Int. J. Advance Soft Compu. Appl, Vol. 10, No. 1, March 2018
ISSN 2074-2827;

A Novel Representative Dataset Generation Approach for Big Data Using Hybrid Cuckoo Search

Deepa Gupta[1], Ajay Rana[1], and Sanjay Tyagi[2]

[1]Amity University, Noida(UP), India
e-mail: deepa19july@gmail, ajay_rana@amity.edu

[2]Kurukshetra University, Kurukshetra(Haryana), India
e-mail: tyagikuk@gmail.com

Abstract

During recent years, scientists have endeavored to meet the challenging necessities of creating top notch programming applications. Building venture class arrangements like a Big Data Analytics Platform includes testing enormous measures of clinical trial information from different investigations, subjects, and installed gadgets. Handling and putting away terabytes or petabytes of information may take days or weeks to finish. Utilizing an extensive informational collection amid programming advancement and testing, postpones the constant incorporation and conveyance endeavors. A novel method has been proposed to reduce the input data set for such big data applications without compromising on quality of results by creating a smaller representative sample out of given big data datasets and using that representative sample to drive the application further. The proposed approach makes use of Pairwise Test Case Generation methodology to identify he diminished arrangement of datasets ensuring that each pair of input parameters have been enclosed by at least one input data tuple. Although using pairwise methodology might not be exhaustive, but it is found to be very useful because it can significantly reduce the input dataset and can still cover hypothetically difficult relations within different input domain parameters. The paper describes the proposed hybrid pairwise test generation approach based upon Cuckoo Search (CS) and Genetic Algorithms (GA) and applies the approach on big data dataset to come up with representative data sets which are much reduced as compared to original ones. The big data datasets taken as input to prove the effectiveness of proposed approach are the test cases intended to be used for defect finding. The quality of newly generated representative datasets

with proposed approach is also evaluated against the original big sized big data datasets.

Keywords: *Big Data, Software Testing, Pairwise Testing, Genetic Algorithm, Cuckoo Search.*

1 Introduction

Big data analysis (BDA) is where progressive procedures operate on huge data collections. The terminology "Big Data" has recurrently been functional to data sets that develop siege that they end up burdensome to work with using conventional database administration frameworks [1].

Producing test data rapidly, proficiently and precisely is essential however complex to execute particularly in cases where input data is in form of large data sets. If an approach can help engineers to create applications using a minor and illustrative data set, it will considerably reduce the development life cycle, hence permitting earlier feedback during code commit stage.

To appreciate the problem, we are considering a case for a small input data to test a multi-function video playing programming device which have following specs:

Attribute(s)	**Possible configuration**			
Operating System	XP	Win-10	Win-7	Linux
RAM	16 GB	32 GB		
DVD Hardware Type	DVDR	DVDRW	USB	
Display	None	VGA	HDMI	

Total we have 4x2x3x3 = 72 distinct mixes of the information tuples, however a pairwise test framer usually make a littler test suite that wraps every one of the sets in more modest number of information tuples.

Let's say that we want to ensure that every pair of possible input parameters is validated at least once. Here is list of test cases which can be created satisfying this requirement:

OS	**RAM**	**DVD Hardware Type**	**Display**
XP	16 GB	DVDR	None
XP	16 GB	DVDRW	VGA
XP	32 GB	USB	HDMI
Win-10	32 GB	DVDR	VGA
Win-10	32 GB	DVDRW	HDMI
Win-10	16 GB	USB	None
Win-7	16 GB	DVDR	HDMI
Win-7	32 GB	DVDRW	None
Win-7	32 GB	USB	VGA
Linux	16 GB	DVDR	None
Linux	16 GB	DVDRW	VGA
Linux	32 GB	USB	HDMI

This table is directly an exhaustive set of test data for the given issue in the representation. If you consider any two factors from the first test information table and any two possible values for them, there is no less than one entry in the subsequent table that refers to two of those. After applying this pairwise approach to tests generation, we get 12 test blends which give all mixes rather than 72.

