



## Review of Combinatorial Testing Strategy

J M Altmemi<sup>1</sup>, R R Othman<sup>2</sup>, R Ahmad<sup>3</sup>

<sup>1</sup>School of Computer and Communication, Universiti Malaysia Perlis, kampus Pauh Putra, Arau, perlis, Malayisa,

<sup>1</sup>Iraq University College, Basra, Iraq, JalalAltmemi@gmail.com

<sup>2</sup>School of Computer and Communication, Universiti Malaysia Perlis, kampus Pauh Putra, Arau, perlis, Malayisa, rozmie@unimap.edu.my

<sup>3</sup>School of Computer and Communication, Universiti Malaysia Perlis, kampus Pauh Putra, Arau, perlis, Malayisa, badli@unimap.edu.my

### ABSTRACT

Combinatorial testing (CT) is an essential technique that usually aims for the generation of test cases. The generation of an optimally sized test case is a Non-deterministic Polynomial (NP) problem. The Combinatorial Interaction has been discussed by several researchers in developing different strategies depending on support interactions. Combinatorial testing taxonomy is thus Suggested to facilitate the analysis. In this paper, 12 algorithms were evaluated based on the strategy approach, supported interactions, randomness, and publication year. The year 2011 was noted as the most successful year in which researchers generated t-way algorithms. For them, the one-test-per-time OTAT strategy was the most promising area for Combinatorial t-way analysis. A small variation was seen in the type of interaction support. Nonetheless, uniform strength was the most common form of interaction from the year 2010 to the first half of the year 2018. This article was therefore implemented to review existing t-way strategies to identify their capabilities and limitations, thus providing a valuable guideline for generating the future.

**Key words:** Combinatorial testing, T-Way Testing, Interaction Testing.

### 1. INTRODUCTION

Computer software is considered an important element these days, which is increasing every day [1, 2]. Software implementation has become an essential component due to its massive benefits. Therefore, performing software utilization properly will help in generating user satisfaction. Software failures can generally be avoided by conducting software testing [1, 3]. However, software testers will be driven to the limit upon exhaustive testing being carried out. When a large number of test cases need to be applied, software testing may be unable to execute the process [4, 5]. In fact, there is a risk of an increasing number of test cases that helps boost time and cost. Therefore, formulating the combinatorial software testing is a new testing technique that serves to address the issue of exhaustive testing, which includes an interaction of parameters. In this case, software failures are discovered, resulting in the interaction of several input parameters called

t-way tests; t refers to the strength of the interaction [6, 7]. CA can be formulated as CA (N, v<sup>P</sup>, t), where N \* P defines the input matrix [0, 1, 2 ..., up to (v - 1)], so that each set of t-columns include at least once of each possible t-tuple, where v is a value, P the degree and t the strength [4, 8]. Several strategies, as well as improved t-way combinatorial testing techniques, have also been developed to optimize any algorithm and to generate smaller test sequences or an almost optimal number of tests. difficult problem [1, 9, 10]. Many researchers in this area have been encouraged by this scenario. Therefore, the purpose of this article is to explore existing algorithms or tools, as well as the current status and future possibilities of t-way combinatorial testing. In addition, this article also discusses possible improvements in t-way testing. It focuses on articles published from 2010 to 2019 on combinatorial testing, whereby published articles from different kinds of existing databases are sorted and ranked by strategic approach, search interaction, and publication year. The purpose of this article is to explore and analyze existing algorithms relevant to t-way testing, as well as to find alternative enhancements for t-way testing. The next section presents a review of combinatorial tests, while the third section describes existing algorithms. Then, the fourth section highlights the analysis and discussion. Finally, the conclusion is given in section four.

