Multi-objective Optimization for Object-oriented Testing Using Stage-Based Genetic Algorithm

P. Maragathavalli and S. Kanmani

Department of Information Technology,
Pondicherry Engineering College, Puducherry, India
{marapriya,kanmani}@pec.edu

Abstract. A multi-objective optimization involves optimizing a number of objectives simultaneously. The Multi-Objective Optimization Problem has a set of solutions, each of which satisfies the objectives at an acceptable level. An optimization algorithm named SBGA (stage-based genetic algorithm), with new GA operators is attempted. The multiple objectives considered for optimization are maximum path coverage with minimum execution time and test-suite minimization. The coverage and the no. of test cases generated using SBGA are experimented with simple object-oriented programs. The data flow testing of OOPs in terms of path coverage are resulted with almost 88%. Thus, the efficiency of generated testcases has been improved in terms of path coverage with minimum execution time.

Keywords: multi-objective optimization, test-suite minimization, stage-based, path coverage, execution time.

1 Introduction

Genetic algorithms which are more advanced heuristic search techniques have been successfully applied in the area of software testing. For a large search space and getting optimal set of solutions GA is the best choice in software testing. Commonly, these techniques are referred as evolutionary testing. Evolutionary testing tries to improve the effectiveness and efficiency of the testing process by transforming testing objectives into search problems, and applying evolutionary computation in order to solve them. In this testing of software can be done with a single objective or with multiple objectives. Instead of fixing only one criteria or quality parameter for generating test cases, multiple objectives like minimizing the time & cost and no. of test cases simultaneously maximizing the coverage (i.e., the test requirements) would be considered.

2 Existing Methods

Genetic Algorithms are the most popular heuristic technique to solve Multi-Objective Optimization Problems [2]. A Multi-Objective Optimization Problem has a number of

V.V. Das and J. Stephen (Eds.): CNC 2012, LNICST 108, pp. 246–249, 2012.

objective functions, which are to be minimized or maximized. MOOP can be expressed as, fitness function $f_m(x)$, where m = 1, 2, 3… M no. of objective functions & x = single candidate solution. The methods [1] used for multi-objective optimization are ranking, diversity and combined approach, NRGA [3].

3 Stage-Based Genetic Algorithm (SBGA)

The design diagram of stage-based genetic algorithm in connection with multi-objective optimization is shown in Fig. 1.

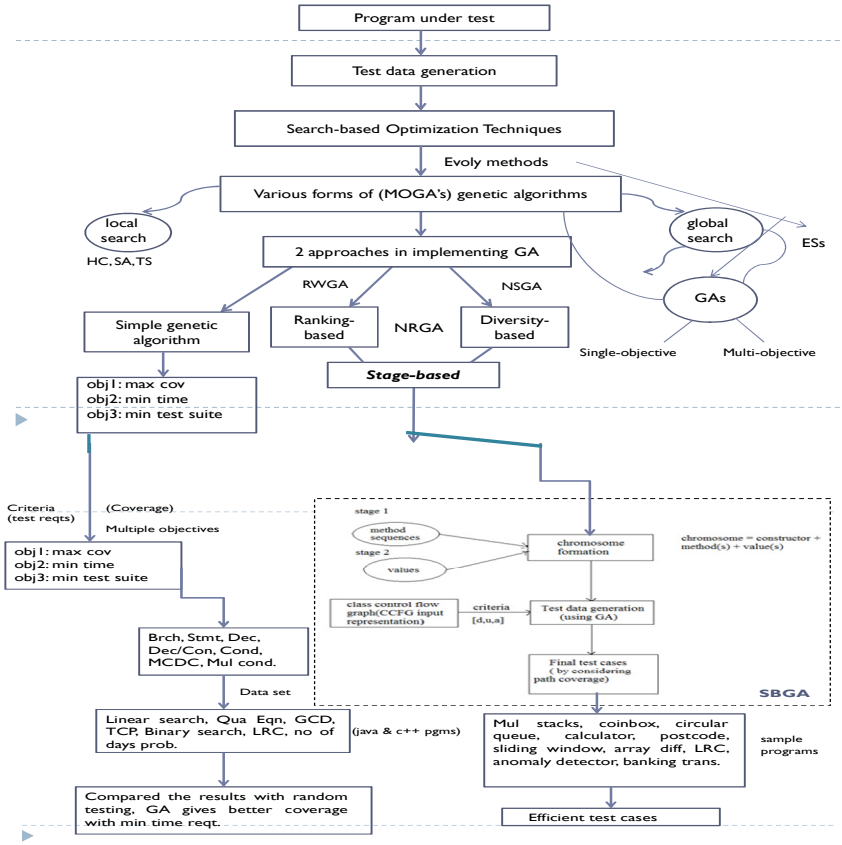


Fig.1. Interconnection of multi-objective optimization with stage-based GA

3.1 Test Data Generation

The parameters and operators considered for test data generation in GA, their initial and range of values are tabulated in table1 and the results are shown in table 2.

