IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

Test Case Selection Using Multi-Objective Evolutionary Algorithms

S. Raheja[1], R. Singh[2*]

[1,2]Dept. C.S.E, Deenbandhu Chhotu Ram University of Science and Technology, Murthal, India

**Corresponding Author: rajvirsingh.cse@dcrustm.org , Tel.: 9811731389*

Available online at: www.ijcseonline.org

Accepted: 19/Jul/2018, Published: 31/Jul/2018

***Abstract*—** Regression testing is needed to ensure the correct behavior of software after change. For the process of automation and selection of test cases, a number of meta-heuristic techniques have been used in literature. In this paper, bat algorithm, cuckoo search and multi-objective binary genetic algorithms have been discussed. The proposed multi-objective binary genetic algorithm is evaluated against test functions and its performance is analyzed in comparison to existing algorithms i.e. bat and cuckoo search algorithm. For this, we have considered factors such as fault coverage and execution time. The related dataset is extracted from benchmark repository named flex object which originates from SIR. Results indicate that multi-objective performs better than bat and cuckoo search algorithm.

***Keywords*—** Software Testing, Regression Testing, Bat Algorithm, Cuckoo Search Algorithm, Software Maintenance

I. INTRODUCTION

Optimization is required in each domain of life. A large portion of genuine optimization issues are multi-objective. Relapse experiment determination is one such multi-target optimization issue that has pulled in the analysts in most recent years because of some reasons:

a. If we perform all test suites, it takes very much time which is unacceptable.

b. For testing, much man power is needed which is practically not feasible.

c. Above mentioned problems give rise to increase in time period, each and every time regression testing is performed.

More current meta-graphy calculations roused by nature are developing and they turn out to be progressively famous. For instance, PSO (particle swarm optimization) came into existence from the example of fish and winged creatures insight, whereas the Firefly algorithmic rule was motivated from glimmering example of fireflies [1], [2], [5], [10], [11]. These nature-motivated meta-graphy calculations have been utilized as part of an extensive variety of improvement issues, including NP-difficult issues e.g. the voyaging salesperson issue [1], [2], [5]. Energy of all advanced meta-graphy originates from the way that they copy the best element in nature, particularly organic frameworks developed from regular choice for more than millions of years. Two vital qualities are determination of the fittest and adjustment to nature. Numerically, these can be converted into two pivotal qualities of the cutting edge meta-graphy: increase and expansion [2]. Heightening helps us to analyze the present arrangements to choose the best, and whereas enhancement ensures the calculation can investigate the inquiry space effectively.

This paper intends to plan another calculation, called Cuckoo Search (CS), in light of the intriguing rearing behavior for example, brood parasitism of specific types of cuckoos. We will initially present the rearing behavior of cuckoos what's more, the attributes of L'evy flights of a few fowls and natural product flies, and afterward figure the new CS, trailed by its usage. At long last, we will think about the proposed look procedure with other prominent advancement calculations what's more, talk about our discoveries and their suggestions for different advancement issues.

The remaining paper is set in the following manner: Part B of the paper discusses about the study of related work made by the previous analysts. Test case selection problems with regression testing is discussed in Part C. Brief discussion about bat, cuckoo search algorithmic rule and proposed methodology is done in Part D, E and F. Practical aspects of design, setup and result analysis is discussed in Part G. Part H concludes the paper.

Background

REGRESSION TESTING

Out of different kinds of testing, Regression testing is one of them to check the instance. Changes in code performed, after this coding process changes are necessary in software enabling fixing the defects and for increase in existing functionality or add new functionality.

Already tested features are affected by changes made in system [21], but new faults should not be generated with such changes. As such regression testing is done.

Retest process is needed for maintaining the software after change. It should be assumed that the other components of software are not badly affected [20].

