Zhongguo Kuangye Daxue Xuebao
Journal of China University of Mining and Technology

2024 | Vol 29 | Issue 4 | Page 1-21
Journal Homepage: https://zkdx.ch/
DOI: 10.1654/zkdx.2024.29.4-1

Combinatorial Interaction Testing for T-Way Test Case Generation: A Scoping Review of the Perspective Features

Aminu Aminu Muazu*
Computer & Information Sciences Department, Faculty of Science and Information Technology, Universiti Teknologi PETRONAS, Malaysia;
Computer Sciences Department, Faculty of Natural and Applied Science Umaru Musa Yar'adua University Katsina, Nigeria
*Corresponding author

Ahmad Sobri Hashim
Computer & Information Sciences Department, Faculty of Science and Information Technology, Universiti Teknologi PETRONAS, Malaysia

Umar Ismaila Audi
Computer & Information Sciences Department, Faculty of Science and Information Technology, Universiti Teknologi PETRONAS, Malaysia

Yusuf Aliyu
Computer & Information Sciences Department, Faculty of Science and Information Technology, Universiti Teknologi PETRONAS, Malaysia

Muhammad Sabo Yahaya
Computer & Information Sciences Department, Faculty of Science and Information Technology, Universiti Teknologi PETRONAS, Malaysia

Abstract
Combinatorial t-way testing techniques aim to identify faults that arise from interactions among system components. Test case generation is a prominent area within combinatorial t-way testing, presenting challenges due to its classification as a non-deterministic polynomial-time hardness (NP-Hard) problem. Numerous t-way strategies have been proposed in the literature to generate optimal test data. While some of these strategies are optimization-based and focus on factors such as uniformity, variability, and input-output interaction strength. This paper presents a scoping review that will assess and evaluate the perspective features of the existing combinatorial t-way testing strategies from 2013 to 2023. More so, we describe t-way testing techniques, analyze existing literature, and suggest future research directions. The objective is to provide a valuable resource for researchers and practitioners involved in combinatorial t-way testing. Additionally, we present a quantitative assessment that includes an evaluation of combinatorial t-way testing strategies' literature-based, approach-based, interaction-based characteristics, support-based, and search-based methods. Finally, we proposed potential possibilities for further exploration of combinatorial t-way testing.

Keywords
Combinatorial interaction testing, T-way technique, Exhaustive testing, Test case generation, Final test suite

1. Introduction
Information and communication technology is anticipated to foster sustainable global development due to its rising impact [1]. Software testing is crucial in the software development life cycle that executes a program or system purposely to find errors [2], [3]. Software testing is far from simple and is characterized by various complexities. These complexities span across a wide range of testing types, numerous tools used to facilitate testing, and the challenges associated with selecting the right test cases. Software testing can be costly in terms of human effort, or the technology required

enhancing the effectiveness of human effort. Recent research shows that testing normally takes closer to 50% of the total development time [4]. As such, many different testing strategies are employed in a wide range of applications for software testing.

Generally, software testing methods are classified into two categories: static testing and dynamic testing. Static testing does not involve executing the code and includes techniques like inspection, walkthrough, and code review, aiming to identify errors early in development. Dynamic testing involves executing the code and analyzing output values based on input values, classified into white box testing (focused on internal structure), black box testing (focused on input-output behavior), and grey box testing (limited knowledge about code) [5]. White box testing evaluates code structure through methods like statement, branch, and path testing. Black box testing prioritizes finding deviations from specified behavior using techniques like equivalence partitioning, boundary value analysis, and combinatorial testing [6]. Combinatorial testing, a black box technique, is effective in creating test cases by combining parameter values to enhance system reliability and reduce failures caused by interactions between parameters [7], [8].

According to [9], the objective of combinatorial testing is to detect faults that may arise in a software configuration system by utilizing a concise test suite that covers all feasible parameter values and their combinations. A test case is a set of conditions used in software testing to check if a particular software configuration system functions as intended [6]. In the realm of combinatorial testing, test cases are created by combining various parameter values. This approach aims to minimize system failures and enhance overall system reliability. Exhaustive testing aims to test all possible parameter value combinations to ensure that the product cannot be destroyed. Most researchers agree that employing exhaustive testing is not practical and combinatorial testing can overwhelm the problem of exhaustive testing since software failures are detected when interaction happens between values of parameters [10], [11].

The combinatorial testing approach known as the t-way technique utilizes the concept of interaction strength, represented by the variable 't' [12]. For instance, when ‘t’ is set to 2, it involves testing combinations of two parameters within the system under testing. The range of 't' can be adjusted from 2 to encompass all parameters in the system. By systematically generating test cases using t-way testing, all essential interaction elements are covered at least once, leading to a reduction in test suite size proportional to the interaction strength represented by 't'. Research findings indicate that faults in software systems are often triggered by two-parameter interactions, some of which may result from higher-order interactions [13], [14]. Thus, t-way techniques create a concise test suite that maximizes interaction tuple coverage by systematically ensuring that each test case covers the maximum number of interaction tuples based on t-way coverage. Combinatorial t-way testing techniques come in three types: uniform interaction strength, variable interaction strength, and input-output-based relations.

Certainly, several surveys have been conducted in the past decade such as Othman et al. [15] that focused on computational implementation, supported interaction, strategy approach, automation support and deployment; Alsewari and Zamli [16] focus on those based on Simulated Annealing, Genetic Algorithms, Ant Colony Algorithms, Particle Swarm Optimization, and Harmony Search; Khalsa and Labiche [17] focused on support for selection criteria, mixed covering arrays, coverage strength, and support for constraints among parameters; Alsewari et al. [18] that focus on supporting input-output features based on nature-inspired algorithms; Mudarakola and Padmaja [19] search mechanism (such as Particle Swarm Optimization, Genetic Algorithm, Ant Colony Algorithm, Simulated Annealing, and Bee Colony Optimization); Fadhil et al. [20] conducted a comprehensive systematic review focused on various criteria, such as generation technology, supported interactions, test strategy method, mixed coverage, and support for parameter constraints. Additionally, Alazzawi et al. [21] conducted a comprehensive survey focusing on some selected test case generation strategies. Recently, La chance et al. [22], conducted a review only focus on application of combinatorial testing in a distributed computing. While the literature has surveyed and examined state-of-the-art t-way test suite generation strategies during the period of their studies, they have not accounted for recent advancements, particularly some combinatorial prospective feature combination such as literature-based approach-based, t-way interaction-based, and support-based (seeding and constraint). Moreover, they have overlooked the application of newer search methods such as metaheuristic methods, hybridization methods, and hyper heuristic methods. Consequently, the goal of this research is to enhance previous reviews and surveys by integrating and giving emphasis to a select few newly developed t-way strategies.

One of the most recent challenges in optimization is the combinatorial explosion problem since it is considered NP-hard. Muazu et al. [23] states that numerous algorithms have been created to tackle ongoing challenges in the combinatorial explosion, and metaheuristics are an effective optimization approach. Originally introduced by Glover in 1986 as “modern heuristics”, the term later changed to “metaheuristics” [24]. The name comes from the Greek words “meta” and “heuristic”, meaning “beyond, in an upper level” and “to find”, respectively. Ant Colony Optimization, Genetic Algorithms, Evolutionary Computation, Tabu Search, Simulated Annealing, and Harmony Search Algorithm are all part of the metaheuristic class of algorithms. Eventually, many of these metaheuristic algorithms have been employed to solve various optimization problems, including scheduling problems [25], X-ray segmentation problems [26], feature selection problems [27], skin lesions problems [28], and so on. Similarly, some combinatorial t-way strategies employed metaheuristic methods to generate test cases to overcome the standing problem. Examples of metaheuristic-based t-way strategies are HGHC [29], ABCVC [30], GS [8], and so on by utilizing the Hill Climbing algorithm, Artificial Bee Colony, and Genetic algorithm, respectively.

This study is primarily motivated by the goal of evaluating and exploring existing combinatorial t-way strategies for addressing optimization problems. Additionally, it seeks to investigate aspects that might have been overlooked in prior research, with the aim of uncovering opportunities for improvements in the field. Through a literature review and a detailed methodology, this review offers fresh perspectives on some key features of combinatorial t-way testing, with the intention of inspiring further analysis and discussions, while considering ways to enhance future studies in the same domain. While some of the existing strategies exhibit impressive results, practical implementation can be challenging due to their differing support requirements. By providing insights into their potential contributions and limitations, we aim to offer a realistic outlook on their prospective features.

An entire aspect of our work, which we plan to expand upon in future research, is the application of metaheuristic, hybridization, or hyper-heuristic methods with constraint and seeding support in combinatorial t-way techniques for upcoming strategies. The objective is to enhance software quality and generate an efficient test suite. Therefore, this survey paper will provide an assessment and evaluation of existing combinatorial t-way strategies from 2013 to 2023. Additionally, it will present an analysis of the performance of these strategies with their challenges, identify research gaps, and propose directions for future research.

To present the information clearly and logically, this paper is structured as follows: Section 2 introduces the research's search method and selection criteria; Section 3 gives an overview of combinatorial t-way testing; Section 4 reviews existing strategies in combinatorial t-way testing literature; Section 5 discusses the evaluation of the strategies' analysis, challenge, and limitation with future recommendations; and finally, Section 6 present the conclusion.

2. Search Method and Selection Criteria

In this aspect, we intend to classify the studies based on the perspective features of combinatorial t-way testing such as literature-based, approach-based, interaction-based, support-based, and search-based. Moreover, we performed a literature review by following the structured Preferred Reporting Items for the Systematic Review and Meta-Analysis (PRISMA) framework outlined in [31]. This method involved several key steps, beginning with the formulation of specific research questions. We then identified suitable search engines and carefully defined relevant search terms to ensure a thorough exploration of existing literature. We then established specific criteria to determine which articles would be included in our review and which would be excluded. These criteria were crucial in maintaining the relevance and quality of the articles selected for analysis. By adhering to these guidelines, we were able to systematically assess and synthesize the existing combinatorial t-way testing strategies accordingly.

