IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 1, Ver. I (Jan.-Feb. 2018), PP 01-08
www.iosrjournals.org

Optimization Algorithm for Cost-Aware Test Suite Minimization in Software Testing

FaizBaothman
Department of Computer Science, College of Computers and Information TechnologyTaif University, Taif, Saudi Arabia
f.baothman@tu.edu.sa

Abstract: *Regression testing value is optimized by reducing subsets of check cases from a check suite while notcompromising the check demand. Researchers have given varied test-suite reduction techniquesmistreatment coverage metrics and greedy search algorithms. Besides greedy algorithms, optimizationbasedalgorithms have contend a significant role in check suite reduction. consequently, we tend todeveloped a brand new optimisation rule, rule to handle the variety drawback in generating newsolutions whereas finding the optimum check cases. Here, a fitness operate is developed to pick the checkcases optimally through the rule mistreatment 2 constraints, satisfying the complete check demand andminimizing the value live. The planned rule is experimented with 5 programs from SIR mistreatment fourcompletely different analysis metrics. The empirical study on the performance of the rule is analyzed withvaried parameters and therefore the comparison is completed with the greedy–based rule and thereforethe pulsation Genetic Search (SGS) rule. The experimental outcome showed that the planned rule*
outperformed the present rule in reaching the marginal value necessities
Keywords**:** *BAT algorithm, Optimization, Test case, Test suite, Greedy, Objective based, minimization.*

I. Introduction

The software applications in Banking, Medicaland Commercial Applications are subjected toextremely intricate verification and validationprocedures which involve some different tasks[1]. One of the validation procedure which helpsin improving the quality of the software issoftware testing and is the most important methodwhich guarantees the quality of the developingsoftware. Recently, Regression testing is the mostoften used maintenance process hichrevalidates the modified software. As the size ofthe test suite grows, the cost of regression testingincreases. It happens because as the software ismodified, the new test cases are added to testchanged requirements. Test-suite size problem isaddressed by two approaches namely test-suitereduction and test selection. Test-suite reductionis also known as test set minimization algorithms[3] which identify the minimized test suite thatprovides the same coverage of the software as theoriginal test-suite. In test-suite selection, a subsetof the test suite that will execute code or entitychanges is selected by the test selectionalgorithms..EIrreplaceability is a recent metricthat enables decrementing the number of testcases through greedy search algorithm [2].

The approaches presented in the literature fortest suite reduction are classified into four majortypes, i) Measure based test suite reduction, ii)
Greedy search-based test suite reduction, iii)

Optimization-Search based test suite reductionand iv) Multi-Objective-based test suitereduction. In measure based test suite reduction,coverage-based variants are widely applied aslike [2, 3, 17, 18] for test suite reduction. Thegreedy search based techniques are utilized thedifferent criteria and constraints to find theoptimal test suite as like [6, 15]. In optimizationbased testing, genetic algorithm,PSO algorithmis widely applied for test suite reduction. Thegenetic algorithm-based test suite reduction canbe found in [5, 19, 23, 24, 25].In this work, we bring an optimization algorithmcalled, PBAT (Poly BAT) algorithm to select testcases optimally with the constraint that test suiteshould satisfy all the test requirements.. At first,initial solutions are generated randomly with theconstraint that selected test cases in each andevery solution should satisfy the entire testrequirement. Then, fitness is evaluated using thetotal cost which is the aggregated execution timeof all the selected test cases. The solution setwhich has the minimum aggregated cost measureis then selected as the best solution set with thehelp of the proposed PBAT algorithm,

