iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45247

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

Comparative Analysis of Efficacious Metaheuristic Technique with Genetically Modified- Flower Pollination Algorithm (GM-FPA) for Test Case Prioritization in Regression Testing

Priyanka Dhareula[1], Anita Ganpati[2]

Abstract: ***Regression Testing is most imperative activity of software development life cycle. Test case prioritization being one of the most adopted branch for regression testing and with the invent of nature inspired metaheuristic techniques in optimization, this study makes an attempt to augments the features of test case prioritization with nature inspired metaheuristic techniques to determine the most efficacious metaheuristic techniques from Cuckoo Searh (CS) algorithm, Genetic Algorithm (GA) and Flower Pollination Algorithm (FPA) for three different case studies. APFD metrics is used to compare the algorithms. Further the study compares the most efficacious technique with Genetically Modified- Flower Pollination Algorithm (GM-FPA) to identify the most efficient technique for regression test case prioritization.***

Keywords: ***Regression Testing, Test Case Prioritization, Cuckoo Search Algorithm, Genetic Algorithm, Flower Pollination Algorithm, APFD.***

I. INTRODUCTION

The present study compared metaheuristic techniques for test case prioritization [4] in regression testing [1, 15]. Efficacy of an algorithm is determined by the maximum number of faults it can identify for a given version of a case study. The most efficacious metaheuristic technique for test case prioritization has been identified by comparing Cuckoo Search (CS) algorithm[13,14], Genetic Algorithm (GA) [3,10], and Flower Pollination Algorithm (FPA) [2, 12] for test case prioritization by using the Average Percentage of Faults Detected (APFD) metrics.

Further, the efficiency of the proposed Genetically Modified-Flower Pollination Algorithm (GM-FPA) [11] is measured in terms of APFD value by comparing it with the most efficacious metaheuristic technique identified in this study.

Also, GM-FPA is compared with random order, and reverse random order of test case execution for test case prioritization in regression testing.

This study uses three case studies to perform the comparative analysis of the metaheuristic techniques. The research culminates with the detailed discussion of the results produced during the course of this study.

II. RESEARCH METHODOLORY

Empirical study was performed to analyze the most efficacious metaheuristic technique for test case prioritization. To determine the efficiency of the proposed technique, it was compared with the previously identified most efficacious metaheuristic technique, and two non-metaheuristic techniques namely: random order and reverse random order of test case execution for test case prioritization. All the techniques were implemented in java.

For the empirical evaluation three case studies namely: Puzzle Game Application (PGA), Triangle Classification Problem (TCP), and AreaandPerimeter Application (APA) designed in java were used. The case studies were picked from different online code repositories. Various open source software namely: Eclipse IDE, EclEmma code coverage tool, TestNG tool, MySql were considered to code the algorithms, to maintain the database, to analyze the code coverage, fault coverage, and time of execution of the test cases respectively.

Three metrics were used for the empirical evaluation of this study namely: Average Percentage of Statement Coverage (APSC), Average Percentage of Fault Coverage (APFD), and time of execution.

©IJRASET: All Rights are Reserved

To achieve the objectives of the study three case studies considered from different open source software are shown in Table 1.

Table 1: Case Studies

Sr. No.	Case Studies	Size in LOC	No. of Versions/ Unique Faults	No. of Test Cases Used Per Version	Source
1.	Puzzle Game Application (PGA)	246	5	33	Github [5]
2.	Triangle Classification Problem (TCP)	106	6	45	SIR [6]
3.	AreaandPerimeter Application (APA)	916	8	113	Stack Overflow [7]

All the three metaheuristic algorithms namely CS, GA, FPA have been implemented for test case prioritization in the prior studies [8,9,11]. Also GM-FPA have been proposed and implemented in [11]. This study tries to determine the most efficient technique by comparing the efficacious technique with GM-FPA in the next sections.

III. EFFICACIOUS TECHNIQUE: COMPARATIVE ANALYSIS OF CS, GA, AND FPA

In this section most efficacious metaheuristic technique for test case prioritization is determined. To determine the efficacious technique GA, CS, and FPA have been compared on the basis of APFD metrics.The analysis is done by comparing the APFD values of all the techniques for test case prioritization on three case studies having nineteen version in total. The word efficacious here symbolizes desirable results in terms of maximum fault coverage in reduced time of execution. Table 2 summarizes the APFD results and time of execution of CS, reverse order for CS algorithm, GA, reverse order for GA, FPA, and reverse order for FPA respectively for test case prioritization in regression testing.