2.1 Research Questions

Conducting a literature review to evaluate the current state of three distinct research questions is formulated as follows:

Question 1-What are the combinatorial t-way testing techniques?

Question 2-What are the existing combinatorial t-way testing strategies?

Question 3-How can the combinatorial perspective features of the existing combinatorial t-way testing strategies be evaluated?

Conducting a literature review to analyze the current state-of-the-art of the existing combinatorial t-way testing strategies and to propose potential future directions is essential. The first research question will be addressed in Section 3 by reviewing the three possible t-way techniques, while the second research question will be explored in Section 4 through a comprehensive review and analysis of all existing combinatorial t-way testing strategies found in the literature. Furthermore, we will identify limitations and demonstrate that combining multiple support mechanisms enhances software quality with an optimal test suite. On the other hand, the third research question will be explored in the results and discussion in Section 5 through various quantitative assessments that include the strategy's year of publication, approached-based, search-based, interaction-based, and support-based.

2.2 Search Engines & Search Terms

In this subsection, we outline the search engines and search terms utilized during the literature review. We carried out and organized our research within the period of years from 2013 to 2023 across five primary platforms commonly used by researchers: Web of Science, Google Scholar, Springer, ResearchGate, and IEEE Explore. The specifics of our initial searches on these platforms are presented in Table 1, which demonstrates that most of the articles were sourced from ResearchGate, followed by Google Scholar, Web of Science, IEEE Explore, and Springer. Figure 1 illustrates the compilation of 58 articles that were identified, retrieved, screened, and included from the mentioned platforms for use in the study's analysis.

Table 1 Search Detail

Platform	No of Articles
Web of Science	48
Google Scholar	81
ResearchGate	89
IEEE Explore	42
Springer	22
Total	282

Additionally, specific search terms were employed on each platform to locate pertinent studies aligned with the previously mentioned keywords. The utilized search terms are as follows: "*combinatorial testing*", "*testing strategy*", "*t-way testing*", "*test case generation*", "*final test suite*", and "*covering array*". It's worth noting that a total of 282 articles were identified and retrieved from the mentioned platforms for use in the study, although some were discovered to fall outside the scope of the research that include test suite generation in areas like fairness testing [32], big data [33], software product line [12], software defect [34], network security [35], and self-adaptive system [36].

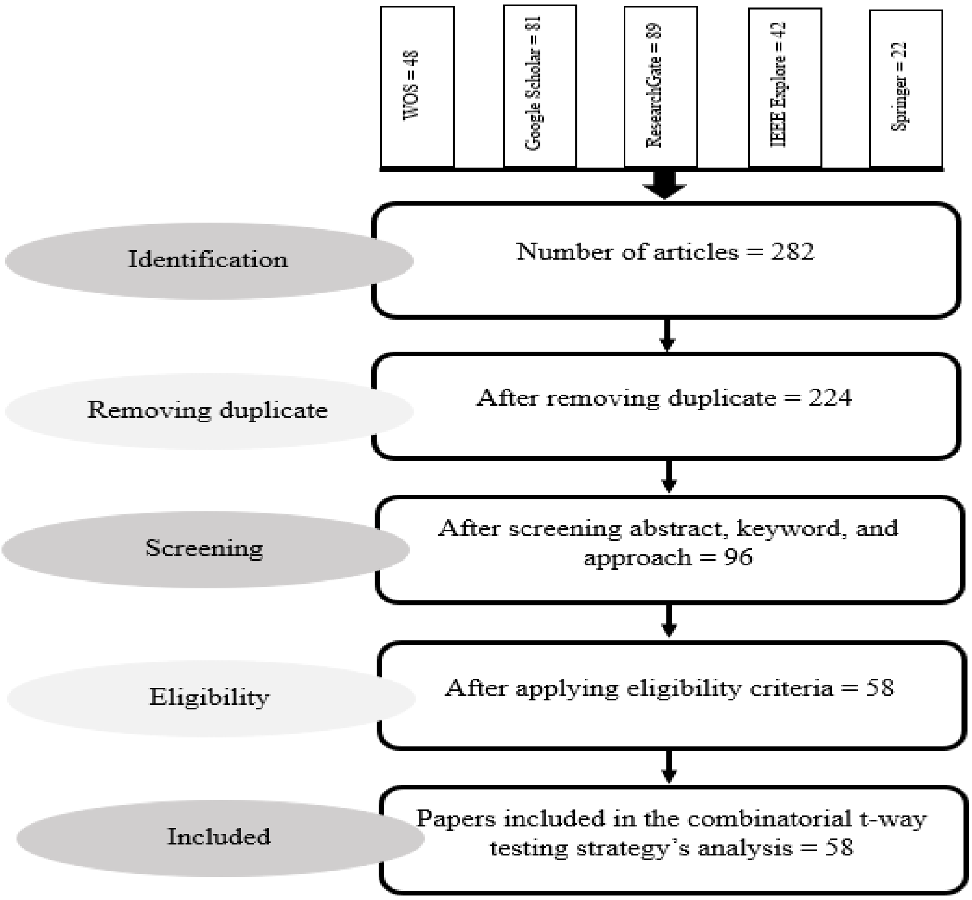

Fig. 1 PRISMA flow diagram for the Paper Selection Process

2.3 Criteria for Inclusion and Exclusion of Articles

Numerous articles were located by employing the search terms mentioned earlier within the search engine. After a comprehensive review of the search results, a thoughtful choice was made to select a total of 58 combinatorial t-way testing strategies articles as shown from Figure 1. This selection process was carried out following the specific criteria outlined in Table 2, which acted as a guideline for inclusion. It's worth highlighting that the authors undertook the task of meticulously sifting through the articles, purposefully excluding those that were not directly relevant to the research focus. The objective was to guarantee the inclusion of articles where the search terms were evident in their titles or abstracts, thus prioritizing those with the utmost relevance and significance for integration into the study.

Table 2 Selection Process for Inclusion and Exclusion of Articles

Criteria	Conditions
Inclusion	The requirement is that the content must be composed in the English language.
	The material must be published in a journal, conference, or lecture note.
	It should concentrate on combinatorial t-way testing.
	The published t-way strategies should be between 2013 to 2023.
Exclusion	Publications that are not written in the English language.
	Articles that are duplicated across multiple platforms.
	Theses, dissertations, and final year project reports that have been published.
	Textbooks

3. Combinatorial T-Way Testing Techniques

Combinatorial t-way testing is a standard for testing software configurations, which requires that every combination of parameter values for each t-way combination of input parameters must be covered by at least one test case based on its specifications [6], [37]. Over 10 years, combinatorial t-way techniques have attracted much attention for constructing a minimal test case which is an NP-hard problem [38]. By treating combinatorial interaction testing as an optimization problem, researchers have focused on using optimization methods to improve t-way strategies [39], [40], [41].

According to [42], all t-way strategies are classified into three search-based categories: either algebraic-based, computational-based, or metaheuristic-based strategies. In the algebraic method, mathematical functions are employed on the t-way strategies to produce test cases. However, a drawback of the algebraic approach is that the computations involved are typically lightweight; they are only applicable to small configuration systems; they impose restrictions on the strength of interactions, despite producing optimal test suites, it achieved the fastest execution time. This limitation hinders the applicability of the algebraic approach to t-way strategies [43]. An illustration of a strategy based on algebraic principles is the use of Orthogonal Array (OA). In the computational approach, all limitations associated with the algebraic approach are eliminated. However, this approach can be expensive due to considering the entire combination space [43]. The Automatic Efficient Test Generator (AETG) serves as a representative example of a strategy based on computational approaches. Recently, the emerge of metaheuristic approach overcomes the interaction testing problem [23]. An illustration of a strategy based on metaheuristics is Cuckoo Search (CS).

Furthermore, when constructing a test case, literature [4], [29] has categorized t-way strategies into two primary approaches: the One-test-at-a-time (OTAT) approach and the One-parameter-at-a-time (OPAT) approach. With OTAT, an empty test suite is initially created, and test cases are added one by one until all are covered. Whenever a test case is chosen, it is included in the final suite (vertical extension). In contrast, OPAT starts with an initial test suite and gradually includes one parameter at a time until all are covered (horizontal extension). If any test cases are missing after this step, they are added in the vertical extension to ensure maximum interaction coverage. Figure 2 presents the organization of combinatorial interaction t-way testing strategies of all the mentioned features (search-based, approach-based, support-based, and interaction-based). Nevertheless, t-way testing techniques are classified into three forms: uniform interaction strength, variable interaction strength, and input-output-based relations.

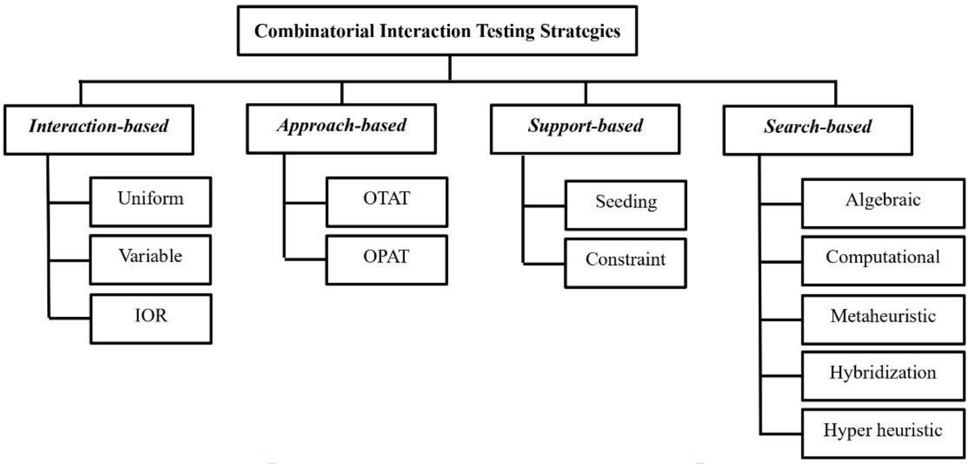

Fig. 2 Illustration of Combinatorial Interaction Testing Strategies Organization

3.1 Uniform Interaction Strength

The uniform interaction strength is a method used in combinatorial t-way testing to combine parameter values based on a consistent level of interaction strength. This method ensures that each t-way combination of input parameters in a software configuration system has the same level of interaction strength [44]. This means that each test case covers the same number of interaction tuples. The number of interaction strengths represents the maximum count of parameters that can be involved in an interaction. For instance, if t=2 and the number of parameters is 4, then the maximum interaction strength will be 2. This means that the uniform interaction strength will combine the values of parameters in pairs to ensure that each combination of values is covered by at least one test case [6], [42]. Moreover, uniform interaction strength handles two mathematical notations: Covering Array (CA) and Mixed Covering Array (MCA). The CA notation is represented as CA (N, t, v^p). Where the variables N, t, v, and p refer to the final test suite size, strength interaction, number of values, and number of parameters of a system under test, respectively.