II. Related Works

EIrreplaceability metric is incorporated with theexisting test case metric Ratio using the wellknowntest suite reduction algorithms, such asGreedy, GRE, and HGS. This method attains alow cost test reduction strategy to yield a highlevel of test coverage. Reetika Nagar *et al.*proposed hybrid Particle Swarm Optimization (PSO) algorithm for test suite reduction [26]. Thismethod is effective in choosing the minimum setof test cases that possess the possibility of thefaults and bugs for which it takes minimum time.Martín Pedemontea*et al.* [28] proposed aSystolic Genetic Search (SGS) algorithm to solvethe real-world problem like the Test SuiteMinimization Problem (TSMP) existing in thefield of software engineering. This algorithmserves as a best method and it is highly effectivemethod for the TSMP with the xcellent scalablebehavior. Irreplaceability and EIrreplaceabilityfor test suite reduction were introduced. Thereduction of the test suite is done with thesemetrics and greedy search algorithm which is oneof the popular algorithms for the search processThe objective of this research is to develop aneffective test suite reduction approach forregression testing using an optimizationalgorithm called BAT algorithm [27]. Thisalgorithm aims to overcome the challengesdiscussed above and reduce the test suiteoptimally without compromising the testrequirements.

3-1 Representation Of Test Pool

A test case is a set of instructions which processinput variables required by the software toproduce desired results and test case requirementis a specific software function, loop or branchHere, the test case requirement is branchcoverage and the output provided specifieswhether the given test case covers the specificbranch or not. Let us assume that the number oftest case for the algorithm is d and number of testrequirement is m . Then, test pool can berepresented as, $P = \{cij;\ 0< i <d\ ;\ 0 < j < m\}$. cijmay be zero or one based on the requirementsatisfied by the test cases. Every value in Psignifies whether the corresponding test case cansatisfy the corresponding test requirement. Thecost value of each test case ci is computed byfinding the execution time of the test case. So, the
cost vector for all the test cases can be indicatedas,

$C_{T}= \{ y_i;\ 0< i <d \}$

3-2 Test Suite Reduction

The test suite reduction is carried out in thispaper that satisfies two constraints, such as, i)satisfying all test requirements, ii) Minimizingthe cost value. Let PR be the selected test suiteand x be the number of test cases removed. Then,the test suite reduction problem with costminimization is formulated as the following
objective

$PR = \{ckj;\ 0< k <d-x\ ;\ 0 < j < m\}$

$$\text{i)}\quad S = m \quad ;$$

$$\text{ii)}\quad S = \sum_{j=1}^{m} z_j \quad \text{where,} \quad z_j = 1 \ \text{if} \ \sum_{k=1}^{d-x} c_{kj} > 0$$

$$\text{iii)}\quad Min\left(\sum_{k=1}^{d-x} y_k\right)$$

3-3 Pbat Search Algorithm

3-3 Pbat Search Algorithm

Initialization: Let us assume that n bats arerandomly initialized their positions within thesearch space as, $bp=b$ $p1,bp2,\ ,bpq$where $p=1,2,...,n$
and$q$ is the dimension of the solution whichsignifies the number of test cases taken foroptimization. The variables such as, loudness A,pulse rate r, iteration t,minimum frequency,maximum frequency Qmax and velocity Vi
$t$areinitialized.

Evaluation: Every bat is then evaluated withfitness function and the best one having minimumfitness is stored as, xb

$$Q_i = Q_{\min} + (Q_{\max} - Q_{\min})^* \gamma$$
$$v_i^t = v_i^{t-1} + Q_i\left(x_i^{t-1} - x_b\right)$$
$$x_i^t = x_i^{t-1} + v_i^t$$

Where, is a random value which is used to updatethe frequency of the bat using *Q*min and *Q*max . Itranges between -1 to 1.The frequency *Qi* is thenutilized to update the velocity of the bats (*vit*)using the best position of the bats *xb*.Accordingly, the above equation of velocity canbe written as, *vit = vit*- 1 *α *Qi (xit - xb +β(U- xb*Qi))*

$$y_i^t = x_i^{t-1} + v_i^t$$
$$x_i^t = \begin{pmatrix} 1 & ; & y_i^t < 0.5 \\ 0 & ; & y_i^t < 0.5 \end{pmatrix}$$