Table 2: Efficacious Metaheuristic Technique For Test Case Prioritization

Sr. No.	Technique	PGA		TCP		APA	
		APFD Results	Time (in sec)	APFD Results	Time (in sec)	APFD Results	Time (in sec)
1.	CS	0.91213	0.352	0.71489	0.072	0.58851	0.103
2.	Reverse Order CS	0.85758	0.281	0.33334	0.072	0.58851	0.103
3.	GA	0.85152	0.313	0.58888	0.054	0.54979	0.098
4.	Reverse Order GA	0.88789	0.313	0.18149	0.054	0.43364	0.098
5.	FPA	0.85152	0.268	0.67038	0.048	0.44691	0.061
6.	Reverse Order FPA	0.8099	0.268	0.397	0.048	0.50222	0.061

As it is evident from the APFD results given in Table 2 that, CS algorithm outperforms GA and FPA by having highest value for APFA metrics for test case prioritization in regression testing. Therefore, it is further stated that CS algorithm is the most efficacious metaheuristic technique out of GA and FPA for test case prioritization in this study.

IV. RESULTS AND DISCUSSION

This section measures the efficiency of the proposed metaheuristic technique termed as Genetically Modified-Flower Pollination Algorithm (GM-FPA) [11], by comparing it with the efficacious metaheuristic techniques i.e. CS algorithm identified in this study. In this section GM-FPA is also measure with the traditional approaches for test case prioritization i.e., the random order and reverse random order of test case execution. The TestOrderRandomizer class is used from the TestNG tool to run the original test suites randomly for all the case studies. Figure 1 shows the random order of execution of original test suite for PGA in 1.548 seconds.

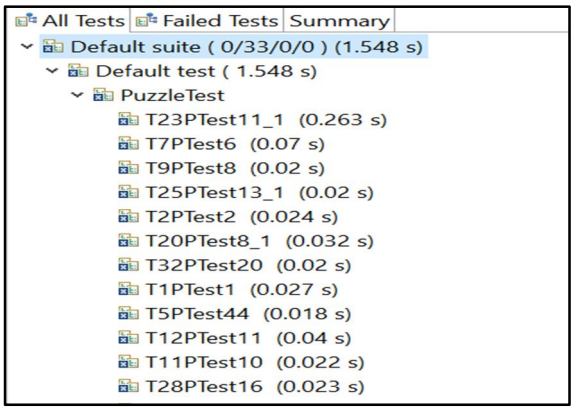


Figure 1: Random Ordering And Time Of Execution For PGA

APFD results of random order of execution for PGA is computed below:

$$APFD = 1 - \frac{19 + 19 + 27 + 15 + 29}{33 * 5} + \frac{1}{2 * 33}$$

$$APFD = 1 - \frac{99}{165} + \frac{1}{66}$$

$$APFD = 1 - 0.6 + 0.01515$$

$$APFD = 1 - 0.61515 = 0.38485$$

Figure 2 shows the random order of execution of original test suite for TCP in 0.282 seconds.

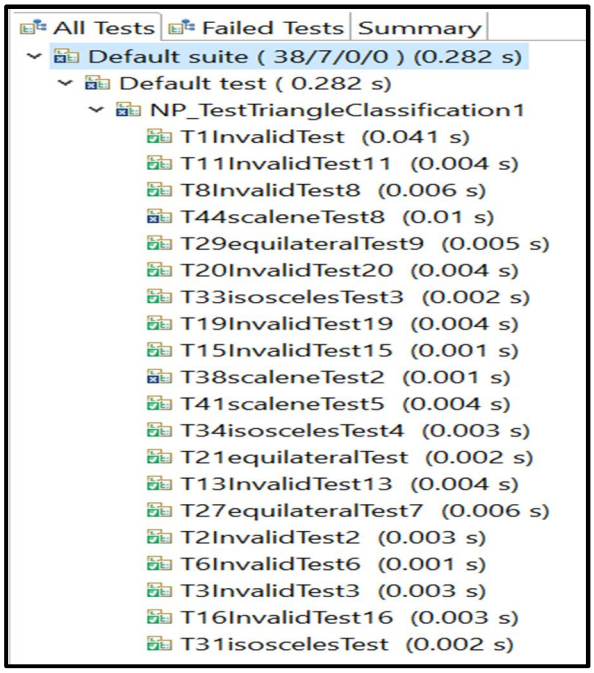


Figure 2: Random Ordering And Time Of Execution For TCP

©IJRASET: All Rights are Reserved

APFD results of random order of execution for TCP is computed below:

$$APFD = 1 - \frac{5 + 43 + 33 + 40 + 45 + 37}{45 * 6} + \frac{1}{2 * 45}$$

$$APFD = 1 - \frac{203}{270} + \frac{1}{90}$$

$$APFD = 1 - 0.75185 + 0.01111$$

$$APFD = 1 - 0.76296 = 0.23704$$

Figure 3 shows the random order of execution of original test suite TS_o for APA in 0.7 seconds. APFD results of random order of execution for APA is computed below:

$$APFD = 1 - \frac{108 + 108 + 80 + 35 + 7 + 68 + 41 + 70}{113 * 8} + \frac{1}{2 * 113}$$

$$APFD = 1 - \frac{517}{904} + \frac{1}{226}$$

$$APFD = 1 - 0.57190 + 0.00442$$

$$APFD = 1 - 0.57632 = 0.42368$$

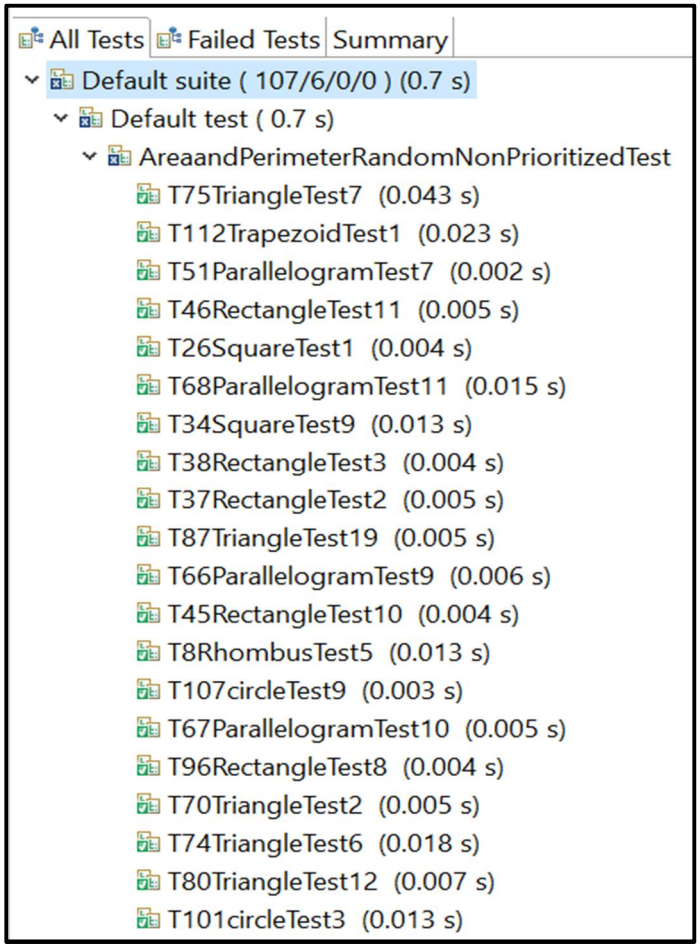


Figure 3: Random Ordering And Time Of Execution For APA

To determine the efficiency of the proposed technique GM-FPA, Table 3 summarizes the APFD results and time of execution for GM-FPA (proposed technique) [11], CS algorithm (efficacious technique), random order of execution, and reverse random order of execution for PGA, TCP, and APA.

Table 3: Comparison of GM-FPA With CS Algorithm And Traditional Ways of Test Case Prioritization

Sr. No.	Algorithm Used	PGA		TCP		APA	
		APFD Results	Time (in sec)	APFD Results	Time (in sec)	APFD Results	Time (in sec)
1.	GM-FPA	0.92429	0.434	0.77038	0.04	0.59514	0.095
2.	Reverse Order GM-FPA	0.91213	0.434	0.41852	0.04	0.49116	0.095
3.	CS	0.91213	0.352	0.71489	0.072	0.58851	0.103
4.	Reverse Order CS	0.85758	0.281	0.33334	0.072	0.58851	0.103
5.	Random order of execution	0.38485	1.548	0.23704	0.282	0.42368	0.7
6.	Reverse random order of execution	0.27576	1.548	0.35186	0.282	0.43363	0.7

As it is evident from Table 3 that GM-FPA has outperformed the most efficacious technique for test case prioritization i.e., the CS algorithm and also the traditional ways of test case prioritization i.e., the random order and reverse random order of test case execution for test case prioritization in regression testing.