3.2 Variable Interaction Strength

The term "variable interaction strength" refers to a measure that considers multiple levels of interaction intensity between different variables [45]. This measure can be adjusted to account for different subsets of parameters, meaning that it can be customized to capture the interactions of interest within a given system or model. By considering these various levels of interaction intensity, the variable interaction strength measure provides a more nuanced understanding of how different variables affect one another and can help identify important patterns or relationships within a complex system. Moreover, the concept of variable interaction strength can be utilized in testing any software configuration system or a system that requires executing multiple configurations [6], [42]. Unlike uniform interaction, the variable interaction strength deals with Variable Covering Array (VCA) notation which can be represented as VCA (N, t, G1, G2). In this context, N and t represent the size of the final test suite and the strength of interaction, respectively. G1 and G2 denote the number of values and various parameters of a system under test, which can be either CA and/or MCA, allowing it to accommodate different strengths of interaction.

3.3 Input-Out Based Relation

The input-output-based relation refers to the connection between the inputs and outputs of a system, and the combination of parameters that affect a specific output. In many systems or models, there can be multiple inputs that can influence one or more outputs, and the nature of the relationship between these inputs and outputs can be complex. The input-output-based relation can help to simplify this complexity by identifying the specific parameters or combination of parameters that have the greatest impact on a given output. The reason for introducing it was to avoid duplication of test cases, as not all software configuration systems possess identical features [6], [42]. The input-output-based relation utilizes the VCA notation, which can be expressed as IOR (N, $\{A_1, A_2, \ldots, A_r\}$, v_1^{p1}, v_2^{p2}, …, v_z^{pz}). N, v, and p have the same meanings as in VCA. However, in this case, the symbols A represents multiple sets of parameters, and these sets collectively determine the relationships contributing to the outputs. It's worth noting that these parameter sets in K can be indexed from 0, 1, 2, and so forth, up to z-1.

3.4 Combinatorial Interaction t-way Running Example

From a mathematical perspective, the Covering Array (CA) notation CA (N; t, v^p) involves the utilization of parameters N, t, p, and v. These parameters represent the test size, interaction strength, the number of parameters involved, and the requirement for uniform values, respectively. Let's examine a basic software system represented as a CA (N; 2, 2^3), where t is 2 with 3 parameters having 2 values each as shown in Figure 3, whereas the parameters and values of the system under testing are represented in Table 3. Here, the exhaustive combinations at full strength t = 3 in the final test suite will contain eight test cases as shown in Table 4. When t = 2 was selected based on Figure 3, three potential 2-way interactions were identified: PQ, PR, and QR. The ideal 2-way test suite contains only four test cases, covering all the necessary interactions. However, in this specific example, there is a 50% reduction in size compared to an exhaustive test suite.

Table 3 Parameters and Values of the System Under Testing

Parameters	P	Q	R
Values	p1	q1	r1
	p2	q2	r2

Table 4 CA (N; 2, 2^3) notation by exhaustive testing

Parameters	Test case							
	1st	2nd	3rd	4th	5th	6th	7th	8th
P	p1	p1	p1	p1	p2	p2	p2	p2
Q	q1	q1	q2	q2	q1	q1	q2	q2
R	r1	r2	r1	r2	r1	r2	r1	r2

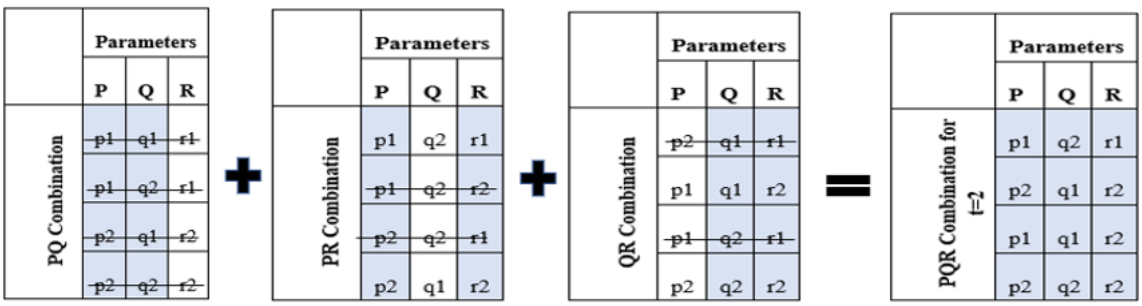

Fig. 3 Uniform interaction strength combination of CA (N; 2, 2^3)

4. Literature Review

In this section, existing combinatorial t-way strategies are further surveyed. However, these strategies are identified according to their names, authors, year of publication, interaction-based, and search-based, as well as their limitations. Furthermore, combinatorial t-way testing strategies are classified into three forms of interaction: uniform interaction, variable interaction, and input-output-based relations.

4.1 Uniform Interaction-Based Strategies

These are the t-way combinatorial strategies that ensure uniform interaction strength. The N-IPO, short for Novel IPO, is an innovative adaptation of the IPO strategy, incorporating the Fibonacci method for constraint testing [46]. N-IPO is built upon Pairwise testing and can be described as a hybrid model that combines the strengths of the IPO strategy, boundary value analysis, the Fibonacci series, and a pseudo-recursive technique. In [47] TS_OP is considered a distributed t-way testing strategy that utilizes Map and Reduce techniques on a network of workstations through Tuple Space Technology. It adopts the OPAT approach for test case generation and is designed to maintain uniform interaction strength ($t \leq 6$). When TS_OP was applied to five different environments, it yielded varying results, which led to its classification as a nondeterministic strategy. The ACTS [48] strategy merges elements of IPOG strategies for combinatorial testing to achieve a balanced trade-off between the size of the test suite and execution time. ACTS is capable of supporting t-way strengths up to 6 (where t is less than or equal to 6) while incorporating mixed-strength and constraint testing.

The BA [49] strategy employs the Bees Algorithm mechanism to create test data for uniform interaction testing, ensuring the strength does not exceed t<=10. SITG [50] draws inspiration from the Particle Swarm Intelligence algorithm's mechanism to create optimal test cases. SITG is designed to maintain a uniform interaction strength, accommodating values up to t=6. Nasser et al. [51] discusses the t-way strategy known as the Cuckoo Search Strategy (CSS), which utilizes the egg-laying behavior of cuckoo birds to generate test cases. The fundamental concept of CSS revolves around three core rules of cuckoo search algorithm: each cuckoo lays one egg at a time and deposits it in a randomly chosen nest; nests with superior egg quality are preserved for successive generations; the count of accessible host nests remains constant, and the host bird detects an egg laid by a cuckoo with a certain probability. CSS supports uniform interaction strength, where t is limited to 3, except for seeding and constraint support. In [52], the BST strategy was introduced to support uniform interaction strength up to t <= 6. BST adopts the OTAT approach of generating test cases with the bat algorithm. In subsequent work, a revised version of BST was presented in [53], which added support for constraints. However, it is important to note that this revised version only accommodates pairwise (2-way) testing.
LAHC [54] utilizes the Late Acceptance Hill Climbing algorithm to produce a t-way test suite while incorporating constraints. LAHC is tailored to ensure uniform interaction strengths with t <= 4. MTTG [55] strategy is inspired by the 'Kids Card' game, where a player randomly draws cards from a deck and aims to assemble a complete 'Set' of a specific card by sharing it with other participants. MTTG facilitates uniform interaction with increased strength when t is 12 or less. The FS strategy, introduced in reference [56], employs the Flower Pollination Algorithm mechanism as its core approach for generating uniform interaction strengths, supporting values up to t=10. PMBOS [57] is a pairwise strategy that utilizes the Migrating Birds Optimization concept to create test cases. Later, a hybrid strategy was introduced by merging the Migrating Birds Optimization with Genetic Algorithm, which is known as EMBO-GA [58]. Notably, the EMBO-GA can handle interactions of up to t-way with a maximum of 4. The HHH [59] strategy employs a hyper-heuristic approach, with Tabu Search as its high-level metaheuristic, and it harnesses the capabilities of Teaching Learning-based Optimization, Particle Swarm Optimization, Global Neighborhood Algorithm, and Cuckoo Search Algorithm as its low-level metaheuristics. HHH is designed to enable uniform interactions with a strength of $t \leq 6$.
In [60], the pairwise Artificial Bee Colony algorithm (PABC) was introduced. PABC is integrated into the Artificial Bee Colony algorithm to facilitate uniform interaction strength with t equal to 2. PCFHH is a pairwise strategy using three criteria to choose from four low-level heuristics (known as choice function) during the search process [61]. Ahmed et al. [62] introduces a novel approach for creating constrained combinatorial interaction test suites known as MOPSO. MOPSO utilizes a combination of multi-objective particle swarm optimization and multithreading to identify the best possible test cases and execute the algorithms concurrently. MOPSO is designed to facilitate uniform interaction strength, with the condition that 't' must not exceed 6. Nasser et al. [63] proposed four variations of the FPA strategy, all of which are based on the Flower Pollination Algorithm. These variations include the original FPA, hybrid elitism FPA (eFPA), hybrid mutation FPA (mFPA), and hybrid local search FPA (lFPA). The eFPA gives priority to stronger individuals and replaces weaker ones with new pollen in a random fashion. The mFPA introduces diversity through a mutation operator, and the lFPA employs an intense local search to improve local intensification. Both strategies are designed to accommodate uniform interaction strengths $t \leq 3$.