Regression Test case Selection

There were some redundancies contained in large test suites due to the fault covered in more than two test cases [19]. The same should be reduced, manually selection of test cases can take ample of time period and fault as search based optimization problem may cause hence meta-heuristic algorithmic rules are applied for problem rectification solutions [14]. The paper explains test adequacy criteria for fault coverage usage and time taken by test cases for purpose of maximizing fault coverage and time execution reduction.
Suppose,

Test suit be T = {T1, T2, …., Tm}
Fault set considered be F = {F1, F2, …., Fn}
Fault coverage of the test cases is given by the function F (Ti). So, fault coverage can be calculated as,

Fault coverage = 100 * $\bigcup_{t=1}^{r}\{E(Ti)\}$/p (1)

Here, r denotes number of selected cases whereas p denotes total number of faults

Time period of execution should be minimum. We can define execution time as complete time needed to check code [3].
The test cases execution time in total, which are selected is shown by –

$\sum_{t=1}^{s} ti$ (2)

For ith test cases, we are considering ti as the execution time. Rate of detection of faults can be reduced by test case selection as the effect of test suite, if minimized in shape may also be minimized.

Bat Algorithm

Bat Algorithm consolidates the great highlights of different kind of nature propelled meta-heuristics like HS (harmony search), FA (firefly algorithm) and SA (simulated annealing) [12]. Bat meta-heuristic algorithmic rule depends on the echo sounding property of miniaturized scale bats. This property controls the searching conduct of small scale bats and causes them to discover their prey and enables them to recognize distinctive kinds of bugs regardless of whether it is excessively dull [13].

Cuckoo Search Algorithm

Cuckoo Search algorithm additionally has a place with the Nature roused advancement Meta heuristics. It was presented by Young and Deb in 2009 and has ended up being exceptionally encouraging for explaining numerous hard real world optimization issues [14]. Forceful generation procedure is the primary fascination of the cuckoos. It is a sort of parasitism in which a cuckoo lays its eggs in the home of host species. Some cuckoo species lay their eggs in other fledgling's homes and may expel the have fledgling's eggs to expand the incubating likelihood of their own eggs [14]. Some host flying creatures don't care for interlopers and struggle with them. For this situation either the host winged creature will toss their eggs out or may essentially relinquish its home and fabricate another home at some other place.

II. RELATED WORK

Biswas et al (2012)

The paper explains the technique which is based on prioritizing and arranging the test cases in manner so that fewer lines of code ought to be processed and due to this process faults are detected rapidly. This particular process is been initiated for regression testing, however it will be helpful for order test to perform the best at initial stage of software testing. In alternate version change process is also performed that may be not so much effective afterwards.

Srivastav et al (2012)

An algorithm is suggested by this paper to generate test sequences on optimum basis and to obtain full software coverage based on cuckoo search. This paper explains the role of test sequence optimum theory, its importance and how it makes the task easier through cuckoo search. The proposed technique can also be enhanced through taboo search.

De souza et al (2013)

Another type of algorithm is PSO (Particle swarm optimization) quite powerful and simple when compared to different techniques. BCPSO has already produced better conclusions for the considered optimum tasks and also hybrid algorithmic rules produced better conclusions. The latest researches also give us the improved structure for obtaining better qualities of the selection process.

Yang et al (2013)

While analyzing the present system of bat algorithm, basic parameter control strategy has been inducted for improvement which will also provide theoretical understanding for meta-heuristic algorithms and various problems in the real world applications.

Tavakoli et al (2013)

The modified CS (cuckoo search) algorithm ensures the realistic figures and focuses on its own rate. Some engineering optimization problems gone through modified CS algorithm and results shown it's superiority over other similar methods. We prefer to go outside for search only for better solutions otherwise not.

Feng et al (2014)

The researcher explains about the paper that another improved encoding cuckoo search algorithm (ICS) has been introduced to solve 0-1knapsack problem. The Genetic

Mutation operator provokes the particular algorithm for fast, better focused results and leaves less space for local optimum theories. It presents the superiority of proposed algorithm in comparison with HS and CS but it may take time to solve knapsack problems which are increased with increment of size of instance.