Figure 1 means that the test cases selectedthrough this solution encoding procedure is 1, 3,4 and 6. In PBAT algorithm, bat population isrepresented as, *bpq*: 0 *p n* ; 0 *q d* . Here, *n* is thenumber of bats considered and *d* is the dimensionof the solution or number of test caseS

b_p						
1	0	1	1	0	1	0

Fig1 representation of BAT algorithm

Fitness evaluation: The fitness of every bat(solution) is evaluated using the fitness function,*F*(*bp*) . This function computes the total cost ofthe selected test cases through the solution *bp*onlyif the selected test cases can satisfy the entire testrequirement.

$$F(b_p) = \begin{cases} \sum_{q=1}^{d} b_{pq} * y_q & ; \ if \quad S = m \\ \infty & ; \ else \end{cases}$$
$$S = \sum_{j=1}^{m} z_j$$
$$z_j = 1 \quad ; \quad if \quad \sum_{k=1}^{d-x} c_{kj} > 0$$

```
Algorithm: PBAT
Input:     P → Test pool
           CT → Cost Factor
Output:
           Xb → best solution (Selected test cases)
Begin
     Initialize variables such as A, r, t, Qmin, Qmax, ∝, β
     Initialize p = 1, bat population bp velocity Vi^t
      While p < t
           Find fitness for bp using P and CT
           Update velocity Vi^t and frequency Qi
           Update bats position by xi^t
           Store best solution xb
           If (γ > r)
           Generate local solution around best solution
          Endif
          If (γ < A)
          Generate random solution, xr
Find fitness of xr using P and CT
```

```
If(fitness(xr) < fitness(xb)
    Update xb , r and A
Endif
Endif
P = p+1
Endwhile
Return xb
End
```

PBAT Search Algorithm for Test Suite Reduction

PBAT algorithm: The input for the PBATalgorithm is test pool P and cost vector CT. Table1 shows the inputs test pool P and cost vector CTof the PBAT algorithm. The variables areinitialized as, *n* =5, *t* =2; *A* =0.5, *r* =0.5,*Q*min =0;*Q* max =1; *d* =7. Table 3 shows the initialization ofx, Q and v. In the first iteration, for □=-0.35,∝=0.8, □=0.2, frequency, velocity andposition values are updated which is shown inTable 5. Then, fitness is computed for everysolution of x: Fitness(x(1))=∞ ; Fitness(x(2))=∞ ; Fitness(x(3))=∞ ; Fitness(x(4))=∞ ; Fitness(x(5))=∞ . We obtained no solutionswhich satisfies the entire test requirement. So,again, x (1) is taken as best solution and is givenin Table 6 at the end of first iteration. In thesecond iteration, for□ =0.35, □=0.8, □=0.2,frequency, velocity and position values areupdated as shown in Table 7. Then, fitness iscomputed for new solution of x values:
Fitness(x(1))=∞ ; Fitness(x(2))=47;
Fitness(x(3))=∞ ; Fitness(x(4))=∞ ;
Fitness(x(5))=∞ . At the end of second iteration,the minimum fitness is obtained for x (2). So, weselected x (2) as the best solution, shown in Table8. After finishing two iterations, the selected testcases through best solution are (c1j, c2j, c4j, c5j,and c7j) and requirements solved are (ci1, c i2,ci3, ci4, ci5, ci6, ci7). The total cost required is47 (1+2+11+23+10) which is obtained by doingthe summation of all the cost values of selectedtest cases.

Table: 2 Test Pool P and Cost Vector CT

	C_{i1}	C_{i2}	C_{i3}	C_{i4}	C_{i5}	C_{i6}	C_{i7}	Cost(C_T
C_{1j}	1	1	0	0	0	0	0	1
C_{2j}	0	1	1	0	0	0	0	2
C_{3j}	0	0	1	1	0	0	0	5
C_{4j}	0	0	0	1	1	0	0	11
C_{5j}	0	0	0	0	1	1	0	23
C_{6j}	1	0	0	0	0	1	0	40
C_{7j}	0	0	0	0	0	0	1	10