It is further stated that by modifying FPA the rate of convergence has improved, since the APFD value of GM-FPA for all the case studies is much higher as compared to APFD values of other metaheuristic techniques for test case prioritization as reflected in Table 4. The study states that GM-FPA performs a good balance between exploitation and exploration that was lacking in original FPA [12]. It is empirically proved in this study, that GM-FPA resulted in better rate of convergence by delibrating higher APFD value as compared to simple FPA.

Table 4: Summarized APFD Results And Time of Execution For All The Metaheuristic Techniques And Traditional Ways For Test Case Prioritization

Sr. No.	Algorithm Used	PGA		TCP		APA	
		APFD Results	Time (in sec)	APFD Results	Time (in sec)	APFD Results	Time (in sec)
1.	GM-FPA	0.92429	0.434	0.77038	0.04	0.59514	0.095
2.	Reverse Order GM-FPA	0.91213	0.434	0.41852	0.04	0.49116	0.095
3.	CS	0.91213	0.352	0.71489	0.072	0.58851	0.103
4.	Reverse Order CS	0.85758	0.281	0.33334	0.072	0.58851	0.103
5.	GA	0.85152	0.313	0.58888	0.054	0.54979	0.098
6.	Reverse Order GA	0.88789	0.313	0.18149	0.054	0.43364	0.098
7.	FPA	0.85152	0.268	0.67038	0.048	0.44691	0.061
8.	Reverse Order FPA	0.8099	0.268	0.397	0.048	0.50222	0.061
9.	Random order of execution	0.38485	1.548	0.23704	0.282	0.42368	0.7
10.	Reverse random order of execution	0.27576	1.548	0.35186	0.282	0.43363	0.7

V. CONCLUSION

The study performed a comparative analysis of CS, GA, FPA to identify the most efficacious technique in terms of maximum fault coverage in less time. The APFD results conveys the efficacious behaviour of CS algorithm for test case prioritization for three case studies used in this research. Further the study identifies efficient algorithm for test case prioritization by comparing the APFD values of most efficacious technique identified in this study i.e., Cuckoo Search Algorithm with Genetically Modified- Flower Pollination Algorithm. It was evident from APFD values for all the three case studies that GM-FPA outperformed Cuckoo Search algorithm.