Srivastav et al (2014)
To take view of test efforts being conducted, another model using meta-heuristic bat algorithm is also suggested in the paper. The model is proposed, keeping in mind the improved solutions, more accuracy than other methods and actual efforts are demonstrated. Comparative study of previous and proposed model and additional tests are also beneficial.

Wang et al (2014)
Hybrid form of CS (cuckoo search) algorithm has proved its effectiveness if some changes are applied and ISFLA (improved shuffled frog leaping algorithm) is implemented for solution of 0-1 knapsack problem. The model has shown its ability and effectiveness in generating better solutions of good quality they took over the BCS (binary cuckoo search), BDE(binary differential evolution) and GA(genetic algorithm).

Reddy et al (2015)
The proposed bat algorithm leads the Economic load dispatch (ELD) to solve problems to optimum level. This technique is easy to implement gives optimum results of solutions with less effort. It is also easy and it shows accuracy and efficiency over other algorithms. This algorithm containing valve point loading effect was applied to economic load dispatch (ELD) but with constraints. Review should also be taken about multiple fuels, spinning reserve, power flow, up & down rate constraints and hydro thermal scheduling problems by this ELD process.

Huang et al (2016)
An improved KH is presented which introduced the update method of Wn to the basic KH algorithm. The new KH algorithm is used for the numerical optimization problem. an improved KH is presented which introduced the update method of Wn to the basic KH algorithm. The new KH algorithm is used for the numerical optimization problem. It is tested on the ten test function and the performance is compared with the basic KH algorithm, PSO, DE and GA algorithm. There should be an improvement of the KH algorithm and the application for the other complex optimization problems.

Lingzhi et al (2017)
The paper explains about CS (cuckoo search) algorithm to get better optimum parameters of PID to control in induction motor drive system. These fixed parameters remaining constants can affect the efficiency of algorithm.

Yadav et al (2017)
The paper describes about Bat algorithm which shows efficient results for difficult problems and also shows various modifications of Bat Algorithm. In this, Spontaneous parameter tuning and controlling can be done further. Bat Algorithm itself is not similar to crossover operator in Genetic Algorithm and Differential Evolutions. Due to this, convergence performance for Bat Algorithm slows down. Bat Algorithm is used by various researchers as this algorithm is capable and attractive one.

III. METHODOLOGY

Multi-objective problems are caused by two or more objective functions. Special mathematic and algorithmic tools can solve these problems for optimization.

In real world, multi-objective problems are seen here and there. For the theory of optimality varies as foresaid solutions are tradeoffs or carrying better aspects over multi-objective. Here, theory of optimality is known as EW-Pareto optimality which is applied to generate tradeoff solutions stating about the solutions as MO (multi-objective) problem is Pareto optimal, where not any solution is possible to improve some criteria but not causing degradation at the same time in at least to any criterion. Through this concept we can get NDS (non-dominated solutions) called as Pareto optimal set.

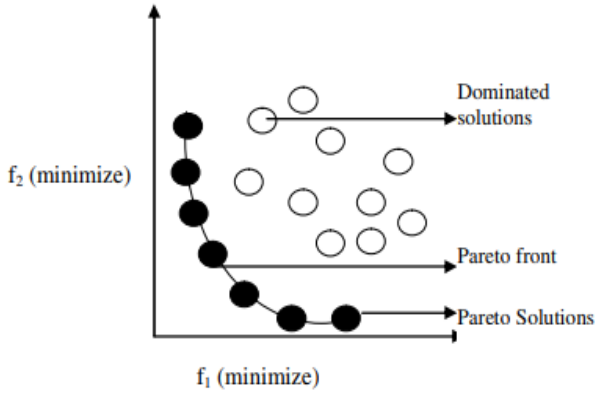

Figure1. Pareto optimality concept

Here, f1 is the objective function i.e. fault coverage and f2 is the another objective function i.e. test cases execution time. Here Fault coverage defines as proportion of some sort of faults which will be detected throughout the test of any designed system. And the Execution time of the task which is give is outlined because the time spent for system execution of that task as well as time spent for the execution of run time or system services on its behalf. In this case, we are considering both test cases selected and input execution time using dataset [2] flex v1, v2 and v3 in table 1.

