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 life 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.

Table 1: Evolution of ABC Algorithm

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

Performance: PSO is a population based algorithm based on the cooperation of each particle. The convergence ability of PSO is faster than of 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.

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 metahueristic 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

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

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 3: Evolution of ACO Algorithm

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

Table 4: Evolution of Genetic Algorithm

Year	Author	Name of algorithm	Problem constrained
1993	C. M. Fonseca and P. J. Fleming	Multi-Objective Genetic Algorithm (MOGA)	Optimization
1994	J. Horn et al	Niched Pareto Genetic Algorithm	Multi-objective optimization
1995	N. Srinivas and K. Deb	Non-Dominated Sorting Genetic Algorithms (NSGA)	Multi-objective optimization
2002	Gehring, H. and Bortfeldt, A.	Parallel Genetic Algorithm	Container loading problem
2002	Deb, K et al	Non-Dominated Sorting Genetic Algorithms II (NSGA-II)	Constrained optimization problem
2002	Hartmann, S.	Self- Adapting Genetic Algorithm	Project scheduling under constrained resources
2002	Drezner, Zvi	New Genetic Algorithm	Quadratic Assignment Problem
2004	Feltl et al	Improved Hybrid Genetic Algorithm	Generalized Assignment Problem
2007	Dong Hwa Kim et al	Hybrid Genetic Algorithm	Global optimization

2011	Manuel Chica et al	Advanced Multi-Objective Genetic Algorithm	time and space assembly line balancing problem
2012	José Fernando Gonçalves, Mauricio G.C. Resende	Parallel Multi Population Biased Random-Key Genetic Algorithm	Container loading problem
2014	Camila Silva de Magalhães et al	Dynamic Nitching Genetic Algorithm	docking highly flexible ligands

Table 5: Modifications for Discrete Variables

Optimization Technique	**Continuous Variable**	**Discrete Variable**	**Discrete Variable with condition**	**Condition**
ABC	✓	x	✓	Random Key encoding Sigmoid Function CABC(Combinatorial ABC) DABC(Discrete ABC)
PSO	✓	x	✓	Sigmoid Function Smallest Position Value Great Value Priority Crossover Operator
ACO	✓	x	✓	Lagrangian Relaxation Lagrangian Decomposition Simulated Annealing Cutting Edge Algorithm based on polyhedral combinations
GA	✓	✓	x	Not Required

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

[1] K.K.Aggarwal & Yogesh Singh, “Software Engineering Programs Documentation, Operating Procedures,” New Age International Publishers, Revised Second Edition – 2005.

[2] Chen, Jun, and Mahdi Mahfouf. "Artificial immune systems as a bio-inspired optimization technique and its engineering applications." Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies: Applying Complex Adaptive Technologies (2009): 22.

[3] Philip T Cox and Baoming Song, “A formal Model for Component-Based Software”, IEEE Computer Society, Document number 07695-474-4/01, 2001, pp.304-310.

[4] Karaboga, Dervis. An idea based on honey bee swarm for numerical optimization. Vol. 200. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

[5] Alaya, I., C. Solnon, and K. Ghedira. "Ant Colony Optimization for Multi-Objective Optimization

Problems." In Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on, vol. 1, pp. 450-457. IEEE, 2007.

[6] Kulkarni, Nandakishore J., K. Venkat Naveen, Puneet Singh, and Praveen Ranjan Srivastava. "Test Case Optimization Using Artificial Bee Colony Algorithm." Advances in Computing and Communications (2011): 570-579.

[7] Karaboga, Dervis, and Bahriye Akay. "A comparative study of artificial bee colony algorithm." Applied Mathematics and Computation 214, no. 1 (2009): 108-132.

[8] Zhu, Guopu, and Sam Kwong. "Gbest-guided artificial bee colony algorithm for numerical function optimization." Applied Mathematics and Computation 217, no. 7 (2010): 3166-3173.

[9] Karaboga, Dervis, and Beyza Gorkemli. "A combinatorial artificial bee colony algorithm for traveling salesman problem." In Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on, pp. 50-53. IEEE, 2011.

[10] Li, Jun-Qing, Quan-Ke Pan, and Kai-Zhou Gao. "Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems."The International Journal of Advanced Manufacturing Technology 55, no. 9-12 (2011): 1159-1169.

[11] Li, Junqing, Quanke Pan, and Shengxian Xie. "Flexible job shop scheduling problems by a hybrid artificial bee colony algorithm." In Evolutionary Computation (CEC), 2011 IEEE Congress on, pp. 78-83. IEEE, 2011.

[12] Karaboga, Dervis, and Celal Ozturk. "Fuzzy clustering with artificial bee colony algorithm." Scientific research and Essays 5, no. 14 (2010): 1899-1902.

[13] Akay, Bahriye, and Dervis Karaboga. "Solving integer programming problems by using artificial bee colony algorithm." In AI* IA 2009: Emergent Perspectives in Artificial Intelligence, pp. 355-364. Springer Berlin Heidelberg, 2009.