HCATS was proposed and implemented in [64] to support a uniform interaction strength (t = 2). The purpose of designing HCATS was to modify HSS [65] to incorporate the OPAT approach to generate test cases. HCATS adopts the Harmony Search Algorithm as its basis in regulating the search for both local and global solutions. The LCS strategy, short for Learning Cuckoo Search, makes use of the Cuckoo Search algorithm to generate test cases following the OTAT approach [66]. LCS is designed to handle uniform interaction strengths up to t, where t is no greater than 3. Furthermore, LCS combines this with a hybrid approach by integrating the Teaching Learning-based Optimization Algorithm. mSITG [67] relies on a swarm intelligence-based search mechanism to generate an optimal test suite. It supports uniform interaction strength with a maximum of t=6. OPAT-HS supports uniform interaction strength ($2 \geq t \leq 3$) [68]. OPAT-HS adopts OPAT based on the Harmony Search Algorithm, and it takes similar parameter settings to HCATS with interaction strength above 2. FATG [69], short for Firefly Algorithm-based Test Suite Generator, was created to speed up the execution of combinatorial t-way testing. FATG makes use of the Firefly Algorithm and accommodates uniform interaction strengths up to t, where t is restricted to 3. Later, an Adaptive Firefly Algorithm (AFA) [70] was presented, but it's important to note that it's designed specifically for pairwise testing. MOCSFO [71] employs the Crow Search Algorithm to create combinatorial t-way testing with constraint support. MOCSFO provides support for uniform interaction strengths up to t, with a maximum limit of 3.

The pATLBO_RO [72] is a pairwise strategy that utilizes Teaching Learning Based Optimization to simulate the guidance of a teacher on learners. GAPSO [73] can generate pairwise test cases by leveraging both Genetic and Particle Swarm Optimization algorithms. Alsewari et al. [74] implemented the GTHS strategy based on the standard Harmony Search Algorithm which only needs to set the parameters. GTHS adopts the OTAT approach, and it is specifically implemented to accommodate interactions with a uniform strength ($2 \geq t \leq 6$). IJA [75] improves its ability to intensify and diversify through the inclusion of new search operators like Lévy flight and mutation operators. It relies on the Jaya

Algorithm to generate optimal test cases and is designed to maintain a uniform interaction strength, accommodating values up to t=10. WOA [76] implements the Whale Optimization Algorithm for t-way testing, incorporating support for constraints, and it can handle uniform interaction strengths up to t, with a maximum limit of 6. LHS-JA [77] is built upon the enhanced Jaya Algorithm, aimed at enhancing search diversity to achieve an optimal test suite. LHS-JA accommodates uniform interaction strengths up to t=10. In [78] introduced PWiseHA which adopts the OPAT approach using the concept in Harmony Search Algorithm and supports uniform interaction strength. PWiseHA supports interaction strength up to 4, which is within the range of 2 to 4, inclusively ($2 \leq t \leq 4$).

MABCTS [79] is built upon the Modified Artificial Bee Colony algorithm for generating t-way test cases, and it offers support for uniform interaction strength when t is less than or equal to 6. FPA-HC is a pairwise strategy based on a hybrid method that combines the Flower Pollination Algorithm and Hill Climbing as the core of its search engine for generating the test suite [80]. A hybrid method developed through the integration of the Migrating Birds Optimization and Genetic Algorithm known as EMBO-GA Strategy [58]. EMBO-GA is designed to accommodate t-way interactions with a strength $t \leq 4$. PGSAS [81] is a pairwise strategy that specifically focuses on pairs of elements ($t = 2$). In contrast, the GSTG [43] strategy is designed for generating test configurations with a much broader scope of interaction coverage compared to PGSAS, accommodating higher values of t, which can go up to 10. Both PGSAS and GSTG utilize the OTAT approach, employing the Gravitational Search Algorithm. The BHA [82] strategy adopts the OTAT approach, utilizing the Black Hole Algorithm to facilitate uniform t-way interaction testing, with t<=4. AutoCCAG [83] represents a method for generating Constrained Covering Arrays, which integrates automated algorithm configuration and selection for t-way testing. It is specifically tailored to enable uniformly 5-way interactions.

BAPSO is a hybrid metaheuristic strategy that combines the strengths of the Bat Algorithm and Particle Swarm Optimization [84]. BAPSO supports uniform interaction strength, encompassing all possible test configurations with t not exceeding 6. TWGH [85] employs three metaheuristic algorithms, namely the Whale Optimization Algorithm, Gray Wolf Optimization, and Harmony Search Algorithm, to generate optimal test cases. It combines the Whale Optimization Algorithm and Gray Wolf Optimization in its exploration and exploitation mechanisms. Additionally, to enhance convergence speed, TWGH adjusts the Harmony Search Algorithm values, harmony memory considering the rate, and pitch adjustment rate. TWGH is designed to support uniform interactions, even for high-strength combinations with t values up to 12. Odili et al. [86] recently introduced four hybrid variations of the African Buffalo Optimization algorithm for t-way testing, including the Mutation Buffalo Strategy (mBS), Local-Search Buffalo Strategy (lBS), Elitism Buffalo Strategy (eBS), and Elitism Local-Search Buffalo Strategy (elBS). It's important to mention that all of these variations are designed to handle uniform interaction strength up to t=10. HGHC [29] is the most recent approach for t-way testing, combining the Greedy and Hill Climbing Algorithms to enhance test case optimization. HGHC ensures uniform interactions with a higher level of strength when $t \leq 15$. The analysis of uniform interaction-based combinatorial t-way testing strategies is presented in Table 5.

Table 5 Analysis of uniform interaction-based combinatorial t-way testing strategies

Literature		Prospective features						Limitation(s)
Strategy's Name	Reference	Approach	t-way strength	Support				
				Constraint	Seeding	Search-based		
N-IPO	[46]	OPAT	$t = 2$	✓	X	Computational		Handle one interaction strength and lack support for metaheuristic-based and seeding.
TS_OP	[47]	OPAT	$t \leq 6$	X	X	Computational		Handle one interaction's strength and lacks support for metaheuristic-based, seeding, and constraints.
ACTS	[48]	OPAT	$t \leq 6$	✓	X	Computational		Handle one interaction strength and lacks support for metaheuristic-based and seeding
BA	[49]	OTAT	$t \leq 10$	X	X	Metaheuristic		Handle one interaction's strength and lacks support for seeding and constraints.
SITG	[50]	OTAT	$t \leq 6$	X	X	Metaheuristic		Handle one interaction's strength and lacks support for seeding and constraints.
CSS	[51]	OTAT	$t \leq 3$	X	X	Metaheuristic		Handle one interaction's strength and lacks support for seeding and constraints.
BST	[52]	OTAT	$t \leq 6$	X	X	Metaheuristic		Handle one interaction strength and lacks support for seeding and constraints
LAHC	[54]	OTAT	$t \leq 4$	✓	X	Metaheuristic		Handle one interaction's strength and lacks support for seeding.

MTTG	[55]	OTAT	t ≤ 12	X	X	Computational	Handle one interaction strength and lacks support for metaheuristic-based, seeding, and constraints
FS	[56]	OTAT	t ≤ 10	X	X	Metaheuristic	Handle one interaction's strength and lacks support for seeding and constraints.
PMBOS	[57]	OTAT	t = 2	X	X	Metaheuristic	Handle one interaction's strength and lacks support for seeding and constraints.
HHH	[59]	OTAT	t ≤ 6	X	X	Hyper-heuristic-Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
PABC	[60]	OTAT	t = 2	X	X	Metaheuristic	Handle one interaction's strength and lacks support for seeding and constraints.
PCFHH	[61]	OTAT	t = 2	X	X	Hyper-heuristic-Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
MOPSO	[62]	OTAT	t ≤ 6	✓	X	Metaheuristic	Handle one interaction strength and lacks support for seeding
FPA	[63]	OTAT	t ≤ 3	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
BST	[53]	OTAT	t = 2	✓	X	Metaheuristic	Handle one interaction's strength and lacks support for seeding.
HCATS	[64]	OPAT	t = 2	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
LCS	[66]	OTAT	t ≤ 3	X	X	Hybrid-Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
mSITG	[67]	OTAT	t ≤ 6	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
OPAT-HS	[68]	OPAT	t ≤ 3	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
FATG	[69]	OTAT	t ≤ 3	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
AFA	[70]	OTAT	t = 2	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
MOCSFO	[71]	OTAT	t ≤ 3	✓	X	Metaheuristic	Handle one interaction's strength and lacks support for seeding.
pATLBO_RO	[72]	OTAT	t = 2	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
GAPSO	[73]	OTAT	t = 2	X	X	Hybrid-Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
GTHS	[74]	OTAT	t ≤ 6	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
IJA	[75]	OTAT	t ≤ 10	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
WOA	[76]	OTAT	t ≤ 6	✓	X	Metaheuristic	Handle one interaction's strength and lacks support for seeding.
LHS-JA	[77]	OTAT	t ≤ 10	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
PWiseHA	[78]	OPAT	t ≤ 4	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
MABCTS	[79]	OTAT	t ≤ 6	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints

FPA-HC	[80]	OTAT	t = 2	X	X	Hybrid-Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
EMBO-GA	[58]	OTAT	t ≤ 4	X	X	Hybrid-Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
PGSAS	[81]	OTAT	t = 2	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
GSTG	[43]	OTAT	t ≤ 10	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
BHA	[82]	OTAT	t ≤ 4	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
AutoCCAG	[83]	OTAT	t ≤ 5	✓	X	Computational	Handle one interactions strength, lacks support for metaheuristic-based, and seeding.
BAPSO	[84]	OTAT	t ≤ 6	X	X	Hybrid-Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
TWGH	[85]	OTAT	t ≤ 12	X	X	Hybrid-Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
mBS, lBS, eBS, leBS	[86]	OTAT	t ≤ 10	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
HGHC	[29]	OTAT	t ≤ 15	X	X	Hybrid-Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints

4.2 Variable Interaction-Based Strategies

These strategies encompass t-way combinations, ensuring varying levels of interaction strength. The GVS_CONST strategy is founded on a computational approach designed to facilitate variable interaction strength (t ≤ 5), while also taking constraints into account when generating test cases [87]. GVS_CONST employs a tuple tree data structure to reduce the time required for tuple generation and the process of checking uncovered scenarios. TSG [88] utilizes a multilevel Greedy Algorithm to build test suites that can handle interactions of varying strengths, with support for strengths up to t=3. ATLBO [89] employs Teaching Learning-Based Optimization, relying on the Mamdani fuzzy inference system. ATLBO offers support for variable interaction strength, with t values up to 4. GS employs a Genetic Algorithm to create test suites that accommodate interaction strengths ranging from uniform to variable, up to a maximum of 20 [8]. Additionally, GS computes the weight before generating test cases to enhance the optimization process.