Table1. This table contributes test cases selected and execution time of input for datasets

Flex v1		Flex v2		Flex v3	
Test cases selected	Execution time	Test cases selected	Execution time	Test cases selected	Execution time
174	0.2707	114	0.3322	165	0.4030
177	0.2862	129	0.3921	253	0.4189
189	0.3275	150	0.4163	473	0.4200
222	0.3591	204	0.4819	477	0.4062
236	0.5751	531	0.4443	519	0.4159

Block diagram of NSGA II
(1)The input given to the proposed methodology is as follows:

Fault Matrix
Fault matrix represents the 0 and 1 row and column wise. Here 0 represents the false. The 1 represents true. When such data is organized in two dimensional form then it is known as fault matrix. This would be input to the proposed model. Cuckoo Search, BAT Algorithm and Multi-objective binary genetic algorithm would process this input.
(2)The objective part for proposed algorithm i.e. MOGA is as follows:

We are considering two different objective functions:
f1 is the objective function i.e. fault coverage
f2 is the another objective function i.e. test cases execution time

Here, fault coverage defines as proportion of some sort of faults which will be detected throughout the test of any designed system.

And Execution time of the task which is give is outlined because the time spent for system execution of that task as well as time spent for the execution of run time or system services on its behalf.

Pareto Front
A set of non-dominated solutions, being chosen as optimal, if no objective can be improved without sacrificing at least one other objective. On the other hand a solution x* is referred to as dominated by another solution x if, and only if, x is equally good or better than x* with respect to all objectives. The plot of the objective functions whose non-dominated vectors are in the Pareto optimal set is called the Pareto front.
Optimum selected test cases

Test case selection techniques identify and eliminate the modification revealing test cases and try to reduce the test suite size for optimization of regression testing.
Through this output is proposed as follows:
The paper explains test adequacy criteria for fault coverage usage and time taken by test cases for purpose of maximizing fault coverage and time execution reduction.
Time complexity
The time complexity is the computational complexity that describes the amount of time it takes to run an algorithm. Time complexity is commonly estimated by counting the number of elementary operations performed by the algorithm, supposing that each elementary operation takes a fixed amount of time to perform. Thus, the amount of time taken and the number of elementary operations performed by the algorithm are taken to differ by at most a constant factor.

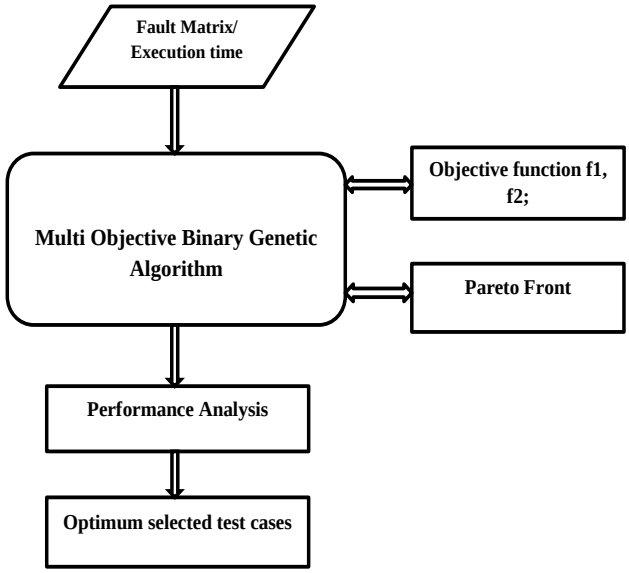

Figure2. Basic block diagram of proposed work

In Pareto optimal set, solutions are identified by the achievement of multi-objective optimization algorithm. For Pareto optimal set is not implemented in much cases of multi-objective difficulties faced. Hence, is the tool multi-objective optimization theory adopted for analyzing process of solution which represents the Pareto optimal set to the maximum extent [16].