Table : 3 Initialization of X , Q and V

x	0	1	0	0	1	0	1
	1	0	0	1	0	0	0
	0	1	0	0	1	0	0
	1	0	0	1	0	1	0
	0	0	1	0	1	0	0

Q	1	1	1	1	1	1	1
v	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0

Table 4 Best Solution From Initialization

X_0	0	1	0	0	1	0	1

Table 5 Updated value of x,y,Q and v after Iteration 1

v	0.2	0.27	0.2	0.2	0.27	0.2	0.27
	-0.01	0.48	0.2	-0.01	0.48	0.2	0.48
	0.2	0.27	0.2	0.2	0.27	0.2	0.48
	-0.01	0.48	0.2	-0.01	0.48	0.2	0.48
	0.2	0.48	-0.01	0.2	0.27	-0.01	0.48
Q	-0.35	-0.35	0.35		0.35		0.35
y	0.2	1.27	0.2	0.2	1.27	0.2	1.27
	0.99	0.48	0.2	0.99	0.48	0.2	0.48
	0.2	1.27	0.2	0.2	1.27	0.2	0.48
	0.99	0.48	0.2	0.99	0.48	0.99	0.48
	0.2	0.48	0.99	0.2	1.27	0.2	0.48
x	0	1	0	0	1	0	1
	1	0	0	1	0	0	0
	0	1	0	0	1	0	0
	1	0	0	1	0	1	0
	0	0	1	0	1	0	0

Table 6 Best solution after Iteration 1

xb	0	1	0	0	1	0	1

Table 7 Updated value of x,y,Q and v after Iteration 2

v	0.4	0.54	0.4	0.4	0.54	0.4	0.54
	-0.02	0.96	0.4	-0.02	0.96	0.4	0.96
	0.4	0.54	0.4	0.4	0.54	0.4	0.96
	-0.02	0.96	0.4	-0.02	0.96	-0.02	0.96
	0.4	0.96	-0.02	0.4	0.54	0.4	0.96
Q	-0.35	-0.35	-0.35	-0.35	-0.35	-0.35	-0.35
y	0.4	1.54	0.4	0.4	1.54	0.4	1.54
	0.98	0.96	0.4	0.98	0.96	0.4	0.96
	0.4	1.54	0.4	0.4	1.54	0.4	0.96
	0.98	0.96	0.4	0.98	0.96	0.4	0.96
	0.4	0.96	0.98	0.4	1.54	0.98	0.96
x	0	1	0	0	1	1	1
	1	1	0	1	1	1	1
	0	1	0	0	1	1	1
	1	1	0	1	1	0	1
	1	1	1	0	1	1	1

Table 8 Best solution after Iteration

xb	1	1	0	1	0	1	0

III. Results And Analysis

5-1 Experimental Setup

The proposed PBAT algorithm is implementedusing Java 1.7 with NetBeans IDE 7.3. Theexperimentation is conducted on Windows 7machines with Intel Core Duo processors and 2GB of memory. At first, required no of test casesare generated randomly through a syntheticprogram. Once we generate test cases for asubject program through synthetic program,branch coverage and cost is computed byapplying test case to the corresponding subjectprogram.

5-2 Performance Evaluation

Figure 3. presents the SCR graph for variousvalues of minimum frequency. When theminimum frequency is increased from 0 to 0.4,the value of the median program is found todecrease from 85.33 to 71.81.

The value of thePBAT using SCR for the elevator programs is90.37, 92.69, 93.28, 94.64, and 92.99 when theminimum frequency value increases as 0, 0.1, 0.2,0.3, and 0.4 respectively. Likewise, the value ofthe trityp programs using the proposed TBAT andSCR is 9.66, 78.09, 87.22, 10.94, and 90.46respectively with the increasing minimumfrequency value from 0 to 0.4. Figure 4.shows theperformance of PBAT using SCR for variousvalues of maximum frequency. The maximumfrequency is varied as 0.6, 0.7, 0.8, 0.9, and 1 forall the programs like the median, elevator, trityp,Apollo, and pool3. The value of the medianprogram using the TBAT with the SCR reaches84.34 from 91.31, elevator programs reaches93.36 from 99.94, trityp attains 85.58 from 99.93,Apollo program attains 40.33 from 99.99, andpool3 attains 99.69 from 99.99. Figure 5. Showsthe performance of TBAT using SCR for variousvalues of loudness. The value of loudness usedfor analysis is 0.2, 0.4, 0.6, 0.8, and 1respectively. The SCR value of the medianprogram when the loudness is 0.2 is 93.45, 76.84for 0.4, 48.66 for 0.6, 27.27 for 0.8, and 70.33for 1 as loudness