In [30], the ABCVS strategy is introduced, which is designed for both uniform and variable strength test suites. ABCVS utilizes the Artificial Bee Colony algorithm as its core component to generate optimal test suites. Furthermore, ABCVS can support interaction strengths up to t ≤ 6. The VCS strategy is designed to provide variable strength support within the Cuckoo Search algorithm to optimize test case sizes [90]. However, VCS is capable of accommodating interaction strengths up to t ≤ 6. In [91], the VS-MGS strategy incorporates a Greedy Algorithm with an elitism mechanism throughout the iterations to enhance the efficiency of test case generation. VS-MGS supports variable interaction strengths up to t ≤ 6. Ramli et al. [92], the Const-TTSGA strategy was created to generate test suites that cover a range of interaction strengths, including both uniform and variable scenarios with t≤ 4. Const-TTSGA is categorized as a metaheuristic approach and employs the Ant Colony Algorithm to optimize the composition of test suites. SCAVS [93] utilizes the Sine Cosine Algorithm to generate variable interaction strengths, with support for up to t ≤ 6. The HABC strategy employs hybridization methods that harness the advantages of both the Artificial Bee Colony Algorithm and Particle Swarm Optimization to generate an optimal test suite [94]. HABC supports variable interaction with a strength of up to t ≤ 6.

The GAMIPOG strategy, as described in [95], is designed to support variable interaction strength, including higher interaction strengths up to t ≤ 5. GAMIPOG is a deterministic approach that combines the features of the Genetic Algorithm and its predecessor, the MIPOG strategy, to generate an optimal test suite. HABCSm [96] ensures both uniform and variable interactions with a strength (t) not exceeding 6. It employs the mechanisms of the Artificial Bee Colony with Particle Swarm Optimization to create test cases. TTSGA [97] utilizes the Ant Colony Algorithm to handle variable interaction strengths through three components: the VS-Tuple generator, search space generator, and test case generator. It's important to note that TTSGA is designed to work with interaction strengths up to t=6. VS-TACO [98] can produce variable interaction strengths up to t=4. It achieves this using Ant Colony Optimization and incorporates Mamdani fuzzy logic to determine the number of ants and select the most suitable search space technique. The analysis of variable interaction-based combinatorial t-way testing strategies is presented in Table 6.

Table 6 Analysis of variable interaction-based combinatorial t-way testing strategies

Literature		Prospective features					Limitation(s)
Strategy's Name	Reference	Approach	t-way strength	Support			
				Constraint	Seeding	Search-based	
GVS_CONST	[87]	OTAT	t ≤ 5	✓	X	Computational	Handle one interaction strength and lacks support for metaheuristic-based, and seeding
TSG	[88]	OTAT	t ≤ 3	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
ATLBO	[89]	OTAT	t ≤ 4	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
GS	[8]	OTAT	t ≤ 20	X	X	Metaheuristic	Handle two interactions strength and lacks support for seeding and constraints
ABCVS	[30]	OTAT	t ≤ 6	X	X	Metaheuristic	Handle two interactions strength and lacks support for seeding and constraints
VCS	[90]	OTAT	t ≤ 6	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
VS-MGS	[91]	OTAT	t ≤ 6	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
Const-TTSGA	[92]	OTAT	t ≤ 4	✓	X	Metaheuristic	Handle two interactions strength and lacks support for seeding
SCAVS	[93]	OTAT	t ≤ 6	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
HABC	[94]	OTAT	t ≤ 6	X	X	Metaheuristic	Handle two interactions strength and lacks support for seeding and constraints
GAMIPOG	[95]	OPAT	t ≤ 5	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
HABCSm	[96]	OTAT	t ≤ 6	X	X	Metaheuristic	Handle two interactions strength and lacks support for seeding and constraints
TTSGA	[97]	OTAT	t ≤ 6	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
VS-TACO	[98]	OTAT	t ≤ 4	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints

4.3 Input-Output-Based Strategies

To the best of the author's knowledge, only two strategies have been identified in the literature that support input-output-based relations, such as CTJ and AFA. The CTJ [99] is incorporated into the Jaya algorithm as a metaheuristic technique for generating a test list based on Input-Output relationships. It's important to note that CTJ is limited to handling relationships of up to 60. moreover, the AFA strategy is applied within the Adaptive Firefly algorithm to handle input-output relationships involving as many as 100 interactions [100]. AFA combines test cases relevant to the generation of t-way test suites by incorporating the elitism operator within the selected firefly algorithm. The analysis of input-output-based combinatorial t-way testing strategies is presented in Table 7.

Table 7 Analysis of input-output-based combinatorial t-way testing strategies

Literature		Prospective features					Limitation(s)
Strategy's Name	Reference	Approach	t-way strength	Support			
				Constraint	Seeding	Search-based	
CTJ	[99]	OTAT	R<=60	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints
AFA	[100]	OTAT	R<=60	X	X	Metaheuristic	Handle one interaction strength and lacks support for seeding and constraints

5. Analysis Discussion

In this section, the focus will be on presenting the principal findings of our investigation into the combinatorial interaction testing strategies for test case generation as conducted in the literature. Following this, we will discuss the challenges, strengths, and limitations of our study. Lastly, we will outline future research directions around combinatorial interaction testing.

5.1 Principal Findings

In this comprehensive scoping review, we undertake an in-depth exploration, categorization, and analysis of combinatorial interaction testing techniques. Our examination of the existing literature highlights the combinatorial perspective of these strategies, showcasing their potential applicability across several key factors, including literature-based, approach-based, interaction-based, support-based, and search-based features.

5.1.1 Analysis of Strategy's Literature-Based

The year of publication of a research paper is a critical aspect of the production of academic research. This information is typically found on the first page of a paper and provides a clear indication of the time frame in which the research was conducted. Knowing the year of publication is important for several reasons [101]. It helps other scholars determine the relevance and currency of the research. Research in combinatorial t-way testing is rapidly evolving, and new developments and discoveries can quickly render the existing strategies. However, by knowing the year of publication, other researchers can quickly determine whether the findings reported in the research work are still relevant and useful.
The authors have reviewed literature from the years 2013 to 2023 and presented it in a table format as presented earlier in Section 4. The review of combinatorial t-way strategies started in 2013, so the literature provides a comprehensive overview of the research conducted over eleven years and highlights the key findings of the strategies. The authors have taken care to ensure the accuracy of the years of publication, making it a useful resource for other researchers to stay current with the latest research and to make informed decisions about their work. Consequently, Figure 4 shows the literature analysis of detailed years of all combinatorial t-way strategies publications reviewed. The most productive years of publication are 2020, 2021, 2019, and 2018 in which they produced 21%, 15%, 14%, and 12% strategies respectively. In 2015 and 2017 each produced 9% and 7% only while 2022, 2016, 2014, and 2013 both produced 5% strategies. However, in 2023 only 2% of the strategy was produced.

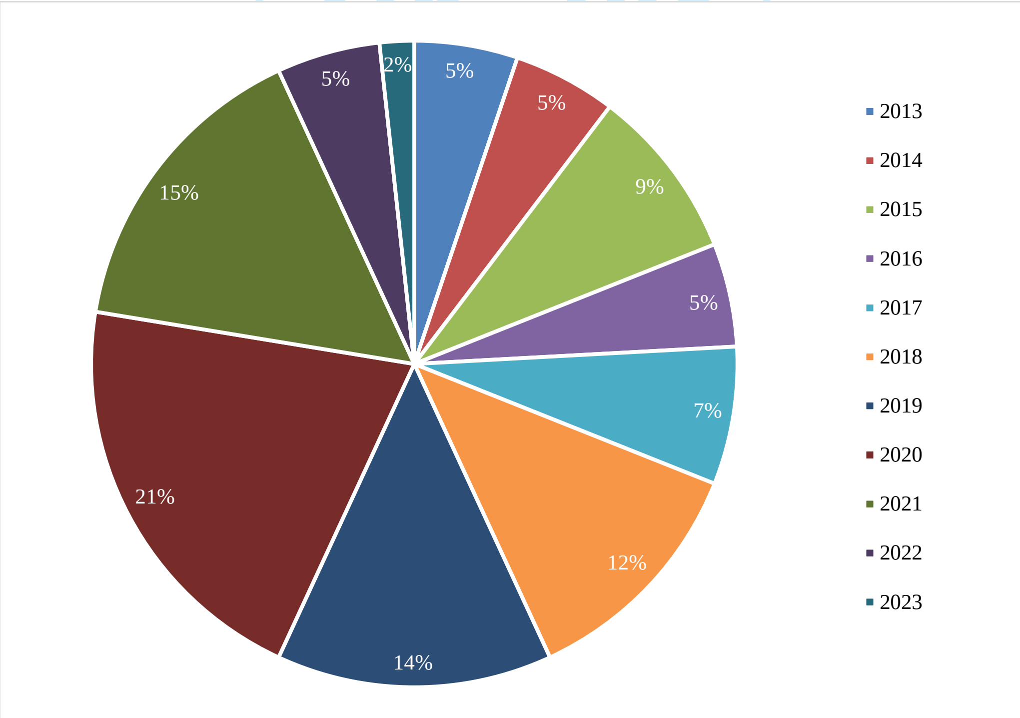

Fig. 4 Strategies Literature-Based Analysis

5.1.2 Analysis of Strategy's Approach-Based

Approach-based analysis is important in a review as it provides a structured method for creating test cases and improving the reliability of the system being tested. This approach helps to identify potential issues and bugs and reduces the time and effort required for testing. As mentioned earlier, the approach used in the combinatorial t-way strategies can be categorized into two main types: one-test-at-a-time (OTAT) or one-parameter-at-a-time (OPAT).