Table 1. GA parameters and operators

Name of the Parameter	Initial value	Range
Population size, M	40	40-200
Maximum generations per era, G	40	40-100
Number of eras, E	1	1-5
Crossover Probability, Cp	0.95	0.1-1.0
Mutation Probability, Mp	0.95	0.1-1.0

Name of the Operator	Type / value
Selection	Steady-state / fitness ≈ 1.0
Crossover	uniform / offsprings
Mutation	similarity / offsprings
Fitness function (*ft*)*	Float / (0.5 – 1.0)
Immigration rate (*I*)*	Float / (0.3 – 0.9)

Table 2. Results obtained for sample java programs

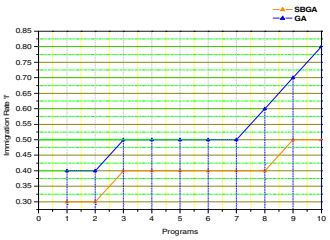
S. No	Sample programs	No. of test cases generated	Path coverage (cov/tot)		Execution time (ms)		Immigration rate (I)		Stopping criteria
			GA	SBGA	GA	SBGA	GA	SBGA	
1.	Multiple Stacks	100	0.90	0.95	2990	2180	0.5	0.4	Tc = 4(reduced)
2.	Coinbox	120	0.75	0.83	4080	3802	0.7	0.5	Upto 3 eras
3.	Circular Queue	105	0.75	0.83	3210	3095	0.5	0.4	Till fitness=1.0
4.	Calculator	110	0.87	0.93	2725	2570	0.4	0.3	Upto 3 eras
5.	Postal code	98	0.78	0.88	5180	4215	0.8	0.5	Upto 10 secs
6.	Sliding Window	100	0.87	0.93	4250	3990	0.5	0.4	Upto 3 eras
7.	Line-rectangle classifier	125	0.80	0.90	3917	3605	0.5	0.4	Tc = 4(reduced)
8.	Anomaly detector	140	0.80	0.90	2312	2196	0.5	0.4	Upto 4 eras
9.	Array difference	150	0.83	0.92	2105	1958	0.4	0.3	Upto 4 eras
10.	Banking transactions	105	0.80	0.87	3635	3220	0.6	0.4	Tc = 4(reduced)

* Fitness function $ft(v_i) = ½(cov\_d / tot\_d) + ½(cov\_u / tot\_u)$

* I = no. of chromosomes in next generation / no. of chromosomes in current generation The crossover used for getting the offsprings is uniform crossover which uses a fixed mixing ratio between two parents. For example, if the mixing ratio is 0.5, then half of the genes from both the parents go to offspring. The mutation is done using similarity mutation in which a similar gene is replaced with a different testcase.

3.2 Results and Discussion

Programs like multiple stacks, calculator, sliding window, and array difference gives more coverage; because, the number of conditions to be checked for selecting test cases are less whereas in coinbox and postal code type of programs conditions are more almost 40, shown in Fig. 2.



(4080, 5180) ms

(3802, 4215) ms

Fig. 2. Immigration rate results in SBGA and GA

4 Conclusion

Thus, the stage-based genetic algorithm with two stages is used for generation of object-oriented test cases. The fitness is purely depends on the path coverage of the test cases in the class. The results for sample java programs show that the efficiency and effectiveness of test cases generated by SBGA in terms of path coverage. In addition to path coverage, the time required for execution and the immigration rate are also satisfactory. This algorithm can be used for similar type of software engineering problems.

References

1. Ghiduk, A.S.: Automatic Generation of Object-Oriented Tests with a Multistage-Based Genetic Algorithm. Journal of computers 5(10), 1560–1569 (2010)
2. Singh, D.P., Khare, A.: Different Aspects of Evolutionary Algorithms, Multi-Objective Optimization Algorithms and Application Domain. International Journal of Advanced Networking and Applications 2(04), 770–775 (2011)
3. Konak, A., Coit, D.W., Smith, A.E.: Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering and System Safety, 992–1007 (2006)