Likewise, consider a greater case where a designing automated framework that has 10 buttons to control and each one has capability to configure 10 different conceivable frameworks will have an aggregate of 10^10 (10,000M) mixes, which is significantly more than a product analyzer would be able to test in complete lifetime. If we apply pairwise test case generation to this situation, we can radically bring down total number of test cases, which will unquestionably give all possible mixes.

Now, given a design problem and input variables, the main challenge for pairwise test case generation based approach is to design an algorithm which can come up with minimal number of test cases covering maximum number of possible interactions between all pairs of input variables and with minimum possible runtime.

A study of the results of the currently available techniques which generate the set of test cases using Pairwise Testing indicates that the GA based technique provides one of the powerful way of providing the reasonably good solutions in most of the benchmarking design problems. Although it can provide the reasonably good results in most of the cases, there are still possibilities where it is far from generating the best solutions (e.g. T6) because of its drawback of getting trapped in local maxima and not finding a way to get out of that. Genetic operations help to create the offspring intelligently based upon individual characteristics of chromosomes with high fitness but sometime fail to make a jump to unexplored design regions for the best results. Contrast this to Simulated Annealing (SA) based approach which can process several local Maxima to be able to explore the different design regions and can identify the global Maxima. This is the reason why this approach can generate better results in some of the cases (e.g. T6 T8 (Table1)).

Besides following the search strategy employed by Cuckoo search, GA would ensure that successive generations are based upon good characteristics of previous generations and SA would make sure that the algorithm is not trapped within local maxima. Also, there is further a scope of taking cues from Ant Colony Optimization which brings in the characteristics of developing and exploring the new solutions based upon the trail of previous good solutions. This can add more power to the hybrid algorithm by focusing on the core of good solutions to bring out another set of solutions which can probably help to increase the quality of global solution. All these ideas are tried and explored in the proposed Cuckoo S based approach.

2. Related Work Done

One possible way to solve the above-mentioned problem is to generate the set of test cases with the help of combinatorial testing and make sure to cover all possible inputs combinations during validation. The challenge for combinatorial testing is to manage the huge number of input parameters combinations created because of combinatorial explosion. This problem can be resolved efficiently with the help of Pair-wise testing which is an effective approach to reduce the final set of test cases combinations count in a test suite [2] and can still make sure that most of the possible input variables combinations are covered [3].

Many metaheuristic algorithms have been proposed to come up with respective algorithms to generate the test cases based upon Pairwise Testing technique [4-10]. The major challenge here is to design an algorithm which can come up with a minimal number of test cases covering maximum number of possible interactions between all pairs of input variables and with minimum possible runtime.

Some of these proposed techniques which have been taken as reference in this paper are GAPTS, GA, AETG, ACO, PSO, PWiseGen and Cuckoo Search [11-19]. These proposed approaches have been published along with their results on benchmark test case generation problems which indicate that none of these proposed approaches can give best results in all possible cases (or even for most part of problems).

The results indicate that Genetic and Cuckoo search based approaches can perform well over a wide set of given test case generation problems. While studying the Cuckoo search based approach proposed in [11] it was noticed that the results of this algorithm can be improved if the algorithm is made to take advantages of good characteristics of GA and SA [15]. Cuckoo search is currently being explored to tackle various issues and have been proven successes in many areas such as machine learning [17], the field of truss optimization problems [18], clustering of web results [19], nurse scheduling problems [20], generating test data generation [21], generating independent paths for software testing [22].
The following sections in the paper discuss the results of these approaches briefly and propose a new Pairwise test case generation strategy which is a Hybrid approach based upon Genetic Algorithm [13] and Cuckoo Search technique [11] addressing the shortcomings of these individual strategies. The last section would discuss the results with new proposed approach to demonstrate the effectiveness and proposed strategy and its future work directions.

3. Existing Metaheuristic Algorithms

A meta-heuristic calculation is an iterative approach to manage an arrangement of heuristics by assembling keenly unique area ideas to investigate and misuse the competitor look space. Based upon some pre-identified and agreed upon measures of quality, these algorithms structure the available information using learning strategies and try to find near optimal solutions [10]. This section describes some of the major meta-heuristic algorithms available today (CS, GA, SA, and ACO).