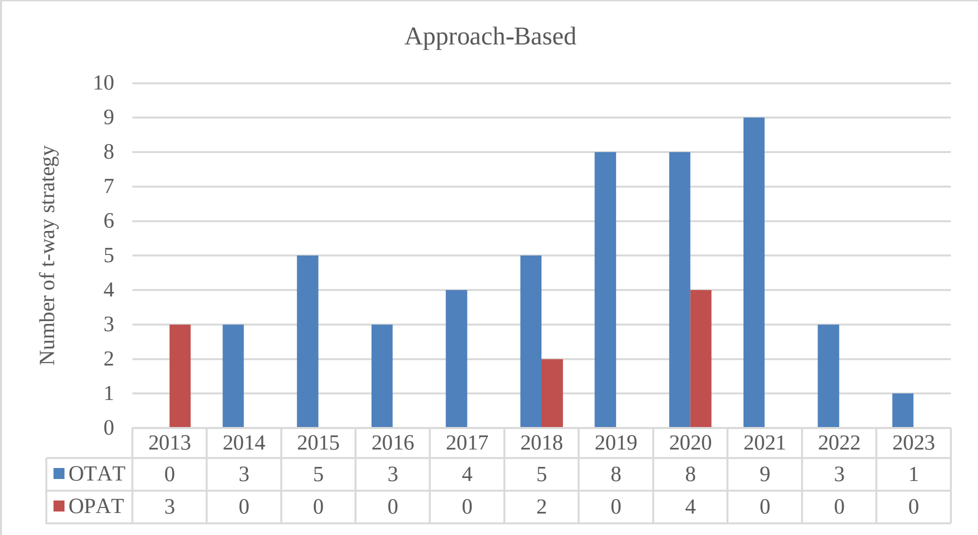

Fig. 5 Strategies Approach-Based Analysis

More specifically, Figure 5 shows that 2021, 2019, 2015, 2017, 2022, 2014, 2016, and 2023 have produced the highest number of OTAT strategies and zero OPAT. Even though, in 2020 OPAT is more competitive producing 50% of OTAT. The OPAT approach took more attention in 2020, 2013, and 2018 which have each produced four, three, and two t-way strategies respectively. Despite the research interest on the rise, it has been found that in 2013 none of the strategies were adopted for the OTAT approach. From these results, it is evident that greater attention was directed towards the OTAT approach, whereas the OPAT approach received less focus. However, the OPAT approach plays a significant role by selecting a representative value from each parameter, resulting in the generation of test cases based on these chosen values. This method yields a smaller set of test cases.

5.1.3 Analysis of Strategy's Search-Based

Analyzing the search-based aspect of a t-way strategy is significant for evaluating its effectiveness, identifying optimization opportunities, and determining its practical applicability. It is a natural phenomenon that search-based t-way strategies excel in this aspect. According to Figure 6, it is evident that metaheuristic-based approaches dominated from 2014 to 2022. Computational-based approaches were produced every year except in 2016, 2019, 2022, and 2023. Hybrid metaheuristic approaches began to emerge from 2018 to 2023. Unfortunately, hyper-heuristic approaches received less attention, being mentioned only in 2016 and 2017. These results highlighted that the hybridization of metaheuristic and hyper-heuristic methods is not frequently utilized in current t-way testing strategies. Consequently, combining two or more metaheuristic algorithms can enhance overall search capabilities by offsetting the limitations of one algorithm with the strengths of others.

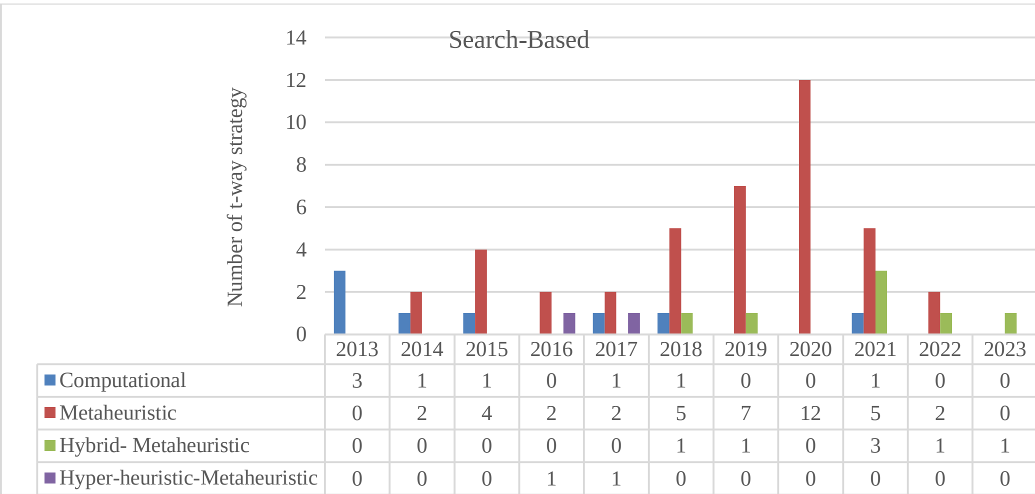

Fig. 6 Strategies Search-Based Analysis

5.1.4 Analysis of Strategy's Interaction-Based

Analyzing the interaction-based aspect of a t-way strategy is significant for evaluating coverage, reducing test cases, assessing fault detection capability, and determining adaptability to system changes. This analysis provides valuable insights into the strategy's effectiveness and suitability for different testing scenarios. As for the interaction-based analysis, knowing that only 58 combinatorial strategies were able to be identified from 2013 until 2023. Figure 7 displays the number of interactions supported by uniform, variable, and IOR for the years.

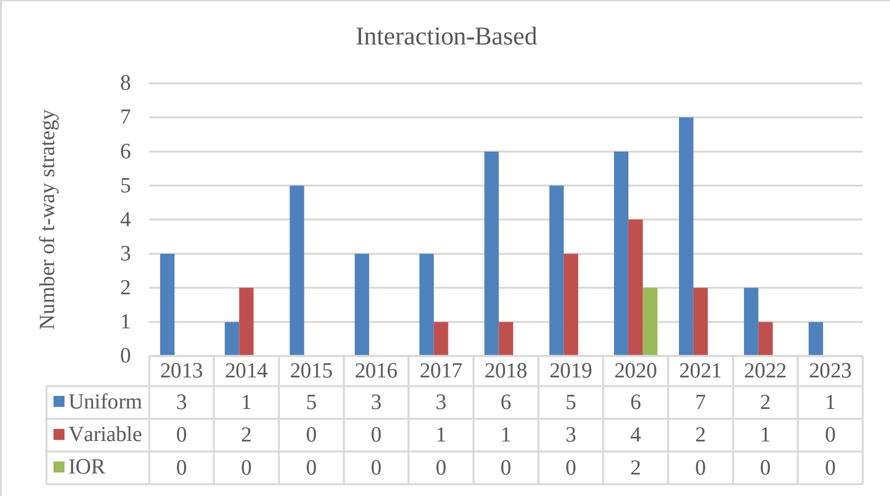

Figure 7: Strategies interaction-based analysis

Figure 7 suggests that uniform interaction strength was the most favored type of interaction over these periods. Variable interaction strength was the second-most favored, while IOR came in last. Combinatorial t-way strategies supporting uniform interaction were present each year. In contrast, variable interaction strength was absent in 2013, 2015, 2016, and 2023. Additionally, IOR-based strategies only made an appearance in 2020. Furthermore, these results indicate that insufficient attention is presently devoted to testing software configuration systems necessitating the execution of multiple configurations (i.e., variable interaction strength) and/or software configuration systems requiring the execution of specific parameters that exert the most significant influence on a given output, aiming to prevent the duplication of test cases (i.e., IOR interaction).

5.1.5 Analysis of Strategy's Support-Based

In this situation, the attention was confined to two forms of support: constraint support and seeding support. These two types of support were deemed relevant and significant in the context of t-way testing strategies. By focusing solely on these two forms of support, the researcher can keep their focus narrow and specific, leading to a deeper understanding of their influence and importance. The sample test data may include combinations that are impossible or undesirable, so constraint support is required to optimize the test suite size. Seeding support is also necessary to guarantee the use of specific combinations and to establish boundaries for the test, resulting in improved software quality.

In Figure 8, support for constraints is notably dominant in almost every year except for 2016, 2022, and 2023. However, the number of strategies supporting constraints is limited. Specifically, only 2013 and 2020 saw the presence of two strategies each, while in the remaining years, there was only one strategy each. However, it is worth noting that none of the strategies in the figure possess the feature of seeding support. This indicates that there is no possibility to initiate the system with a predetermined starting state, which is also a limitation of the existing t-way strategies, e-ven though, in 2012 there is a popular computational t-way strategy known as TTG which supports both seeding and constraint [102].