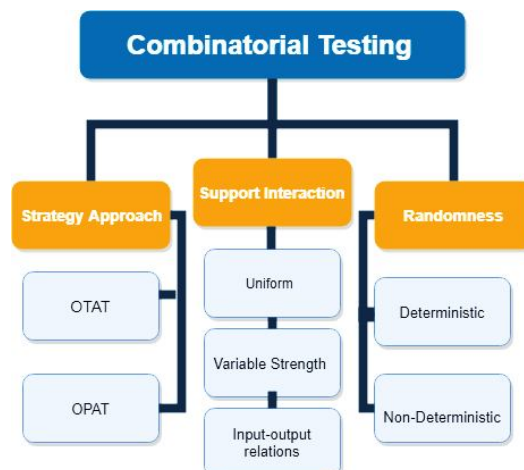


Figure 1: Support areas for Strategies on the T-way

### 1.1-Strategic Approach

Much current research has focused on a computer science approach. It is possible and divided into two types of computer approaches, namely one test at a time (OTAT) and one parameter at a time (OPAT). The strategy in the first situation requires that each test case generated covers at least one unpaired tuple, while the optimal test covers the most unpaired t-way interactions for all parameters. Once the test case is selected, a final test suite will be added. This is called AETG vertical expansion [8, 11]. The OPAT approach starts with an empty initial test scenario, followed by a generation or completed test scenarios. Finally, it is added to a final test suite (FTS) one after the other when all interactions are covered by the FTS (i.e., a horizontal extension). Once completed, new test cases (vertical extension) can be added to obtain maximum coverage of interactions [10, 12]. The Parameter Order Strategy (IPO) was developed by Lei and Tai and represents innovative research using the OPAT approach [13, 14]. Figure 3 highlighted 90 % of cases involved OTAT strategy, whereas 10% OPAT strategy.

### 1.2 Support Interaction

The second description of the t-way strategy is the interaction between support, which can be subdivided into three children, namely uniform , variable strength, and input-output relationships [1, 15]. Uniform strength or t wise interaction: Uniform strength Interaction takes into account the fact that only one interaction strength is found between the parameters. This is the basis of the interaction test in which the CA has the same intensity interaction for all input parameters [9, 16]. Variable Strength (VS) extends the uniform Strength. The difference between the uniform Strength and Variable Strength is that it alone guarantees and has more than one interaction strength for generating test cases[10]. Input-Output Relationships (IORs) involve the use of input-output relationship information for interaction tests. It is affected by a combination of related parameter inputs that impact on certain outputs. The IOR is launched because not all systems have the same attributes. it is also intended to avoid redundancy, test cases or the use of unnecessary test cases [9, 17].

### 1.3-Randomness

Randomness has two kinds first a deterministic set of combination strategies that participate in a property that creates the same test suite for each execution. While the second has non-deterministic concerns, the property of each execution requires a randomly created combination suite to cover all necessary t-way pairs [22].

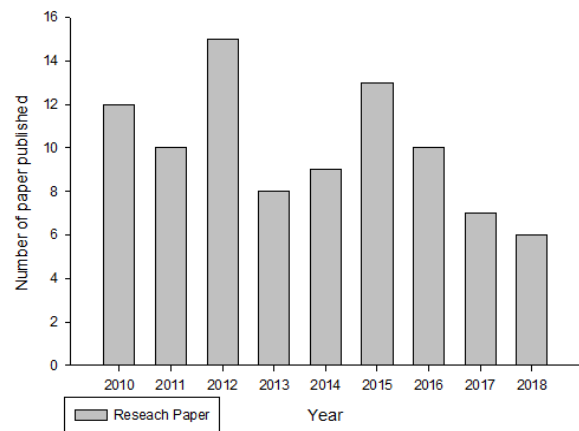
## 2. EXISTING T-WAY STRATEGIES

Several algorithms and techniques have been researched for use in combination with t-way testing. In this article, the focus was placed on existing algorithms produced from 2010 to 2018. Table 1 discusses the current algorithms as identified by

the year of publication. And Figure 2 shows Number of research papers on CT test suite generation.