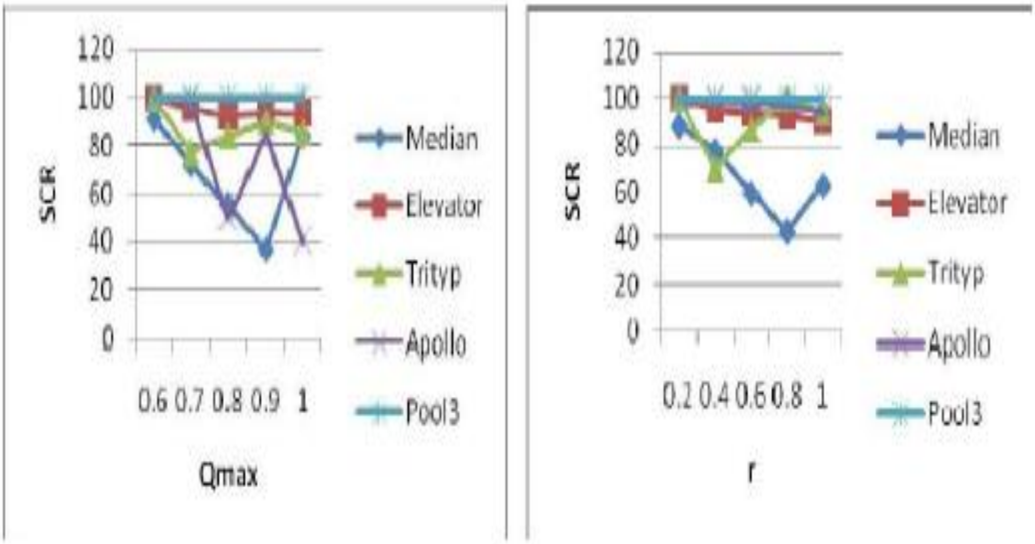

Figure 3 Performance of PBAT using SCR, a) for various numbers of bats b) for variousminimum frequency