[14] Jiao, Jian, Shan Yao, and Chunehe Xia. "Application for artificial bee colony algorithm in migration of mobile agent." In Advanced Intelligent Computing Theories and Applications, pp. 232-238. Springer Berlin Heidelberg, 2010.

[15] C. Vargas Ben´ıtez and H. Lopes. Parallel artificial bee colony algorithm approaches for protein structure prediction using the 3dhp-sc model. Intelligent Distributed Computing IV, pages 255-264, 2010.

[16] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, volume 4, pages 1942–1948 vol.4. IEEE Press, 1995.

[17] Windisch, Andreas, Stefan Wappler, and Joachim Wegener. "Applying particle swarm optimization to software testing." In Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 1121-1128. ACM, 2007.

[18] Hemlata S Urade and Prof. Rahila Patel. Article: Study and Analysis of Particle Swarm Optimization: A Review. IJCA Proceedings on 2nd National Conference on Information and Communication Technology NCICT(4):1-5, November 2011.

[19] van den Bergh, Frans, and A. P. Engelbrecht. "A new locally convergent particle swarm optimizer." In Proceedings of the IEEE international conference on systems, man, and cybernetics, vol. 7, pp. 6-9. 2002.

[20] Parsopoulos, Konstantinos E., and Michael N. Vrahatis. "Particle swarm optimization method in multiobjective problems." In Proceedings of the 2002 ACM symposium on Applied computing, pp. 603-607. ACM, 2002.

[21] Ahmet, Bestoun, and Kamal Zamli. "A Greedy Particle Swarm Optimization Strategy for T-way Software Testing." Journal of Artificial Intelligence 5.2 (2012): 85-90.

[22] Laskari, E. C., K. E. Parsopoulos, and M. N. Vrahatis. "Particle swarm optimization for minimax problems." In Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, vol. 2, pp. 1576-1581. IEEE, 2002.

[23] Dorigo, Marco, and Luca Maria Gambardella. "Ant colony system: a cooperative learning approach to the traveling salesman problem." Evolutionary Computation, IEEE Transactions on 1, no. 1 (1997): 53-66.

[24] Bullnheimer B, Hartl RF, Strauß C ” A new rank based version of the ant system - a computational study.” Central European Journal for Operations Research and Economics 7(1997): 25–38.

[25] Dorigo M, Di Caro G,” The ant colony optimization meta-heuristic”, McGraw-Hill Ltd., UK, Maidenhead, UK, England, (1999) pp 11–32.

[26] Hong Liu, Ping Li and Yu Wen, “Parallel Ant colony optimization algorithm,” in Proc. WCICA, 2006, pp 3222-3226.

[27] Maniezzo, Vittorio, and Alberto Colorni. "The ant system applied to the quadratic assignment problem." Knowledge and Data Engineering, IEEE Transactions on 11.5 (1999): 769-778.

[28] Colorni, Alberto, et al. "Ant system for job-shop scheduling." Belgian Journal of Operations Research, Statistics and Computer Science 34.1 (1994): 39-53.

[29] R. Schoonderwoerd, O. Holland, J. Bruten et L. Rothkrantz, Ant-based load balancing in telecommunication networks, Adaptive Behaviour, volume 5, numéro 2, pages 169-207, 1997.

[30] Bullnheimer, Bernd, Richard F. Hartl, and Christine Strauss. "An improved ant System algorithm for thevehicle Routing Problem." Annals of operations research 89 (1999): 319-328.

[31] Holland, J.H., “Adaptation in natural and artificial systems”, The university of Michigan press, 1975.

[32] Davis, L. (1991): Handbook of Genetic Algorithms. Van Nostrand Reinhold. New York, NY.

[33] Zhang, J., Lo, W.L., and Chung, H., "Pseudocoevolutionary Genetic Algorithms for Power Electronic Circuits Optimization", IEEE Trans Systems, Man, and Cybernetics, Part C., Vol.36, No.4, July 2006, pp. 590–598.

[34] Minglun G. and Yee-Hong Y., "Multi-resolution Stereo Matching using Genetic Algorithm", Stereo and Multi-Baseline Vision, 2001. (SMBV 2001). Proceedings. IEEE Workshop on, pp 21 –29, 2001.

[35] Madureira, A.; Ramos, C.; do Carmo Silva, S.,"A Coordination Mechanism for Real World Scheduling Problems using Genetic algorithms", Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002Congress on, 1, pp 175 –180, 2002.

[36] Krause, Jonas, Jelson Cordeiro, Rafael Stubs Parpinelli, and Heitor Silverio Lopes. "A survey of swarm algorithms applied to discrete optimization problems." Swarm Intelligence and Bio-inspired Computation: Theory and Applications. Elsevier Science & Technology Books (2013): 169-191.