3.1. Cuckoo-Search (CS)

The Cuckoo-Search Algorithm (based upon global optimization) was originally proposed by [11]. This algorithm has been based upon the behavior of how cuckoos lay their eggs and treat them subsequently. Since their proposal, the algorithm has been tried to solve several domain specific problems (neural network training, Reliability Analysis, engineering design, Computer Games etc.) and it has successfully demonstrated its strong usefulness to solve the given problems.

The Cuckoo Search algorithm is a nature inspired algorithm, which draws its motivation from the way Cuckoo bird species lay its eggs and treat it thereafter. The birds from this species are known to lay their respective eggs in nests of other birds. Some of the parasitic cuckoos can even imitate color of eggs of the host bird so that host bird is not able to identify the foreign eggs in his nests. Since host bird is expected to abandon its nest (and create fresh one elsewhere) [11] once it can identify it, female cuckoos try to make sure that the host bird is not able to identify those eggs.

Refer [9], here are the main assumptions of this algorithm:

1. One of cuckoo lays down an egg at a time and puts this in a nest selected randomly.
2. Nests are evaluated based upon a quality function and best nests which have highest fitness of eggs will be passed on to the successive generations. The eggs within these nests represent individual solutions (or part of) to given problem.
3. There are fixed number of total nests available which can act as hosts, and a host bird is assumed to have a probability $P \in (0, 1)$ of detecting a foreign egg in his respective nest. If outside egg is identified by have flying creature, the host fledgling would either tosses out the remote egg or would relinquish its home and make a crisp one at an alternate area.

This algorithm also provides a way to model the third assumption i.e. the way to model the fraction of nests which are replaced with fresh nests. The fresh nests are assumed to be carrying the new solutions chosen randomly. This fraction can be modelled through Levy flight behavior [12].

3.2. Genetic Algorithms (GA)

Genetic Algorithms belong to the category of optimization algorithms which is driven by biological evolution and follow the basis of respective biological operations (such as inheritance, mutation, selection and crossover). A heuristic is developed, based upon these operations and this heuristic is used to guide the algorithm to generate an optimal (or near optimal) solution. The recognized concept of GA was originally developed by John Holland in the 1970s [13], and it has been found suited for application to different domain specific problems where there is very little or no knowledge about the candidate solution set. The formal theory of GA has been successfully demonstrated to work in different areas of engineering, science and businesses [14]. GA based approaches have been found to be effective in different natured (continuous/discrete) combinatorial problems. These approaches work by keeping track of population of candidate solution set for the given problem and by making this population evolve by the application of above mentioned stochastic operators. This evolution happens with the help of different iterations and a new population is created after every iteration. The fitness function for population is expected to improve continuously (or converge) after every successive iteration.

To do operate, GA based approaches require the clear definition of following basic aspects [14] in respect of given problem domain:

- Objective function (which will decide what needs to be optimized and how to evaluate the fitness of given candidate population)
- Genetic representation and implementation (How to encode the problem and candidate solutions into chromosomes form so that algorithm can work on top of these)
- Genetic operators and their implementation (How the GA operators will perform their operations for these encoded chromosomes. It is ok to even tweak the standard GA operators as per the nature of problem)

3.3. Simulated Annealing (SA)

SA is a heuristic based probabilistic method of solving the problems and is focused upon finding the global maxima among a set of many local maxima. The formal theory developed by (Kirkpatrick, Gelett, Vecchi (1983) and Cerny (1985)) [15] inspired from physical annealing process. The name for this approach is also based upon the physical annealing process which is a metallurgical methodology which involves heating/cooling of material in strictly controlled manner. The motivation for this controlled heating/cooling is to crystals formations and thus leads to elimination of any defects.

Taking the analogy from the thermodynamics process (high temperature molecules move freely, low temperatures molecules get stuck) a temperature parameter is modelled into the minimization algorithm. When you get the high temperature, parameter candidate set is explored with full intensity whereas at low temperatures exploration happens with restricted pace.