**TABLE 1:** comparison between strategies

Year of publishing	T-Way Strategies
2010	Particle Swarm Optimization (PSO)
	Pairwise PSO (2010)
	Particle Swarm Test Generator (PSTG)
2011	Generalized T way test suite Generator (GTWay)
	AURA
	Integrated T-Way Test Data Generation (ITTDG)
	Harmony Search (HS) Harmony Search Strategy (HSS)
2012	Discrete Particle Swarm Optimization (DPSO)
	Cuckoo Search (CS)
2015	Flower Strategy (FS)
	Ant Colony System (ACS)
2016	Sine Cosine Algorithm (SCA)
	Adaptive Teaching Learning Based Optimization (ATLBO)
2017	Adaptive Teaching Learning Based Optimization (ATLBO)
	pATLBO_RO pairwise Adaptive Teaching Learning Based Optimization Remedial Operator
2018	pATLBO_RO pairwise Adaptive Teaching Learning Based Optimization Remedial Operator



**Figure 2:** Number of Research papers on CT Test Suite Generation

Particle swarm optimization (pso) the most popular t-way or combination tests were published in 2010. the algorithm mimics the behavior of bird swarms when searching for food. Each swarm individual moves to the best place and position in the world using the best solution. After that, their position and velocity are calculated [18, 19]. Several algorithms based on the PSO algorithm have been developed. In previous studies, active PSO algorithms have facilitated enhancement of the particle swarm test generator (PSTG) [20]. The reason why PSO is chosen as the core platform of PSTG for many reasons, such as faster convergence rate behavior, just a few parameters to control, can be effectively introduced into any

optimization problem and a load of optimization. less heavy calculation. Unfortunately, the PSTG can only support a uniform strength of up to six.

Wu and co-workers [20] [2015] developed a new PSO-based DPSO. The recent DPSO improves performance using two methods, namely particle reset and additional gbest evaluation. The DPSO has performed much better than the traditional PSO [21, 22].

In 2011, GTWay was an OTAT strategy [15, 23] that introduced computational search algorithms, which used the concept of backwards. The algorithm must map the real data as a symbolic representation and then only possible interactions between t-way pairs are generated. When done, the strategy should be to go back to the discovered t-way pairs. The t-way pairs are combined when they can be integrated. As such, they complement each other if the new combination can include the least-known pairs or pathways. When the pairs fail to integrate, the strategy returns to the initial values identified. However, GTWay [15] only supports a uniform strength with a maximum of six.

In 2011, the ITTDG [14] was discovered. It is an OTAT strategy that supports variable strength interaction. Because a test scenario is iteratively generated to cover all interaction nodes, each test scenario generates multiple test case candidates and selects the best solution that covers the most discovered nuances. The best solution will be added to the final test case. ITTDG supports the three types of interaction: uniform, variable and IOR.

PICT[21] is a test suite strategy developed by Czerwonka that has been widely used for software testing at Microsoft. In the beginning, PICT generates all the possibilities of nuances on the SUT configurations and identifies each tuple as an uncovered tuple. The tuples in these restrictions will be marked as eliminated. After that, an uncovered tuple is selected and extended with the gourmet method until its completion and the completed test scenario will be added as a final test. All tuples covered by this test case are marked. This procedure is repeated until there are no uncovered tuples mark.

In 2012, Alsewari invented a HHS-based strategy or HS [24]. HSS imitating the conduct of musicians trying to compose extraordinary music from improvisations, such as modifying a memory melody or random sampling. HSS performs its imitation by sequentially exploiting harmony memory to store the appropriate solution in its local and global search processes through several specified improvisations. A test case is selected for the final test sequence in each improvisation until all desired interactions are covered. CS [25] proposed by Ahmed et al (2015) is a meta-heuristic technique for generating a test scenario that mimics the unique lifestyle and aggressive breeding strategy of cuckoos. Initially, CS generates random initial eggs on the nest of a host bird. The number of eggs in the nest represents the different solutions (each egg represents the test case). Each generation performs two operations. A new nest is initially generated and evaluated against existing nests via a Levy flight. The new

nest is equipped with the current nest if its objective function is improved. Finally, CS adopts elitism to preserve elite solutions for the next round.

AURA [26], as proposed by Ong and Zamli (2011), exploits the combination of recently produced interactions to generate a suite of tests. Then, a real-time mapping algorithm is used to aid symbolic values and effective data output. FS [27], published by Nasser and Al Sarierai (2015), was developed from the activity of the flower pollination algorithm. The FPA is simple, scalable and involves a light calculation. The main advantage of this algorithm is that it is the key to the balance between exploitation and exploration using Levy flight. FS can withstand both uniform strength and VS.