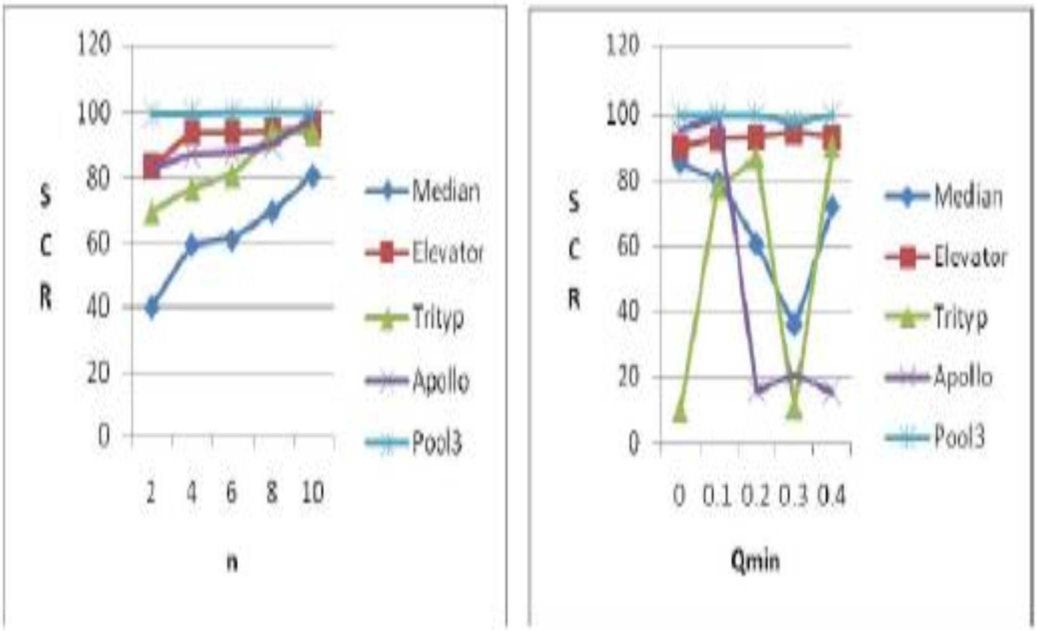

Figure 4 Performance of PBAT using SCR, a) for various maximum frequency b) for variouspulse rate

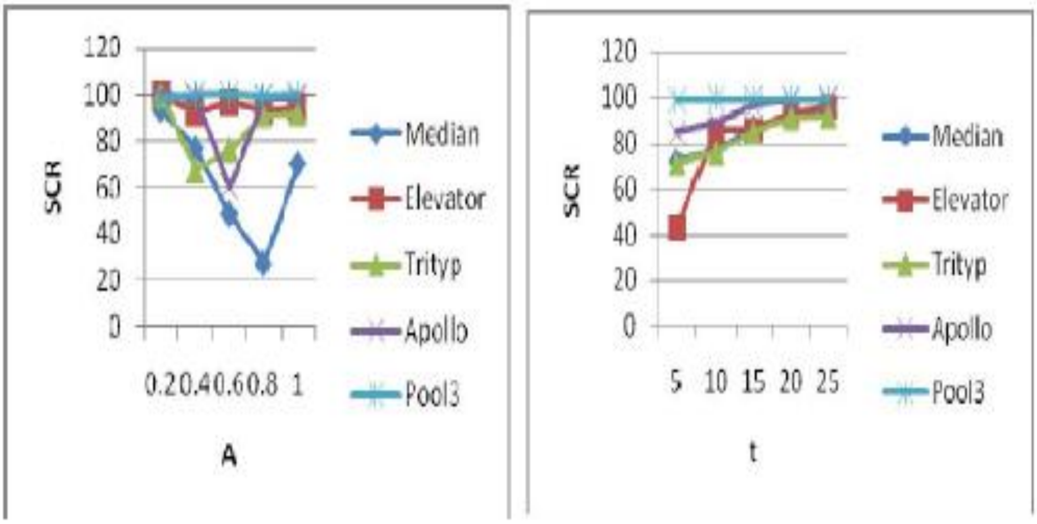

Figure 5 Performance of PBAT using SCR, a) for various loudness b) for various iterations

5-3 Analysis

Test pool is directly given to the algorithms,PBAT, Systolic Genetic Search [28] and GreedyEIrreplaceability. The ultimate aim of thesealgorithms is to select test cases which shouldsatisfy all the test requirements. Accordingly,the test suite is reduced by both the algorithmsand the cost for all the selected test cases arecomputed and shown in Table 9. The total costfor the proposed PBAT algorithm is 30.32 msecfor median program as compared to the value of66.7 for the existing algorithm. PBAT achieved89.2% improvement in the variance of 1% ascompared with the existing algorithm whichimproves only 76.2% in the variance of 5%. Theproposed PBAT obtained SCR values of 93.7%,99.44%, and 99.5% for trityp, apollo and pool3programs