One of the advantages of SA is that it can guide the algorithm to hit and evaluate several local maxima to point out the global maxima with respect to given fitness function [15]. Although SA is found to be like hill climbing in some aspects but it also has some different aspects. For example, SA also allows the downward steps which are not allowed by hill climbing. Additionally, in SA a move is generated randomly and then it is decided if it should be accepted or not. Here is the algorithm to describe the SA in details
of the advantages of SA is that it can guide the algorithm to hit and evaluate several local maxima to point out the global maxima with respect to given fitness function [15]. Although SA is found to be like hill climbing in some aspects but it also has some different aspects. For example, SA also allows the downward steps which are not allowed by hill climbing. Additionally, in SA a move is generated randomly and then it is decided if it should be accepted or not.

3.4. Ant Colony Optimization (ACO)

ACO is a methodology to find solution for optimization problems which draws its inspiration from the way different ants communicate direction to each other. This approach is based upon the way different ants find the shortest path between their current locations and food source using pheromone trails. The approach came up in 1990s [16] and found to be effective to resolve different types of optimization problems (including machine learning/Network Routing/Quadratic Assignment). The basic concept of this approach is based upon the fact that Ants walk randomly and they follow the trail of lead Ant's path to reach the food destination. While wandering, ants keep on depositing the pheromone along the path which is followed and later other Ants can detect that trail with most pheromone to follow that path. Once an ant finds the food, more and more ants start following the path by following the pheromone trail and thus stop wandering randomly. When the food is finished at food spot, this path is no longer used by ants and therefore pheromone get evaporated and ants subsequently find a way to new food location. Here a given path's attractiveness is determined by the amount of pheromone deposited on that path. Like this if a trail can be created towards optimal solution space, the approach can follow the trail to evaluate and find the optimal solution.

4. Proposed – Representative Dataset Generation Approach for Big Data Using Hybrid Cuckoo Search

At the top level, proposed Hybrid-Pair-CS is a hybrid approach based upon 4 individual approaches: Pair-Cuckoo Search (Pair-CS), GA, SA and ACO. CS serves as the master underlying algorithm for proposed approach and all other approaches have been made to be plugged in at different steps of this approach to contribute towards the objective of achieving best possible results. Each one of these approaches has different roles to play:

Pair-CS [5] is the master algorithm, based upon which, this proposed hybrid algorithm has been developed. It is the composition of three main steps: Generating Binary Combinations, Generating Interaction Elements and finding the optimal set of test cases using CS based master algorithm as proposed in [5].
The core of the Algorithm is fitness function, selected to guide the search algorithm which is based upon number of interaction elements that can be covered by candidate nest. The algorithm iterates over different nests and evaluates each one of these. Considering nest weight, a new nest will be picked as a present nest. In the event of something going wrong if the new nest weight is more noteworthy than current nest weight, the new nest is taken as present nest. The calculation repeats all populace and expels the more terrible nest considering the estimation of pa likelihood. The best nest will be picked up and appended to final test cases and the interaction list is also updated with interaction elements which are now covered. Finding optimal test cases phase is repeated till all interaction elements are covered.

One of the drawbacks of above CS based algorithm is that it solely depends upon the fitness function to evaluate the quality of a given nest and does not take advantage of earlier good quality nests to get a direction to create and select new candidate nests. The proposed algorithm plugin the **GA based operations** to build the new nests and gives the preference to these nests over the rest of nests in case the fitness of these nests is within the given range of other left out nests. The GA based operations (selection, mutation and crossover) make sure to take advantage of earlier high fitness nests to build the new nests. More details of individual GA based operations and how they can help in better navigation of given population can be found in [13-14].

Further experiments with the CS and GA based integrated approach indicated that although the results were better as compared to individual Pair-CS or GA but the algorithm was still not able to perform well in some cases due to getting stuck in local maxima and getting converged due to it despite increased number of generations. Interestingly, it was noticed that for these cases incidentally SA could give better results because SA could process several local maxima and has got less chances of getting stuck in local maxima. This motivated the Hybrid-PairCS to take advantage of **SA based approach** [15] and it has plugged in this SA while selecting and including the best available solutions. Due to SA a less than best solution finds a way into candidate set to occasionally guide the algorithm in a different direction to be able to explore the unexplored region.