ACS [28] proposed by Ramli and Othman (2016) is a new version of the Anti-Colony (ACO) algorithm. It has successfully solved several combinatorial optimization problems and has been developed to generate an optimized test suite size that supports uniformity and VS. SCA [29], as presented by Mirjalili (2016), mimics the mathematical behaviors of the sine and cosine. The solutions are updated according to the sine or cosine equations. To facilitate the search process, SCA implements four random parameters (namely,  $R_1$ ,  $R_2$ ,  $R_3$  and  $R_4$ ). Although it is very effective in solving optimization problems, it has a weakness between exploration and exploitation.

ATLBO [30] is a t-way meta-heuristic test that evolved from TLBO algorithms. According to Mamdani's fuzzy inference system, it allows efficient selection of local and global searches through the selection of fuzzy inference. ATLBO can withstand a uniform and variable strength. In addition, pATLBO\_RO [31] proposed by Din and Zamli (2018) is based on ATLBO, but with a correction operator.

### 3. ANALYSIS AND DISCUSSION

This section presents the analysis of Section 3, with existing work and other related issues.

#### 3.1 Supported interaction and strategy analysis

Table 2 describes the summary analysis of the t-way strategy discussed in the previous section. In summary, it presents the algorithms based on their interactions, strategy and randomness. All algorithms support uniform interaction. The second most popular type of interaction is variable strength. The last IOR shows in Table 2 that three algorithms are supported and proposed to support all types of interactions, in particular IOR. Algorithm development was least published in 2012, 2014 and 2017. However, this report included information from the first half of 2019. The adaptive pair algorithm pATLBO\_RO was the only algorithm released in 2018. The algorithms published from 2010 to 2018. It is clear from the table that none of the OPAT algorithms supports IOR. Nevertheless, PSO is considered one of the most used algorithms compared to the other algorithms mentioned; thus, many algorithms have been developed based on PSO (i.e., PSTG, DPSO, FSAPSO) [22, 30, 32]. Figure 3 highlighted the

number of Uniform, variable strength and IOR supported interactions by year.