With these above said points, multiple objective optimization approach can lead us to some conflicting results:-
(a)Very familiar Pareto front should have very similarities of then actual Pareto front.

(b)The solutions of most familiar Pareto set must have qualities of distribution and diversification in comparison to Pareto front in such a way that it may reflect real aspect of tradeoffs.

(c)The most familiar Pareto front should cover all the view of aspect of Pareto front. Solutions process should be ensured that they cover all the objective function space.
The paper provides general access to MOGA (multi-objective genetic algorithm) for achievement of above conflicting results while solving multi-objective problems.

EXPERIMENTAL DESIGN AND SETUP

The particular section discusses the practical experiments which are done for evaluation of cuckoo search, bat and multi-objective binary genetic algorithmic rule. This particular study is about different five versions of flex analyzer are particularly studied which can be known from SIR as the subject program. These are v1-v5.

EXPERIMENTAL DESIGN
Following research questions arises:
RQ1: Is the performance of each of the algorithms has any distinction for problem faced on regression test case selection?
RQ2: What should be the number of attempts needed to save time period in RTCS (regression test case selection)?

EXPERIMENTAL SETUP
Many experiments was carried out extensively, their results were studied in order to answer the research queries. Each algorithm was applied using MATLAB for thirty times on 3 versioned programs of flex to face the challenges. Every algorithm was applied 50 times at one time run as such 1500 function evaluations took place for each algorithm for every flex program version. Every algorithm took 100 as initial population but at the same was made limited to choose only five out of 567 test cases. The same were allotted numbers as T1-T567. Seeded faults were numerised as F1-F20. Table 2 tells about the quantity of test cases and faults selected for every flex program version.

Table2. Characteristics of Subject Program

Program Version	Total no. of test cases	Total no. of faults
Flex v1	567	19
Flex v2	567	20
Flex v3	567	17
Flex v4	567	16
Flex v5	567	9

IV. RESULTS AND DISCUSSION

Table 3 below shows the value of collected metrics which tells about program version i.e. which dataset we are using with fault coverage and execution time of existing and proposed methodology with their time complexity.

Program version	Exe time of bat	Exe time of cuckoo	Exe time of proposed	fault cov of bat	fault cov of cuckoo	fault cov of proposed	time comp of bat	time comp of cuckoo	time comp of proposed
flex v1	73.5421	71.9397	63.252	84.2105	84.2105	84.2105	7.8173	8.5622	2.1968
flex v2	69.1054	70.9901	63.7666	84.2105	84.2105	84.2105	7.5214	8.8231	2.2826
flex v3	68.2674	68.5286	63.7666	84.2105	84.2105	84.2105	7.6792	8.9891	2.3469