Input N: Total number of Parameters, and
K: set of possible values for each one of parameter = [K0 ..Kj]
Output: List of testcases *TS*;
Let IPairs all Interaction Pairs.
Let TS be a set of candidate tests;
Trigger Generation of initial host nest population randomly
MaxGeneration = MaxGeneration1;
Mode = modeGA;
while IPairs is not empty do
while (t < MaxGenerationCount or stop critera not satisfied) continue
if (Mode == modeGA) {
Generate a cuckoo (i) by Levyflights to imitate randomness;
} else if (Mode == modeACO) {
Get a cuckoo (i) from cuckoo collection which have given best offsprings in modeGA
}
When generating new solutions for the ith cuckoo, the following Lévyflight is performed

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\lambda)$$

Calculate the quality (fitness) Qi of above Cuckoo;
Pickup a candidate solution (nest) from N (j) in random fashion;
if (Qi > Qj)
Replace picked up N(j) with a new solution (nest);
end if
Dump a part of (based upon (P)) of worst nests
Create new nests using following GA operations at new locations
GA basic operations {Selection (1 , Production(2), Evaluation(3)) {
(1): build mating pool
(2): Mutation operations (flip/swap/slide)
(3) Population Evaluation
};
Retain the best possible solutions. The fitness of each possible solution E_i may be computed using following formula ($O_{k,i}$ is the result obtained by solution E_i for case k and w_k is the targeted result) :

$$f(E_i) = \sum_{k=1}^{n} |o_{k,i} - w_k|$$

Order the candidate solutions based upon respective ranks
Find the current best with following SA Algo :
If (Quality(Si) >=Quality(Sbest))
Sbest=Si
Else If (Exp((Quality(Si)- Quality(Sbest))/weight(Si)) > Rand())
Sbest =Si
End
The acceptance criteria in this case is basically based upon following equation where R(0,1) is the random number chosen at regular intervals and Q delta represent the difference in quality of given solution :

$$e^{\frac{-\Delta Q}{T}} > R(0,1)$$

*If (t == (*MaxGeneration-1)) {
MaxGeneration = MaxGeneration2;
mode = modeACO;
}
end loop
Append the best candidate testcase into TS.
Evaluate the new set of pair interactions which are covered and update IPairs.
end loop
end-Proc

Fig 1 : Steps of Dataset Generation For Big Data Using Hybrid Cuckoo Search

The results of proposed Algorithm were improved significantly for the problematic test configurations and inched closure to SA based results. Now we tried to take advantage of Ant Colony optimization on top of hybrid algorithm proposed above. As per the **ACO based approach** [16], once an ant finds the food, more and more ants start following the path by following the pheromone trail and thus stops wandering randomly. Similarly, to this phenomenon, we extended the above proposed to make second pass of the algorithm itself but this time the candidates are not the whole set of available combinations but the cases selected in first iteration. The idea was to look around in the set of test cases which have proven their fitness and tried to see if some of them can be replaced with new ones (covering same number, of pairwise combinations with less test cases) or with the one having higher fitness. Since the candidates set is restricted this step is anyway not found to be too much time consuming.

It is observed that after introducing the 2nd set of enhanced generations (Taking the cues from SA and ACO) the algorithm could pick up the test cases in unexplored category as well and thus leading to increase in quality coverage (coverage with less number of test cases). The proposed strategy is summarized in algorithm give below.

