chemical used for unintended communication between ants helps to search the shortest path.

Evolution: In year 1991, Ant System was first proposed by...
Marco Dorigo in his doctoral thesis. Afterwards, some improvements were made into Ant System as the introduction of Elitist ants [23], the ranking of ants [24]. One of the major developments is the description of ACO metaheuristic in year 1999 [25]. Other modifications and variations of ACO are given in the Table 3.

**Methodology:** Ant Colony Algorithm is inspired from the behaviour of natural ants. Ants resolve their problems by collaborating with each other using pheromone. Ants place pheromone on the ground while walking and each ant follow a direction where pheromone intensity is high. When an ant recognizes an obstacle in the path it changes the direction and tries to find new shortest path.

**Performance:** ACO has the ability to find optimal solution in less computational time. The performance of ACO may be improved by introducing approaches like modification of transition rule, parallel ACO [26]. It may be hybridized with other techniques for better results.

**Applications:** The first combinatorial problem solved using ACO was Travelling Salesman Problem (TSP) in 1991 by Dorigo in his Phd dissertation. The nest two applications of ACO were the Quadratic Assignment Problem (QAP) [27] and Job Shop Scheduling [28] 1994. Then it was applied to network routing applications [29], vehicle routing problems [30], sequential ordering, graph colouring problems and design of algorithms for knowledge representation structures.

**Table 2: Evolution of PSO Algorithm**

<table>
<thead>
<tr>
<th>Year</th>
<th>Author</th>
<th>Name of algorithm</th>
<th>Problem constrained</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>Van Den Bergh</td>
<td>Multi-start particle swarm optimization (MPSO)</td>
<td>Encountering a global minimiser</td>
</tr>
<tr>
<td>2002</td>
<td>Van Den Bergh</td>
<td>Guaranteed Convergence Particle Swarm Optimization (GCPSO)</td>
<td>Convergence to local minimum</td>
</tr>
<tr>
<td>2005</td>
<td>Stefan Janson and Martin Midden Dorf</td>
<td>Hierarchical Particle Swarm Optimization (HPSO)</td>
<td>Better solution</td>
</tr>
<tr>
<td>2005</td>
<td>Chunning Yang and Dan Simon</td>
<td>New Particle Swarm Optimization</td>
<td>Better solution</td>
</tr>
<tr>
<td>2007</td>
<td>Hui Wang et al.</td>
<td>Opposition based Particle Swarm Optimization (OPSO)</td>
<td>To accelerate the convergence</td>
</tr>
<tr>
<td>2008</td>
<td>Marco A. Montes Dea et al.</td>
<td>Fully Informed Particle Swarm Optimization (FIPSO)</td>
<td>Optimization problems</td>
</tr>
<tr>
<td>2009</td>
<td>George I. Evers et al.</td>
<td>Regrouping Particle Swarm Optimization (RPSO)</td>
<td>Premature convergence</td>
</tr>
<tr>
<td>2011</td>
<td>X.S. Yang, s. Fong and S. Deb</td>
<td>Accelerated Particle Swarm Optimization (APSO)</td>
<td>Accelerate convergence</td>
</tr>
<tr>
<td>2012</td>
<td>R. Roy et al.</td>
<td>Novel Particle Swarm Optimization (NPSO)</td>
<td>Multi-Objective Combinatorial Optimization</td>
</tr>
</tbody>
</table>

2.4. Genetic Algorithm:

Genetic Algorithm is an adaptive heuristic search method based on the population genetics. It is probabilistic search method inspired from the process of natural selection and reproduction. It is used to generate solutions of optimization search problems. It belongs to the class of evolutionary algorithms which uses initialization, selection, crossover and mutation.

**Evolution:** John Holland introduced Genetic Algorithm in his book named ‘Adaptation in Natural and Artificial Systems’ in 1975 [31]. Then John Koza has used genetic algorithm in 1992 to evolve programs to perform certain tasks. He called his method as ”Genetic Programming” (GP). Some improvements in GA are listed below in Table 4.

**Methodology:** A GA starts with a set of solutions called as population. It consists of four operators: (1) Initialization, (2) Selection, (3) Crossover and (4) Mutation. Initialization operator creates the initial population and assigns a fitness function which evaluates the fitness value. The selection operator chooses the chromosomes from population for mating. Crossover operator is used for sharing the information between two chromosomes. Mutation operator alters one or more gene values of a chromosome from its initial state. The process of evolution is repeated until end condition of the problem is satisfied.