Figure 3 and 4 below displays the collected metrics in graphical form

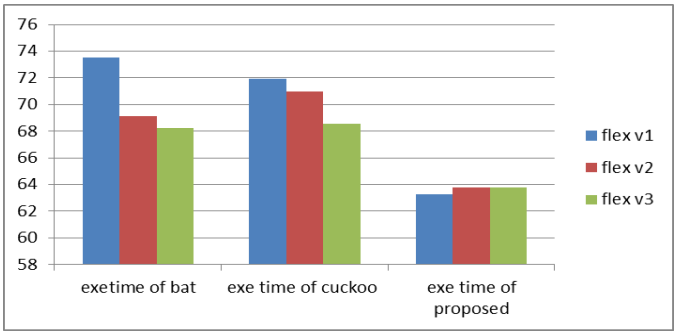

Figure 3 shows the execution time of bat, cuckoo search and proposed algorithm

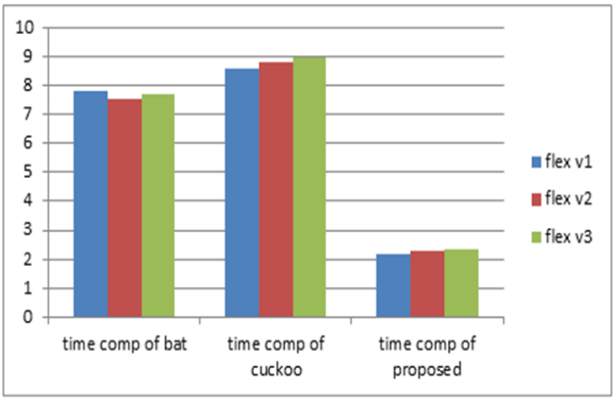

Figure 4 shows the time complexity of the above explained algorithms

We reiterate the fact only about the test cases taken into consideration, where running time of algorithm was lesser which show that it covered high quantity of faults in 30 runs. How much implementation time is needed for every test case is shown at the last in every table. We can say that time for implementation of proposed algorithm is much better in comparison to execute bat and cuckoo search algorithmic rule and it is cleared from the table 4, 5 and 6.

Table4. Test cases selected by Cuckoo search algorithm and their execution time

Cuckoo Search Algorithm					
Flex v1		Flex v2		Flex v3	
Test case	Exe time ti	Test case	Exe time ti	Test case	Exe time ti
T230	69.7324	T232	70.3863	T235	71.4619
T236	71.3669	T228	69.6301	T233	71.2239
T238	71.4144	T245	72.7074	T232	70.8910
T228	69.9209	T229	70.4866	T224	68.3374
T227	68.7849	T239	73.1069	T228	69.0131
$\sum t_i$	351.2195	$\sum t_i$	356.3173	$\sum t_i$	350.9273

Table5. Test cases selected by Bat algorithm and their execution time

Bat Algorithm					
Flex v1		Flex v2		Flex v3	
Test case	Exe time ti	Test case	Exe time ti	Test case	Exe time ti
T222	67.9214	T226	68.5040	T229	69.7980
T212	64.0999	T249	73.2926	T237	71.4983
T248	74.6205	T239	73.2092	T251	77.2678
T218	67.4885	T224	71.7442	T214	65.6625
T234	71.1177	T231	71.2844	T220	67.4358
$\sum t_i$	345.248	$\sum t_i$	358.0344	$\sum t_i$	351.6624

Table6. Test cases selected by proposed multi objective binary genetic algorithm and their execution time

Proposed Algorithm					
Flex v1		Flex v2		Flex v3	
Test case	Exe time ti	Test case	Exe time ti	Test case	Exe time ti
T208	63.7666	T214	64.9150	T203	61.9986
T208	63.7666	T214	64.9150	T203	61.9986
T208	63.7666	T214	64.9150	T203	61.9986
T208	63.7666	T214	64.9150	T203	61.9986
T208	63.7666	T214	64.9150	T203	61.9986
$\sum t_i$	318.833	$\sum t_i$	324.575	$\sum t_i$	309.993

V. CONCLUSION AND FUTURE SCOPE

The paper mainly contributes for investigation of performance of existing bat, cuckoo search algorithmic rule and proposed multi-objective binary genetic algorithm to select test cases taking into consideration the both execution time and fault coverage. It is also pointed out multi-objective binary genetic algorithm was firstly analyzed in the test case selection and it was tried to the best of our knowledge. Same process of algorithm can be done for different test case selection as these are not only for two functions. Results show multi-objective binary genetic algorithm are more efficient than bat and cuckoo search algorithmic rule in terms of time complexity with execution time and fault covered metric.