5. Experiment and Results with Big-Data Test configurations

To establish the relevance of the proposed Hybrid-Pair CS with respect to already established pairwise testing techniques, it was evaluated on some benchmark big data test configurations and the results also compared with that of earlier available approaches (see Tables 1) [26-38]. The Table 1. given below specifies the minimum number of test cases generated with already proposed approaches and the new one to ensure the exhaustive test coverage for given test configuration. The approach with minimum number of test cases in most number of test configurations will be the winner because it will be the one which would take least time to complete the testing with sufficient coverage for most of the cases. The experiments use standard test cases configurations available on web which are already evaluated by previous approaches as a measure of their effectiveness. The outcomes show that while new proposed approach can give the best accessible outcomes in 6 cases, it is additionally ready to offer near best outcomes on 4 different cases. Since no accessible approach can perform best for all test setups, the proposed hybrid approach is positioning high considering the normal number of experiments required for all test designs and furthermore on the size of methodologies with most elevated number of best available solutions.

Table 1. Number of test cases required to exhaustively cover given Big Data test configurations

	Currently available Testcase Generation approaches alongwith proposed approach (Hybrid-PairCS)														
Different Big Data Testcase Configurations for results generations	**Conf**	**PICT**	**AETG**	**AllPairs**	**Jenny**	**IPO**	**IPOG**	**IRPS**	**SA**	**GA**	**ACA**	**PPSTG**	**PHSS**	**PairCS**	**Hybrid-PairCS**
	T1	10	NA	10	9*	NA	11	9*	NA	NA	NA	9*	9*	9*	9*
	T2	13	9*	10	13	9	12	9*	9*	9*	9*	9*	9*	9*	9*
	T3	20	15*	22	20	17	20	17	16	17	17	17	18	18	15*
	T4	170	NA	177	157	169	176	149*	NA	157	159	170	155	151	150
	T5	NA	NA	390	336	361	373	321*	NA	NA	NA	NA	341	333	323
	T6	NA	180	230	NA	212	NA	210	183*	227	225	NA	224	209	187
	T7	47	NA	49	45	47	50	45	NA	NA	NA	45	43	42*	42*
	T8	21	19	21	41	NA	19	17	15*	15*	16	21	20	20	15*
	T9	38	34	NA	31*	NA	36	NA	NA	33	32	39	39	38	33
	T10	46	45	NA	51	NA	44	NA	NA	42*	42*	49	48	47	42*
	T11	101	NA	NA	98	NA	91*	NA	NA	NA	NA	97	95	96	93

Legend:
Big Data Reference configurations:
T1: 3^3 **T2: 3^4** **T3: 3^{13}** **T4: 10^{10}** **T5: 15^{10}**
T6: 10^{20} **T7: 5^{10}** **T8: $5^1\ 3^8\ 2^1$**
T9: $6^1\ 5^1\ 4^6\ 3^8\ 2^3$ **T10: $7^1\ 6^1\ 5^1\ 4^6\ 3^8\ 2^3$** **T11: $10^1\ 9^1\ 8^1\ 7^1\ 6^1\ 5^1\ 4^1\ 3^1\ 2^1$**
x^y means : means that task take x parameters, each parameter with y values.

While the prospective advantages of Big Data are huge and promising, and some underlying triumphs have just been accomplished, there stay numerous specialized difficulties that should be routed to completely understand this potential. The huge chunk of information, obviously, is a noteworthy test, and is the one that is most effortlessly perceived. Be that as it may, there are others. Industry examination organizations jump at the chance to call attention to that there are challenges in Volume, as well as in Variety and Velocity [39]. The above results table indicates that there is huge amount of opportunity to reduce the sheer size of incoming data by application of proposed pairwise generation methodology.

6. Conclusion

Often in case of big data projects, data generation has had many issues including the generation of little and illustrative data set from source to fulfil diverse kinds of limitations including statistical analysis. To understand the difficulties, we developed another strategy utilizing pairwise test case era to make the illustrative informational collections in the huge information condition. With the proposed approach; input space segment testing starts with an information area show ("IDM"). The analyzer sections the IDM, picks test esteems from distributed pieces, and applies combinatorial scope criteria to deliver test information. To the best of our mindfulness, this is the first run through pairwise test case era procedure has been utilized to produce delegate informational indexes got from source in the huge information setting. In an agile procedure, testing applications with gigabytes or terabytes of data is costly. Our demonstrative test generator methodology will guarantee quality and fundamentally abbreviate testing and engineering cycles by generating representative datasets without having to process terabytes of data. Using the approach saves time, expenses, and manual endeavors.