Program	Original	PBAT Cost	Greedy/Eirreplaceability Cost	Systolic Genetic cost	PBAT-SCR	Greedy/ Eirreplaceability-SCR	Systolic Genetic-SCR
Median	280.826	30.32± 3	66716 7	35.1 45	89.2 1	76.24 5	875.0 1
Elevator	16215± 6.3	10141± .3 210	15088.77 ± 256	12568 ± 451	93.74 ± 1	90.69± 0.7	92.24 ± 2
trityp	61145.± 42	3822.9 ± 400	5721.18 ±745	4122 ± 45	90.64	93.62	93.25 ± 3
Apollo	726283 ±3	40525. ±5 550	649642.95 ± 452	41563 .2 ± 545	99.4 ±4 55	91.55± 01	99.42 ± 01
Pool 3	1.17E+1	5.8 E+0.8	375181608± 5689	6.1 E+08 ± 345	99.50 ±025	99.68 ± 03	99.47 ± 01

Table 9 Reduction capability of algorithms (in msec) for input threshold of 0.75

IV. Conclusion

PBAT algorithm is to minimize the cost ofregression testing. This algorithm wasdeveloped to handle the diversity problem ingenerating new movements of bats to reach theoptimal solution easily. The minimizationfunction developed here to improve the speedof convergence contains two constraints,satisfying the entire test requirement andminimizing cost measure. In the proposedPBAT algorithm, initial solutions are generatedrandomly and fitness is evaluated using theproposed minimization function. Thegeneration of the new solution set is done bythe proposed formula to reach the minimumcost function faster than greedy-basedalgorithms. The performance of the proposedPBAT algorithm is extensively analyzed withdifferent parametric values to understand thebest parameters of the proposed algorithm intest suite reduction.

References

[1]. J.Campos, R.Abreu, "Encoding Test Requirements as Constraints for Test Suite Minimization", in Proceedings of 10th International Conference on Information Technology: New Generations, 2013. [2] C-T. Lin, K-W. Tang, G.M. Kapfhammer, "Test Suite reduction methods that decrease regression testing costs by identifying

[2]. irreplaceable tests", Information and Software Technology, vol. 56, pp. 1322– 1344, 2014.

[3]. J.A. Jones and M.J. Harrold, Test-Suite Reduction and Prioritization for Modified Condition/Decesion Coverage. IEEE Trans. On Software Engineering, Vol. 29, no. 3, pp. 195-209, Mar. 2003.

[4]. G. Rothermel and M.J. Harrold, A Safe, Efficient Regression Test Selection Technique. ACM Trans. Software Eng. And Methods, vol. 6, no. 2, pp. 173-210, Apr. 1997.

[5]. X-y.Ma, B-k. Sheng, and C-G. Ye, "Test- Suite Reduction Using Genetic Algorithm", LNCS 3756, pp. 253–262, 2005.

[6]. S. Sampath, R. Bryce, and A. Memon, "A uniform representation of hybrid criteria for regression testing," of tware Engineering, IEEE Transactions on, vol. 39, no. 10, pp. 1326–1344, 2013

[7]. Z. Li, M. Harman, and R. M. Hierons, "Search algorithms for regression test case prioritization," IEEE Transactions on Software Engineering, vol. 33, no. 4, pp. 225–237, 2007.

[8]. T. Y. Chen, F.-C. Kuo, R. G. Merkel, and T. H. Tse, "Adaptive Random testing: The art of test case diversity," J. Syst. Softw., vol. 83, no. 1, pp. 60–66, Jan. 2010.

[9]. Software-artifact Infrastructure Repository (SIR), http://sir.unl.edu/content/sir.php. [10] Y.Wang, R.Gao, Z.Chen, W. E.Wong, B.Luo, "WAS: A weighted attribute-based strategy for cluster test selection", Journal of

[10]. Systems and Software, Vol. 98, pp. 44–58, December 2014.

[11]. G.Fraser, A.Arcuri, P.McMinn, "A Memetic Algorithm for whole test suite generation ", Journal of Systems and Software, Vol. 103, pp. 311–327, May 2015.

[12]. S.Sampath, R.C. Bryce, "Improving the effectiveness of test suite reduction for usersession- based testing of web applications", Information and Software Technology, Vol. 54, no. 7, pp. 724–738, July 2012.