Number of Uniform, Variable and IOR Supported Interaction by Year

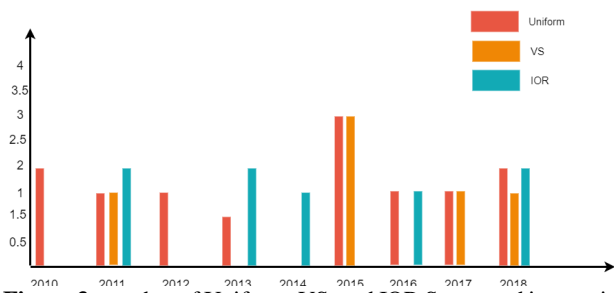


Figure 3: number of Uniform, VS, and IOR Supported interaction by year

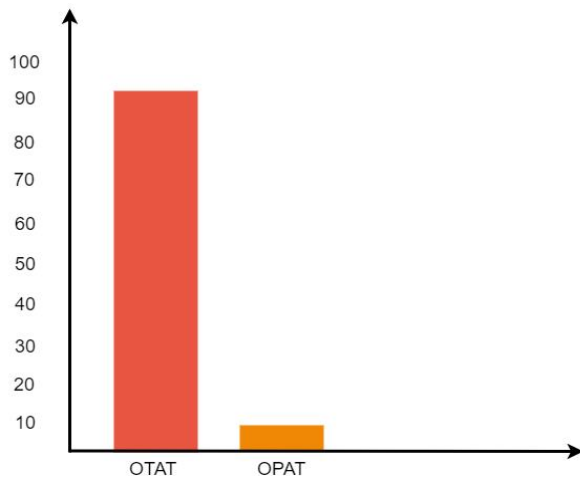


Figure 1: OTAT and OPAT Strategy

Table 2: Analysis of T-Way Strategies

T-Way testing strategies	Supported Interactions			Strategy Approach		Randomness	
	Uniform Strength	Variable Strength	Input-Output Based Relations	OTAT	OPAT	Deterministic	Non Deterministic
(PSO)	S	NS	NS	NS	S	NS	S
(PSTG)	S	NS	NS	S	NS	NS	S
(GTWay)	S	NS	NS	S	NS	S	NS
AURA	S	S	S	S	NS	S	NS
(ITTDG)	S	S	S	S	NS	S	NS
(HSS)	S	S	NS	S	NS	NS	S
DPSO	S	S	NS	S	NS	S	NS
CS	S	S	NS	S	NS	NS	S
FS	S	NS	NS	S	NS	NS	S
ACS	S	S	S	S	NS	S	NS
SCA	S	S	S	S	NS	NS	S
(ATLBO)	S	S	NR	S	NS	S	NS

S: Supported , NS: Not Supported, NR : Not reported

#### 4. CONCLUSION

This article presents t-way combinatorial tests and presents all categories, such as strategic approach, support interaction, and randomness. In addition, the manner in which the proposed strategy was highlighted has been defined. Refer to TABLE 2 Note that many algorithms have been developed using the OTAT strategy. only PSO uses OPAT's strategy. OTAT is common and preferable because it supports a configuration higher than OPAT because its strategy is the most widely used search technique compared to OPAT. IOR of 50% (CS, PSTG, SCA, RSS and FS) and appeared to be a non-deterministic strategy. In addition, in a previous study, the deterministic strategy has a weakness in the fully assisted interaction with the non-deterministic strategy. Finally, this study has successfully proposed a new strategy for future work.

#### REFERENCES

1. Abdel-Basset, M. and L.A. Shawky, Flower pollination algorithm: a comprehensive review. Artificial Intelligence Review, 2018: p. 1-25. <https://doi.org/10.1007/s10462-018-9624-4>
2. Ghandehari, L.S., et al., A combinatorial testing-based approach to fault localization. IEEE Transactions on Software Engineering, 2018.
3. Czerwonka, J., Pairwise testing in the real world: Practical extensions to test-case scenarios. Microsoft Corporation, Software Testing Technical Articles, 2008.
4. Ahmed, B.S., et al., Code-aware combinatorial interaction testing. IET Software, 2019. <https://doi.org/10.1049/iet-sen.2018.5315>
5. Alsewari, A.A. and K.Z. Zamli. Interaction test data generation using harmony search algorithm. in 2011 IEEE Symposium on Industrial Electronics and Applications. 2011. IEEE.
6. Ahmed, B.S., Test case minimization approach using fault detection and combinatorial optimization techniques for configuration-aware structural testing. Engineering Science and Technology, an International Journal, 2016. 19(2): p. 737-753.
7. Din, F. and K.Z. Zamli. Pairwise Test Suite Generation Using Adaptive Teaching Learning-Based Optimization Algorithm with Remedial Operator. in International Conference of Reliable Information and Communication Technology. 2018. Springer.
8. Choi, E.-H., et al. Test effectiveness evaluation of prioritized combinatorial testing: A case study. in 2016 IEEE International Conference on Software Quality, Reliability and Security (QRS). 2016. IEEE. <https://doi.org/10.1109/QRS.2016.17>
9. Ammann, P. and J. Offutt, Introduction to software testing. 2016: Cambridge University Press.
10. Cassels, J. and A. Godbole, Covering Arrays for Equivalence Classes of Words. arXiv preprint arXiv:1803.06507, 2018.
11. Jain, N., U. Nangia, and J. Jain, A review of particle swarm optimization. Journal of The Institution of Engineers (India): Series B, 2018. 99(4): p. 407-411.