References

[1] Nada Elgendy and Ahmed Elragal, "Big Data Analytics: A Literature Review Paper, " ICDM 2014, pp. 214–227, 2014.

D. M. Cohen, S. R. Dalal, J. Parelius, and G. C. Patton, "The combinatorial design approach to automatic test generation," IEEE software, pp. 83-88, 1996.

[2] X. Chen, Q. Gu, J. Qi, and D. Chen,"Applying particle swarm optimization to pairwise testing," in Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual, 2010, pp. 107-116.

[3] D. R. Kuhn, D. R. Wallace, and J. AM Gallo, "Software fault interactions and implications for software testing," Software Engineering, IEEE Transactions on, vol. 30, pp. 418-421, 2004.

[4] B. S. Ahmed, K. Z. Zamli, and C. Lim, "The development of a particle swarm based optimization strategy for pairwise testing," Journal of Artificial Intelligence, vol. 4, pp. 156-165, 2011.

[5] A.B. Nasser, Y.A. Sariera, A. Rahman, A. Alsewari, K. Z. Zamli , "A Cuckoo Search Based Pairwise Strategy For Combinatorial Testing Problem", Journal of Theoretical and Applied Information Technology Vol.82. No.1 pp 154-162, 2015

[6] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence, "IPOG/IPOG-D: efficient test generation for multi-way combinatorial testing," Software Testing,Verification and Reliability, vol. 18, pp. 125-148, 2008.

[7] Y. Lei and K.-C. Tai, "In-parameter-order: A test generation strategy for pairwise testing," in High-Assurance Systems Engineering Symposium, 1998.Proceedings. Third IEEE International, 1998, pp. 254-261.

[8] M. B. Cohen, "Designing test suites for software interaction testing," Citeseer, 2004.

[9] X.-S. Yang and S. Deb, "Cuckoo search: recent ances and applications," Neural Computing and Applications, vol. 24, pp.169-174, 2014.

[10] I. H. Osman and G. Laporte, “Metaheuristics:A bibliography”, Ann. Oper. Res. vol. 63, (1996), pp. 513-623.

[11] X. S. Yang and S. Deb, “Cuckoo Search via Levy Flights”, Proceedings of World Congress on Nature & Biologically Inspired Computing, (2009), pp. 210-225.

[12] E.Valian, S. Mohanna and S. Tavakoli, “Improved Cuckoo Search Algorithm for Global Optimization”, International Journal of Communications and Information Technology, IJCIT, vol. 1, no. 1, (2011), pp. 1-62.

[13] M. Melanie, “An Introduction to Genetic Algorithms”, MIT Press, (1996).

[14] B. F. Al-Dulaimi and H. A. Ali, “Enhanced Traveling Sale sman Problem Solving by Genetic Algorithm Technique (TSPGA)”, World Academy of Science, Engineering and Technology, vol. 14, (2008), pp. 296-302.

[15] D. Bertsimas and J. Tsitsiklis, “Simulated Annealing”, Statistical Science, vol. 8, no. 1, (1993), pp. 10-15. [16] I. Brezina and Z. Č. ková, “Solving the Travelling Salesman Problem Using the Ant Colony”, Management Information Systems, vol. 6, no. 4, (2011), pp. 10-14. [17] R. A. Vazquez, "Training spiking neural models using cuckoo search algorithm," in Evolutionary Computation (CEC), 2011 IEEE Congress on, 2011, pp. 679-686.

[18] A. H. Gandomi, S. Talatahari, X. S. Yang, and S. Deb, "Design optimization of truss structures using cuckoo search algorithm," The Structural Design of Tall and Special Buildings, vol. 22, pp. 1330-1349, 2013.

[19] C. Cobos, H. Munoz-Collazos, R. Urbano-Munoz, M. Mendoza, E. Leon, and E. Herrera-Viedma, "Clustering of Web Search Results based on the Cuckoo Search Algorithm and Balanced Bayesian Information Criterion," Information Sciences, 2014.