REFERENCES

[1] S Nachiyappan, A Vimala devi and C B Selva Lakshmi, *"An evolutionary algorithm for regression test suite reduction"*, In Communication and Computational Intelligence (INCOCCI), 2010 International Conference, pp. 503-508, 2010, December, IEEE.

[2] G Rothermel , S Elbaum, A Kinneer and H Do, 2006. Software-artifact infrastructure repository. URL http:// sir.unl. edu/portal.

[3] R.Y Nakamura, L.A Pereira, K A Costa, D Rodrigues, J.P Papa and X.S Yang, 2012,*"BBA: a binary bat algorithm for feature selection"*. In 2012, 25th SIBGRAPI Conference on Graphics Patterns and Images, pp. 291-297, 2012,August, IEEE.

[4] Zhaolu Guo, Xuezhi Yue, Kejun Zhang, Shenwen Wang and Zhijian Wu, *"A Thermodynamical Selection-Based Discrete Differential Evolution for the 0-1 Knapsack Problem"*, Entropy 2014, 16, 6263-6285; doi:10.3390/e16126263.

[5] Khan, K. and Sahai, A., 2012. A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context. International Journal of Intelligent Systems and Applications ,4(7), p.23.

[6] Songwei Huang, Lifang He, Xu Si, Yuanyuan Zhang and Pengyu Hao, An Effective Krill Herd Algorithm for Numerical Optimization, International Journal of Hybrid Information Technology Vol. 9, No.7 (2016), pp. 127-138 http://dx.doi.org/10.14257/ijhit.2016.9.7.13.

[7] Usha Badhera, G.N Purohit, Debarupa Biswas, 2012 Test case prioritization algorithm based upon modified code coverage regression testing, International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.6, November 2012.

[8] Luciano S. de Souza, Ricardo B.C. Prudêncio, Flavia de A. Barros, Eduardo H. da S. Aranha, 2013 Search based constrained test case selection using execution effort, Expert Systems with Applications 40 (2013) 4887–4896.

[9] Yanhong Feng, Ke Jia and Yichao He, An Improved Hybrid Encoding Cuckoo Search Algorithm for 0-1 Knapsack Problems, Hindawi Publishing Corporation Computational Intelligence and Neuroscience Volume 2014, Article ID 970456, 9 pages http://dx.doi.org/10.1155/2014/970456.

[10] Srivastava P.R, Sravya C, Ashima, Kamisetti S and Lakshmi M, 2012. Test sequence optimisation: an intelligent approach via cuckoo search. International Journal of Bio-Inspired Computation,4(3), pp.139-148.

[11] Nagar R, Kumar A, Singh G.P and Kumar S, 2015, February. Test case selection and prioritization using cuckoos search algorithm. In Futuristic Trends on Computational analysis and knowledge management(ABLAZE) 2015, International Conference on (pp. 283-288).IEEE.