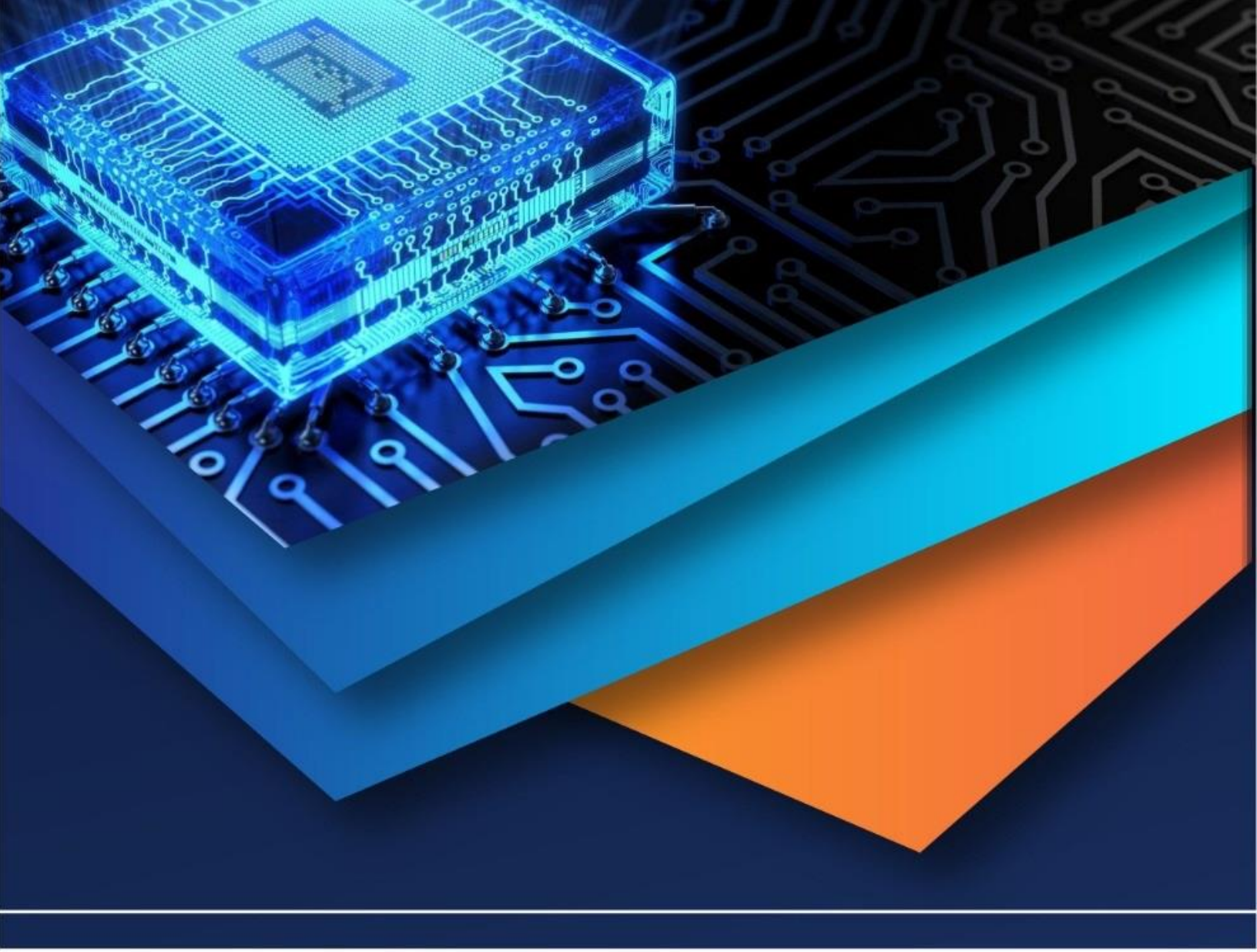
©IJRASET: All Rights are Reserved

Furthermore, the comparative analysis of most efficacious metaheuristic technique, proposed technique, and traditional ways of test case prioritization stated that APFD results of the propose GM-FPA technique are better than the APFD results of CS, GA, FPA, and traditional approaches for test case prioritization in regression testing.

REFERENCES

[1] Bertolino, E. Marchetti. "A brief essay on software testing", in Software Engineering, 3rd ed. Development Process, Nov. 2005, pp. 393-411.
[2] D. Singh, U. Singh, R. Salgotra. "An extended version of flower pollination algorithm", in Arabian J. for Sci. and Engineering, 43(12), Dec. 2018, pp. 7573-603.
[3] D.K. Yadav and S. Dutta. "Regression test case prioritization technique using genetic algorithm", in Advances in Computational Intelligence, Springer, Singapore, 2017, pp. 133-140.
[4] G. Rothermel, R.H. Untch, C. Chu, M.J. Harrold. "Prioritizing test cases for regression testing", in IEEE Transactions on software engineering, 27(10), Oct. 2001, pp. 929-48.
[5] https://github.com/jkriti/JApps/blob/master/JApps.java
[6] https://sir.csc.ncsu.edu/portal/bios/trityp.php
[7] https://stackoverflow.com/questions/4479624/area-and-perimeter-calculation-of-various-shapes?rq=1
[8] P. Dhareula A. Ganpati, 2019. Cuckoo Search Algorithm for Test Case Prioritization in Regression Testing in International Journal of Recent Technology and Engineering (IJRTE) Vol.8(3) ISSN: 2277-3878.
[9] P. Dhareula, A. Ganpati, 2020. Flower pollination algorithm for test case prioritization in regression testing. In *ICT analysis and applications* (pp. 155-167). Springer, Singapore.
[10] P. Konsaard, L. Ramingwong. "Total coverage based regression test case prioritization using genetic algorithm", in Elect. Eng./Electronics, Computer, Telecommunications and Inform. Technology, IEEE, 2015, pp. 1-6.
[11] P.Dhareula, and A. Ganpati, 2019. Software Test Case Prioritization Using Genetically Modified Flower Pollination Algorithm (Gm-Fpa) in International Journal Of Scientific & Technology Research Vol. 8(12), ISSN 2277-8616.
[12] X.S Yang. "Flower pollination algorithm for global optimization", in Int. Conf. on Unconventional Computing and Natural Computation, Springer, Berlin, Heidelberg, Sep. 2012, pp. 240-249.
[13] X.S. Yang, Nature- Inspired Optimization Algorithm, Elsevier Insight, 2014.
[14] X.S. Yang, S. Deb. "Cuckoo search via Lévy flights", in World Congress on Nature & Biologically Inspired Computing, IEEE, Dec. 2009, pp. 210-214.
[15] Z. Li, M. Harman, R.M. Hierons. "Search algorithms for regression test case prioritization", in IEEE Transactions on Software Engineering, 33(4), Apr. 2007.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)