[20] L. H. Tein and R. Ramli, "Recent advancements of nurse scheduling models and a potential path," in Proceedings of 6th IMT-GT Conference on Mathematics, Statistics and its Applications, 2010, pp. 395-409.

[21] K. Perumal, J. M. Ungati, G. Kumar, N. Jain, R. Gaurav, and P. R. Srivastava, "Test data generation: a hybrid approach using cuckoo and tabu

Search," in Swarm, Evolutionary, and Memetic Computing, ed: Springer, 2011, pp. 46-54.

[22] P. R. Srivastava, R. Khandelwal, S.Khandelwal, S. Kumar, and S. Santebennur Ranganatha, "Automated test data generation using cuckoo search and tabu search (csts) algorithm," 2012.

[23] A. Hartman and L. Raskin, "Problems and algorithms for covering arrays," Discrete Mathematics, vol. 284, pp. 149-156, 2004.

[24] R. Mandl, "Orthogonal Latin squares: an application of experiment design to compiler testing," Communications of the ACM, vol. 28, pp. 1054-1058, 1985.

[25] K. A. Bush, "Orthogonal arrays of index unity," The Annals of Mathematical Statistics, vol. 23, pp. 426-434, 1952.

[26] A. W. Williams, "Determination of test configurations for pair-wise interaction coverage," in Testing of Communicating Systems, ed: Springer, 2000, pp. 59-74.

[27] D. M. Cohen, S. R. Dalal, M. L. Fredman, and G. C. Patton, "The AETG system: An approach to testing based on combinatorial design," Software Engineering, IEEE Transactions on, vol. 23, pp. 437-444,1997.

[28] A. Williams, "TConfig download page [Online]," p. University of Ottawa. Available: http://www.site.uottawa.ca/~awilliam/[Accessed 23 Dec 2014]. 2008.

[29] B. Jenkins, "Jenny download page [Online]," p. Available : http://www.burtleburtle.net/bob/math.[Accessed 16 Dec 2014]. 2003.

[30] A. Hartman, T. Klinger, and L. Raskin, "IBM intelligent test case handler," Discrete Mathematics, vol. 284, pp. 149-156, 2010.

[31] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence, "IPOG: A general strategy for t-way software testing," in Engineering of Computer-Based Systems, 2007. ECBS'07. 14th Annual IEEE International Conference and Workshops on the, 2007, pp. 549-556.

[32] M. Forbes, J. Lawrence, Y. Lei, R. N.Kacker, and D. R. Kuhn, "Refining the inparameter- order strategy for constructing covering arrays," Journal of Research of the National Institute of Standards and Technology, vol. 113, pp. 287-297, 2008.

[33] M. Patil and P. Nikumbh, "Pair-wise testing using simulated annealing," Procedia Technology, vol. 4, pp. 778-782, 2012.

[34] T. Shiba, T. Tsuchiya, and T. Kikuno, "Using artificial life techniques to generate test cases for combinatorial testing," in Computer Software and

Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International, 2004, pp. 72-77.

[35] C. Nie and H. Leung, "A survey of combinatorial testing," ACM Computing Surveys (CSUR), vol. 43, p. 11, 2011.

[36] X.-S.Yang and S. Deb, "Cuckoo search via Levy flights," in Nature & Biologically Inspired Computing, 2009. NaBIC 2009.World Congress on, 2009, pp. 210-214.

[37] J. Aviles, B. Stokke, A. Moksnes, E. Roskaft, M. Asmul, and A. Moller, "Rapid increase in cuckoo egg matching in a recently parasitized reed warbler population," Journal of evolutionary biology, vol. 19, pp. 1901-1910, 2006.

[38] A. B. Nasser, A. R. A. Alsewari, and K. Z.Zamli, "Tuning of Cuckoo Search Based Strategy for T-way Testing," in International Conference on Electrical and Electronic Engineering, 2015.

[39] Pattern-Based Strategy: Getting Value from Big Data. Gartner Group press release.July2011.Available at http://www.gartner.com/it/page.jsp?id=1731916