**Performance:** GA is a search technique used to find the exact or approximate solutions to an optimization problem. GA exhibits “inherent parallelism” as the evaluation of individuals within a population is conducted simultaneously. It generally finds the global optima in complex spaces, hence GA is fast [32]. GA may be hybridized with any other search method to achieve an optimization goal.
Applications: Various application areas of GA are: global optimization problems, scheduling problems, power system optimization problems [33], wireless adhoc networks, stereo image processing [34], real time systems [35] etc.

<table>
<thead>
<tr>
<th>Year</th>
<th>Author</th>
<th>Name of algorithm</th>
<th>Problem constrained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996</td>
<td>Stutzle , Hoos</td>
<td>Max-Min Ant System (MMAS)</td>
<td>TSP and QAP</td>
</tr>
<tr>
<td>1997</td>
<td>Bullnheimer, Hartl and Strauss</td>
<td>Rank Based Ant System(AS rank)</td>
<td>TSP</td>
</tr>
<tr>
<td>1998</td>
<td>Stutzle</td>
<td>Parallelization of Ant system</td>
<td>Combinatorial optimization</td>
</tr>
<tr>
<td>1999</td>
<td>Cordon et. Al</td>
<td>Best Worst Ant System</td>
<td>Different instances of TSP</td>
</tr>
<tr>
<td>2007</td>
<td>Chunhui Zhao Bing Qi</td>
<td>Multiple Ant Colony Optimization</td>
<td>Resource allocation in network sessions</td>
</tr>
<tr>
<td>2008</td>
<td>Z Hu, J Zhang and Y Li</td>
<td>Continuous Orthogonal Ant Colony</td>
<td>Continuous optimizing problems</td>
</tr>
<tr>
<td>2009</td>
<td>Bin Y, Zhang-Zhen and Baozhen Y</td>
<td>Improved Ant Colony Optimization</td>
<td>Vehicle routing</td>
</tr>
<tr>
<td>2010</td>
<td>Ku Ruhana and Ku Mahamud Alaa Aljanaby</td>
<td>Interacted Multiple Ant Colony Optimization</td>
<td>Different instances of TSP</td>
</tr>
<tr>
<td>2012</td>
<td>Gupta DK, Arora Y et. Al</td>
<td>Recursive Multiple Ant Colony Optimization</td>
<td>Estimation of parameters of a function</td>
</tr>
</tbody>
</table>

Table 3: Evolution of ACO Algorithm

<table>
<thead>
<tr>
<th>Year</th>
<th>Author</th>
<th>Name of algorithm</th>
<th>Problem constrained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1993</td>
<td>C. M. Fonseca and P. J. Fleming</td>
<td>Multi-Objective Genetic Algorithm (MOGA)</td>
<td>Optimization</td>
</tr>
<tr>
<td>1994</td>
<td>J. Horn et al</td>
<td>Niched Pareto Genetic Algorithm</td>
<td>Multi-objective optimization</td>
</tr>
<tr>
<td>2002</td>
<td>Gehring, H. and Bortfeldt, A.</td>
<td>Parallel Genetic Algorithm</td>
<td>Container loading problem</td>
</tr>
<tr>
<td>2002</td>
<td>Deb, K et al</td>
<td>Non-Dominated Sorting Genetic Algorithms II (NSGA-II)</td>
<td>Constrained optimization problem</td>
</tr>
<tr>
<td>2002</td>
<td>Hartmann, S.</td>
<td>Self- Adapting Genetic Algorithm</td>
<td>Project scheduling under constrained resources</td>
</tr>
<tr>
<td>2002</td>
<td>Drezner, Zvi</td>
<td>New Genetic Algorithm</td>
<td>Quadratic Assignment Problem</td>
</tr>
<tr>
<td>2004</td>
<td>Feltl et al</td>
<td>Improved Hybrid Genetic Algorithm</td>
<td>Generalized Assignment Problem</td>
</tr>
<tr>
<td>2007</td>
<td>Dong Hwa Kim et al</td>
<td>Hybrid Genetic Algorithm</td>
<td>Global optimization</td>
</tr>
</tbody>
</table>

Table 4: Evolution of Genetic Algorithm
3. SUMMARY
All the algorithms discussed in the above section are optimization problems. These are compared with each other and every algorithm has its own advantages. Genetic Algorithm is much more popular because of its parallel computation. One more advantage of Genetic Algorithm is that it may handle both continuous and discrete variable without any gradient information where as all other techniques may give best performance for continuous problems but need slight modifications to handle discrete variables [36]. The variations used are shown in the Table 5.

4. CONCLUSION
In this critical review, we examined various optimization techniques by a comparative analysis based on evolution, methodology, performance and applications. In this paper, we find that these algorithms may be applied in various domains whether using as a direct approach or as any modified version. In future, new improved algorithms can be found for different scope areas.

5. REFERENCES