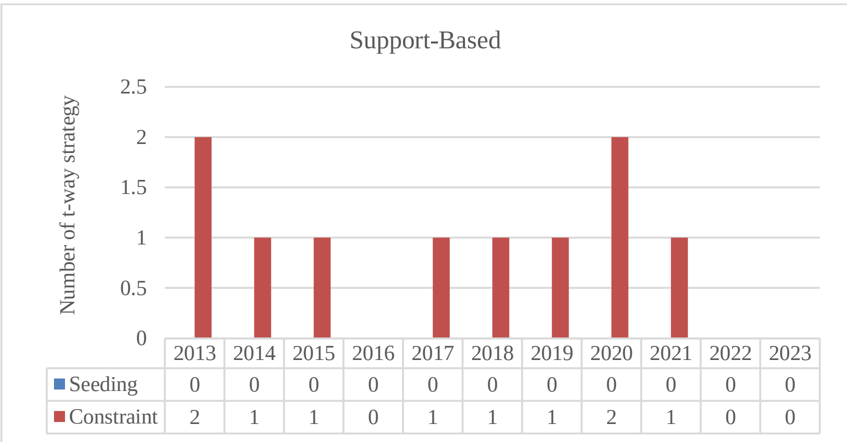

Figure 8: Strategies Support-Based Analysis

5.2 Challenges, Strengths and Limitations

Notably, we thoroughly identify 58 existing t-way strategies within this period; two main approaches for generating test cases, two support methods for enhancing software quality and reducing test case size, three types of interaction support, and four primary search methods for the effectiveness of test case generation. This scoping review contributes a carefully documented classification and a rigorous reviewing framework, facilitating the evaluation of existing combinatorial interaction testing strategies for test case generation.

In terms of challenges encountered in combinatorial interaction testing, our findings suggest that generating test cases should incorporate not only traditional t-way testing methods but also advanced and non-functional techniques. Researchers' efforts assist testers in ensuring data quality and facilitating the creation of efficient test cases and automation tools. These tools validate that each combination of parameter values for every t-way combination is addressed by at least one test case, ultimately producing an optimal test suite. Table 8 outlines the challenges in combinatorial interaction testing, providing descriptions and proposed solutions based on research findings.

Table 8 Combinatorial interaction testing challenges and proposed solutions

Challenge	Description	Proposed solution
Interaction Coverage	Ensuring all necessary interactions are covered without an explosion in the number of test cases.	T-way techniques can dynamically adjust the level of interaction coverage based on testing requirements and resource availability.
Scalability	With the increase in the number of parameters and values, the number of possible combinations grows exponentially, making it challenging to generate and execute all possible test cases within a reasonable time frame and with available resources.	Use incremental approaches where testing starts with a small number of parameters and gradually includes more, reducing the complexity at each step. For instance, adopting OTAT or OPAT approach in generating test cases.
Test Case Minimization	Finding the smallest possible set of test cases that still provides maximum coverage of the interactions is a difficult optimization problem.	Minimizing the number of test cases while ensuring comprehensive testing requires sophisticated algorithms. Methods such metaheuristic-based, hybridization, or even hyper heuristic method.
Handling Constraint	The selected test data might encompass undesirable or impractical combinations of parameter values.	There is a need for adherence to constraint support. Consequently, this approach provides a software tester with the advantage of excluding certain test cases when warranted.
Handling Seeding	Certain combinations of parameter values must be incorporated in the testing processes.	There is a need for adherence to seeding support. Consequently, this approach provides a software tester with the advantage to incorporating necessary test cases into the final test list.
Multiple Interaction Methods	Combining more than one interaction method to create a comprehensive testing strategy for testing different configurations.	Integration of uniform and variable interaction will help for testing CA, MCA, and VCA notations. Integration of uniform, variable, and IOR interaction will help for testing CA, MCA, VCA, IOR notations.
Tool Support	There is a need for vigorous and user-friendly tools that can support the various aspects of combinatorial interaction testing.	Combinatorial interaction test data generator, also known as t-wat testing strategy.
Test Case Redundancy	Redundant test cases can waste resources and time without adding new information or coverage.	Use optimization techniques such metaheuristics to minimize the test suite while maximizing coverage, ensuring no redundant test cases.
High Computational Cost	Creating and checking every combination gets costly, especially with more parameters and values.	Implement metaheuristics or hyper heuristic methods.

In terms of the strengths and limitations of our research, our conducted study offers a comprehensive overview of the current state of research on existing combinatorial t-way testing strategies, with a focus on perspective features. However, it's important to acknowledge that conducting a scoping review involves a predominantly manual process, which introduces the possibility of missing some relevant studies. To mitigate this risk, we closely adhered to the PRISMA guidelines, widely recognized as a leading framework for conducting and reporting high-quality reviews.

Additionally, our search for relevant terms was limited to five strong search terms such as combinatorial testing, testing strategy, t-way testing, test case generation, final test suite, and covering array, aiming to yield desired results. We believe these approaches contribute to the study's fullness. On the other hand, like any chosen research method, ours has inherent limitations. This research focused primarily on popular platforms such as Web of Science, Google Scholar, Springer, ResearchGate, and IEEE Explore to search for and extract relevant primary studies. While these platforms are reputable, it's possible that other platforms could also offer relevant studies. However, we selected these platforms based on their established reputations.

5.3 Future Research Directions

The area of combinatorial t-way testing strategies is still relatively new and constantly evolving with innovative ideas and applications. These are some potential research directions in this exploration:

- Simultaneous use of constraint and seeding support to reduce the number of generated test data and improved the software quality. Integrating both seeding and constraint support within either OTAT or OPAT approach is crucial. Particularly, the OPAT approach as it involves selecting a representative value from each parameter and generating test cases based on these chosen values. Consequently, the method will yield a smaller set of test cases, assuming that managing the risk of interaction among non-representative components while ensuring the completion of system testing using representative values remains achievable within a reasonable budget.

- Implementing metaheuristic hybridization or hyper heuristic methods to enhance either/both uniform, variable or/and IOR interaction by incorporating seeding and support.
- The Combinatorial Interaction Test Data Generator, also known as the t-way testing strategy, supports all types of interactions, including uniform, variable, and IOR.

6. Conclusion

In summary, this paper provides an overview of the current state-of-the-art of existing combinatorial t-way strategies, emphasizing the strengths and limitations of each strategy. To achieve our goals, we have identified 58 existing combinatorial t-way testing strategies within 2013 to 2023 which are evaluated into various perceptions that include the literature-based, approach-based, search-based, interaction-based, and support-based. In the literature-based, the most productive years of publication are 2020, 2021, 2019, and 2018. In the approach-based category, OTAT is dominant and receives more attention than OPAT. In the search-based category, most strategies adopt metaheuristics methods, and then computational methods, however, less attention is given to hybridization and hyper-heuristic methods. In the interaction-based category, uniform interaction is the most prominent strategy, and then variable t-way strategy, unfortunately, only two of them support IOR. In the support-based category, less consideration is given, only ten strategies support constraints, but none of them supports both or seeding.

Acknowledgement

The authors express gratitude for the support of this research provided by the Ministry of Higher Education (MoHE) Malaysia under the Fundamental Research Grant Scheme FRGS/1/2023/ICT01/UTP/02/2.

Reference

1. E. B. Ogunwole, J. A. Asaleye, M. I. Tabash, A. Ahmed, Y. Elsantil, and A. I. Lawal, "Debt service and information communication technology on employment and productivity: Short- and long-run implications," *Sci Afr*, vol. 24, p. e02227, Jun. 2024, doi: 10.1016/J.SCIAF.2024.E02227.
2. H. Mamman, S. Basri, A. O. Balogun, A. A. Imam, G. Kumar, and L. F. Capretz, "Search-Based Fairness Testing: An Overview," in *2023 IEEE International Conference on Computing, ICOCO 2023*, 2023. doi: 10.1109/ICOCO59262.2023.10397906.
3. Z. Sun, C. Hu, C. Li, and L. Wu, "Domain ontology construction and evaluation for the entire process of software testing," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3037188.
4. A. Aminu Muazu, A. Sobri Hashim, A. Sarlan, and M. Abdullahi, "SCIPOG: Seeding and constraint support in IPOG strategy for combinatorial t-way testing to generate optimum test cases," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 1, pp. 185–201, Jan. 2023, doi: 10.1016/J.JKSUCI.2022.11.010.
5. N. Anwar and S. Kar, "Review Paper on Various Software Testing Techniques & Strategies," *Global Journal of Computer Science and Technology.*, vol. 19, no. 2(1.0), 2019.
6. A. Aminu Muazu, A. Sobri Hashim, and A. Sarlan, "Application and Adjustment of 'don't care' Values in t-way Testing Techniques for Generating an Optimal Test Suite," *Journal of Advances in Information Technology*, vol. 13, no. 4, pp. 347–357, 2022, doi: 10.12720/jait.13.4.347-357.
7. A. A. Muazu, A. S. Hashim, A. Sarlan, and U. D. Maiwada, "Proposed Method of Seeding and Constraint in One-Parameter-At-a- Time Approach for t-way Testing," in *2022 International Conference on Digital Transformation and Intelligence (ICDI)*, IEEE, Dec. 2022, pp. 39–45. doi: 10.1109/ICDI57181.2022.10007210.
8. S. Esfandyari and V. Rafe, "A tuned version of genetic algorithm for efficient test suite generation in interactive t-way testing strategy," *Inf Softw Technol*, vol. 94, 2018, doi: 10.1016/j.infsof.2017.10.007.
9. E. Pira, V. Rafe, and S. Esfandyari, "A three-phase approach to improve the functionality of t-way strategy," *Soft comput*, 2023, doi: 10.1007/s00500-023-08199-5.
10. K. M. Htay, R. R. Othman, and A. Amir, "Utilization of Gravitational Search Algorithm for Combinatorial T-Way Testing," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1755/1/012007.
11. A. Aminu Muazu and A. Aminu Muazu, "One-parameter-at-a-time combinatorial testing strategy based on harmony search algorithm supporting mixed covering array mathematical notation (OPATHS)," in *1st International Conference on Information Technology in Education & Development (ITED)*, Information Technology in Education & Development (ITED), 2018, pp. 64–70.
12. M. A. Jamil, M. K. Nour, S. S. Alotaibi, M. J. Hussain, S. M. Hussaini, and A. Naseer, "Software Product Line Maintenance Using Multi-Objective Optimization Techniques," *Applied Sciences (Switzerland)*, vol. 13, no. 15, Aug. 2023, doi: 10.3390/app13159010.