[13]. J-W. Lin, C-Y. Huang, "Analysis of test suite reduction with enhanced tie-breaking techniques", Information and Software Technology, Vol. 51, no. 4, pp. 679–690, April 2009.

[14]. I.Rodriguez, L.Llana, P.Rabanal, "A General Testability Theory: Classes, Properties, Complexity, and Testing Reductions", IEEE Transactions on Software Engineering, Vol. 40, no. 9, pp. 862 - 894, June 2014.

[15]. M. Cohen, M. Dwyer, and J. Shi, "Constructing interaction test suites for highlyconfigurable systems in the presence of constraints: A greedy approach," Software Engineering, IEEE Transactions on, vol. 34, pp. 633–650, 2008.

[16]. H. Hemmati, A Arcuri, and L. Briand, "Achieving Scalable Model-Based Testing Through Test Case iversity", ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 22, no. 1, February 2013.

[17]. E.Shaccour, F.Zaraket, and W.Masri, "Coverage Specification for Test Case Intent Preservation in Regression Suites", in Proceedings of IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops, 2013.

[18]. J.A. Jones, M.J. Harrold, "Test-suite reduction and prioritization for modified condition/decision coverage", IEEE Transaction on Software Engineering, vol. 29, no. 3, pp. 195– 200, 2003.

[19]. X.Y. Ma, Z.F. He, B.K. Sheng, C.Q. Ye, "A genetic algorithm for test-suite reduction", in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, IEEE, pp. 133–139, October 2005.

[20]. A.M. Smith, G.M. Kapfhammer, "An empirical study of incorporating cost into test suite reduction and prioritization", in: Proceedings of the 24th ACM Symposium on Applied Computing, Software Engineering Track, ACM, pp. 461–467, March 2009.

[21]. A.Panichella, R.Oliveto, M.D. Penta, A.D. Lucia,"Improving Multi-Objective Test Case Selection by Injecting Diversity in Genetic Algorithms", IEEE Transactions on Software Engineering, Vol. 41, no. 4, pp. 358 - 383, 2015.

[22]. S. Yoo, M. Harman, "Regression testing minimization, selection and prioritization: a survey", Journal Software Testing, Verification & Reliability, Vol. 22, no. 2, pp. 67-120, March 2012.

[23]. S. Selvakumar, M.R.C. Dinesh, C. Dhineshkumar, and N.Ramaraj, "Reducing the Size of the Test Suite by Genetic Algorithm and Concept Analysis", CCIS 90, pp. 153–161, 2010.

[24]. W.Zhang, B.Wei, and H.Du, "Test Case Prioritization Based on Genetic Algorithm and Test-Points Coverage", LNCS 8630, pp. 644– 654, 2014

[25]. S.Wang, S.Ali, A.Gotlieb, "Cost-effective test suite minimization in product lines using search techniques", The Journal of Systems and Software, pp. 1–22, 2014.

[26]. R.Nagar, A.Kumar, S.Kumar, A.S.Baghel, "Implementing Test Case Selection and Reduction Techniques using Meta- Heuristics", in Proceedings of 2014 5th International Conference -Confluence The Next Generation Information Technology Summit (Confluence), pp. 837-842, 2014.

[27]. ShounakRushikeshSugave,MIT College of Engineering, Pune SuhasHaribhauPatil,BharatiVidyapeeth

[28]. University College of Engineering, Pune B Eswara Reddy(3) JNTUA College of Engineering, Kalikiri [27] X. S. Yang, A New Metaheuristic Bat- Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization

[29]. (NISCO 2010) (Eds. J. R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin, 284, Springer, pp. 65-74, 2010.

[30]. M.Pedemonte, F.Luna, E.Alba, "A Systolic Genetic Search for reducing the execution cost of regression testing", Applied Soft Computing, Vol. 49, pp. 1145–1161, December 2016

FaizBaothman. "Optimization Algorithm for Cost-Aware Test Suite Minimization in Software Testing." IOSR Journal of Computer Engineering (IOSR-JCE) , vol. 20, no. 1, 2018, pp. 01-08.