12. Ahmed, B.S., et al., Towards an Automated Unified Framework to Run Applications for Combinatorial Interaction Testing. arXiv preprint arXiv:1903.05387, 2019.
13. Chen, X., et al. Variable strength interaction testing with an ant colony system approach. in 2009 16th Asia-Pacific Software Engineering Conference. 2009. IEEE. <https://doi.org/10.1109/APSEC.2009.18>
14. Othman, R.R. and K.Z. Zamli, ITTDG: Integrated T-way test data generation strategy for interaction testing. Scientific Research and Essays, 2011. 6(17): p. 3638-3648.
15. Zamli, K.Z., et al., Design and implementation of a t-way test data generation strategy with automated execution tool support. Information Sciences, 2011. 181(9): p. 1741-1758.
16. Lawrence, J., et al., A survey of binary covering arrays. the electronic journal of combinatorics, 2011. 18(1): p. 84.
17. Cohen, M.B., et al. A variable strength interaction testing of components. in Proceedings 27th Annual International Computer Software and Applications Conference. COMPAC 2003. 2003. IEEE.
18. Zhang, T. and Z.W. Geem, Review of harmony search with respect to algorithm structure. Swarm and Evolutionary Computation, 2019. 48: p. 31-43. <https://doi.org/10.1016/j.swevo.2019.03.012>
19. Ahmed, B.S. and K.Z. Zamli. T-way test data generation strategy based on particle swarm optimization. in 2010 Second International Conference on Computer Research and Development. 2010. IEEE.
20. Ahmed, B.S., K.Z. Zamli, and C.P. Lim, constructing a t-way interaction test suite using the particle swarm optimization approach. International Journal of Innovative Computing, Information and Control, 2012. 8(1): p. 431-452.
21. Alsewari, A.R.A. and K.Z. Zamli, Design and implementation of a harmony-search-based variable-strength t-way testing strategy with constraints support. Information and Software Technology, 2012. 54(6): p. 553-568.
22. Al-Sewari, A.A. and K.Z. Zamli. An orchestrated survey on t-way test case generation strategies based on optimization algorithms. in The 8th International Conference on Robotic, Vision, Signal Processing & Power Applications. 2014. Springer. [https://doi.org/10.1007/978-981-4585-42-2\\_30](https://doi.org/10.1007/978-981-4585-42-2_30)
23. Klaib, M.F.J., Development of an automated test data generation and execution strategy using combinatorial approach. PhD, School of Electrical and Electronics, Universiti Sains Malaysia, 2009.
24. Alsariera, Y.A., et al., A bat-inspired testing strategy for generating constraints pairwise test suite. Advanced Science Letters, 2018. 24(10): p. 7245-7250.
25. Ahmed, B.S., T.S. Abdulsamad, and M.Y. Potrus, Achievement of minimized combinatorial test suite for configuration-aware software functional testing using the cuckoo search algorithm. Information and Software Technology, 2015. 66: p. 13-29.
26. Ong, H. and K.Z. Zamli, Development of interaction test suite generation strategy with input-output mapping supports. Scientific Research and Essays, 2011. 6(16): p. 3418-3430. <https://doi.org/10.5897/SRE11.427>
27. Nasser, A.B., et al. Assessing optimization-based strategies for t-way test suite generation: the case for flower-based strategy. in 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE). 2015. IEEE.
28. Ramli, N., R.R. Othman, and M.S.A.R. Ali. Optimizing combinatorial input-output based relations testing using Ant Colony algorithm. in 2016 3rd International Conference on Electronic Design (ICED). 2016. IEEE.
29. Mirjalili, S., SCA: a sine cosine algorithm for solving optimization problems. Knowledge-Based Systems, 2016. 96: p. 120-133. <https://doi.org/10.1016/j.knosys.2015.12.022>
30. Zamli, K.Z., et al., Fuzzy adaptive teaching learning-based optimization strategy for the problem of generating mixed strength t-way test suites. Engineering Applications of Artificial Intelligence, 2017. 59: p. 35-50.
31. Din, F. and K.Z. Zamli. Fuzzy adaptive teaching learning-based optimization strategy for pairwise testing. in 2017 7th IEEE International Conference on System Engineering and Technology (ICSET). 2017. IEEE.
32. Mahmoud, T. and B.S. Ahmed, An efficient strategy for covering array construction with fuzzy logic-based adaptive swarm optimization for software testing use. Expert Systems with Applications, 2015. 42(22): p. 8753-8765. <https://doi.org/10.1016/j.eswa.2015.07.029>