[12] Yang, X.S., 2010. A new metaheuristic bat-inspired algorithm. In Nature Inspired Cooperative Stratergies for Optimization (NICSO 2010) (pp. 65-74). Springer Berlin Heidelberg.
[13] Biswal, S., Barisal, A.K., Behera, A. and Prakash, T., 2013, April. Optimal power dispatch using BAT algorithm.InEnergy Efficient Technologies for Sustainability (ICEETS) 2013, National Conference on(pp. 1018-1023). IEEE.
[14] Yang, X.S. and Deb, S., 2009, December. Cuckoo search via Lévy flights. In Nature and Biologically Inspired Computing, 2009 NaBIC 2009, World Congress on (pp. 210-214). IEEE.
[15] Deb, K., 2001, "Multi-Objective Optimization using Evolutionary Algorithms," Wiley Chichester, UK.
[16] Abdullah Konak, David W. Coit , Alice E. Smith, Multi-objective optimization using genetic algorithms: A tutorial, Reliability Engineering & System Safety Volume 91, Issue 9, September 2006, Pages 992-1007, Elsevier.
[17] P. Sudheer Kumar Reddy* P. Anil Kumar G.N.S. Vaibhav, Application of BAT Algorithm for Optimal Power Dispatch, International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163 Issue 2, Volume 2 (February 2015).
[18] Lingzhi Yi, Yue Liu, wenxin Yu, Genping Wang, Yongbo Sui, Adaptive Cuckoo Search Algorithm for the Speed Control System of Induction Motor, SCIREA Journal of Electrical Engineering http://www.scirea.org/journal/DEE February 21, 2017 Volume 2, Issue 1, February 2017.
[19] De Souza, L.S., Prudêncio, R.B. and Barros, F.D.A., MultiObjective Test Case Selection: A study of the influence of the Catfish effect on PSO based strategies.
[20] Ali, A., Nadeem, A., Iqbal, M.Z.Z. and Usman, M., 2007, December. Regression testing based on UML design models. In Dependable Computing, 2007. PRDC 2007. 13th Pacific Rim International Symposium on (pp. 85-88). IEEE.
[21] Gupta Nirmal Kumar and Rohil Mukesh Kumar "Improving GA based Automated Test Data Generation Technique for Object Oriented Software", IEEE International Advance Computing Conference (IACC), Ghaziabad, pp.249 – 253, 2013.
[22] Srivastava, P.R., Bidwai, A., Khan, A., Rathore, K., Sharma, R. and Yang, X.S., 2014. An empirical study of test effort estimation based on bat algorithm. International Journal of Bio-Inspired Computation,6(1), pp.57-70.
[23] Praveen Ranjan Srivastava, Chandolu Sravya, Ashima, Sai Kamisetti and Manogna Lakshmi, Test sequence optimisation: an intelligent approach via cuckoo search, Int. J. Bio-Inspired Computation, Vol. 4, No. 3, 2012.
[24] Ehsan Valian, Saeed Tavakoli, Shahram Mohanna, Atiyeh Haghi, Improved cuckoo search for reliability optimization problems, Computers & Industrial Engineering 64 (2013) 459–468, Elsevier.
[25] Xin-She Yang, Bat algorithm: literature review and applications, Int. J. Bio-Inspired Computation, Vol. 5, No. 3, pp. 141–149 (2013). DOI: 10.1504/IJBIC.2013.055093
[26] S. L. Yadav, M. Phogat, *"A Review on Bat Algorithm",* International Journal of Computer Sciences and Engineering, Volume-5, Issue-7, E-ISSN: 2347-2693.

Authors Profile

Rajvir Singh is an Assistant Professor in Computer Science & Engineering Department , Deenbandhu Chhotu Ram University of Science & Technology (DCRUST) (A State Govt. University), Murthal: 131039 (India) since 5th Feb., 2010 to till date. Before joining DCRUST he has served in Delhi College of Engineering / Delhi Technological University, Delhi: 110042 (India) from 19th July, 2007 to 5th Feb., 2010 as Contractual Lecturer (Computer Engineering). He has completed his B.E.(CSE) degree in 2003 from MDU, Rohtak, Haryana:124001(India) and M.Tech. in Computer Engineering in 2007 from YMCA Institute of Engineering / YMCA University of Science & Technology, Faridabad, Haryana, India. He is pursuing Ph.D.(CSE)(Part-Time) from DCRUST, Murthal:131039(India). He have qualified various national level tests like GATE-2003, UGC-NET-2009(June), UGC-NET-2012(June), Research Aptitude Test (RAT)-2010 conducted by Jamia Millia Islamia, New Delhi-110025(India). His areas of interest are Software Testing, Software Engineering, Data Base Management Systems, Computer Graphics, Theory of Automata Computation and Artificial Intelligence.

Sumedha Raheja is pursuing M.tech (Computer Science & Engineering) from Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Murthal. She has completed her B.tech (Information Technology) from BMIET, Sonipat, Haryana, India in 2016.