13. A. O. Balogun *et al.*, "Empirical Analysis of Data Sampling-Based Ensemble Methods in Software Defect Prediction," in *In book: Computational Science and Its Applications – ICCSA 2022 Workshops, Malaga, Spain.*, 2022, pp. 363–379. doi: 10.1007/978-3-031-10548-7_27.
14. S. Böhm, S. Krieter, T. Heß, T. Thüm, and M. Lochau, "Incremental Identification of T-Wise Feature Interactions," in *ACM International Conference Proceeding Series*, 2024. doi: 10.1145/3634713.3634715.
15. R. R. Othman, K. Z. Zamli, and S. M. S. Mohamad, "T-way testing strategies: A critical survey and analysis," *International Journal of Digital Content Technology and its Applications*, vol. 7, no. 9, p. 222, 2013.
16. A. A. Al-Sewari 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.*, Springer Verlag, 2014, pp. 255–263. doi: 10.1007/978-981-4585-42-2_30.
17. S. K. Khalsa and Y. Labiche, "An Orchestrated Survey of Available Algorithms and Tools for Combinatorial Testing," in *2014 IEEE 25th International Symposium on Software Reliability Engineering*, 2014, pp. 323–334. doi: 10.1109/ISSRE.2014.15.
18. A. A. Alsewari, N. M. Tairan, and K. Z. Zamli, "Survey on Input Output Relation Based Combination Test Data Generation Strategies," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 18, 2015, [Online]. Available: www.arpnjournals.com
19. L. P. Mudarakola and M. Padmaja, "The survey on artificial life techniques for generating the test cases for combinatorial testing," *International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)*, vol. 2, no. 6, pp. 19–26, 2015.
20. H. M. Fadhil, M. N. Abdullah, and M. I. Younis, "Combinatorial Testing Approaches: A Systematic Review," *IRAQI JOURNAL OF COMPUTERS, COMMUNICATIONS, CONTROL AND SYSTEMS ENGINEERING*, vol. 22, no. 4, pp. 60–79, 2022, doi: https://doi.org/10.33103/uot.ijccce.22.4.6.
21. A. K. Alazzawi *et al.*, "Recent t-way Test Generation Strategies Based on Optimization Algorithms: An Orchestrated Survey," in *International Conference on Artificial Intelligence for Smart Community: AISC 2020, 17–18 December, Universiti Teknologi Petronas, Malaysia*, Springer, 2022, pp. 1055–1060.
22. E. La Chance and S. Hallé, "An investigation of distributed computing for combinatorial testing," *Software Testing Verification and Reliability*, vol. 33, no. 4, 2023, doi: 10.1002/stvr.1842.
23. A. A. Muazu, A. S. Hashim, and A. Sarlan, "Review of Nature Inspired Metaheuristic Algorithm Selection for Combinatorial t-Way Testing," *IEEE Access*, vol. 10, pp. 27404–27431, 2022, doi: 10.1109/ACCESS.2022.3157400.
24. F. Glover, "Artificial intelligence, heuristic frameworks and tabu search," *Managerial and Decision Economics*, vol. 11, no. 5, pp. 365–375, 1990.
25. A. Rezaeipanah, F. Sarhangnia, and M. J. Abdollahi, "META-HEURISTIC APPROACH BASED ON GENETIC AND GREEDY ALGORITHMS TO SOLVE FLEXIBLE JOB-SHOP SCHEDULING PROBLEM," *Computer Science*, vol. 22, no. 4, pp. 463–488, 2021, doi: 10.7494/csci.2021.22.4.4130.
26. S. Chakraborty, A. K. Saha, S. Nama, and S. Debnath, "COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction," *Comput Biol Med*, vol. 139, p. 104984, Dec. 2021, doi: 10.1016/J.COMPBIOMED.2021.104984.
27. J. Piri and P. Mohapatra, "An analytical study of modified multi-objective Harris Hawk Optimizer towards medical data feature selection," *Comput Biol Med*, vol. 135, p. 104558, Aug. 2021, doi: 10.1016/J.COMPBIOMED.2021.104558.
28. G. I. Sayed, M. M. Soliman, and A. E. Hassanien, "A novel melanoma prediction model for imbalanced data using optimized SqueezeNet by bald eagle search optimization," *Comput Biol Med*, vol. 136, p. 104712, Sep. 2021, doi: 10.1016/J.COMPBIOMED.2021.104712.
29. H. M. Fadhil, M. N. Abdullah, and M. I. Younis, "Innovations in t-way test creation based on a hybrid hill climbing-greedy algorithm," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 2, pp. 794–805, Jun. 2023, doi: 10.11591/ijai.v12.i2.pp794-805.
30. A. K. Alazzawi, H. M. Rais, and S. Basri, "ABCVS: An Artificial Bee Colony for Generating Variable T-Way Test Sets," 2019. [Online]. Available: www.ijacsa.thesai.org
31. Y. Alkhurayyif and A. R. W. Sait, "A comprehensive survey of techniques for developing an Arabic question answering system," *PeerJ Comput Sci*, vol. 9, 2023, doi: 10.7717/peerj-cs.1413.
32. M. Zhang, J. Sun, J. Wang, and B. Sun, " TestSGD: Interpretable Testing of Neural Networks against Subtle Group Discrimination," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 6, Sep. 2023, doi: 10.1145/3591869.
33. I. Arshad, S. H. Alsamhi, and W. Afzal, "Big Data Testing Techniques: Taxonomy, Challenges and Future Trends," *Computers, Materials and Continua*, vol. 74, no. 2, 2023, doi: 10.32604/cmc.2023.030266.
34. J. Chen, Y. Liang, Q. Shen, J. Jiang, and S. Li, "Toward Understanding Deep Learning Framework Bugs," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 6, 2023, doi: 10.1145/3587155.
35. S. Wang, Z. Cui, J. Xu, and B. Cui, "An Efficient Vulnerability Detection Method for 5G NAS Protocol Based on Combinatorial Testing," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 193, 2024. doi: 10.1007/978-3-031-53555-0_7.

36. M. A. Jamil, M. K. Nour, S. S. Alotaibi, M. J. Hussain, S. M. Hussaini, and A. Naseer, "Adaptive Test Suits Generation for Self-Adaptive Systems Using SPEA2 Algorithm," *Applied Sciences*, vol. 13, no. 20, p. 11324, Oct. 2023, doi: 10.3390/app132011324.
37. A. Bombarda and A. Gargantini, "Design, implementation, and validation of a benchmark generator for combinatorial interaction testing tools," *Journal of Systems and Software*, vol. 209, 2024, doi: 10.1016/j.jss.2023.111920.
38. A. A. Muazu, A. S. Hashim, U. D. Maiwada, U. A. Isma'ila, M. M. Yakubu, and M. A. Ibrahim, "Pairwise test case generation with harmony search, one-parameter-at-at-time, seeding, and constraint mechanism integration," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 3, pp. 3137–3149, Jun. 2024, doi: 10.11591/ijece.v14i3.pp3137-3149.
39. M. Ahmed, A. B. Nasser, and K. Z. Zamli, "Construction of Prioritized T-Way Test Suite Using Bi-Objective Dragonfly Algorithm," *IEEE Access*, vol. 10, pp. 71683–71698, 2022, doi: 10.1109/ACCESS.2022.3188856.
40. S. Li, Y. Song, and Y. Zhang, "Combinatorial Test Case Generation Based on ROBDD and Improved Particle Swarm Optimization Algorithm," *Applied Sciences*, vol. 14, no. 2, 2024, doi: 10.3390/app14020753.
41. E. Pira and M. Khodizadeh-Nahari, "Combinatorial t-way test suite generation using an improved asexual reproduction optimization algorithm," *Appl Soft Comput*, vol. 150, 2024, doi: 10.1016/j.asoc.2023.111070.
42. N. Ramli, R. R. Othman, Z. I. Abdul Khalib, and M. Jusoh, "A Review on Recent T-way Combinatorial Testing Strategy," in *MATEC Web of Conferences*, EDP Sciences, Dec. 2017. doi: 10.1051/matecconf/201714001016.
43. K. Maung Htay, R. Razif Othman, A. Amir, and J. Mohammed Hachim Alkanaani, "Gravitational search algorithm based strategy for combinatorial t-way test suite generation," *Journal of King Saud University - Computer and Information Sciences*, 2021, doi: 10.1016/j.jksuci.2021.06.020.
44. J. Torres-Jimenez, I. Izquierdo-Marquez, and H. Avila-George, "Methods to Construct Uniform Covering Arrays," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2907057.
45. A. A. Muazu, A. S. Hashim, U. D. Maiwada, and A. Muppidi, "Enhanced Version of Seeding and Constraint support in IPOG strategy for Variable Strength Interaction T-way Testing," *Malaysian Journal of Computer Science*, vol. 36, no. 4, 2023.
46. D. Gupta and A. Rana, "Fibonacci driven novel test generation strategy for constrained testing," in *Proceedings of the 2013 3rd IEEE International Advance Computing Conference, IACC 2013*, 2013, pp. 1475–1478. doi: 10.1109/IAdCC.2013.6514444.
47. Z. H. C. Soh, S. A. C. Abdullah, and K. Z. Zamli, "A Distributed T-Way Test Suite Generation Using 'One-Parameter-at-a-Time' Approach," 2013.
48. L. Yu, Y. Lei, R. N. Kacker, and D. R. Kuhn, "ACTS: A combinatorial test generation tool," in *Proceedings - IEEE 6th International Conference on Software Testing, Verification and Validation, ICST 2013*, 2013, pp. 370–375. doi: 10.1109/ICST.2013.52.
49. M. H. M. Zabil and K. Z. Zamli, "Implementing a t-way test generation strategy using bees algorithm," *International Journal of Advances in Soft Computing and its Applications*, vol. 5, no. SPECIALISSUE.3, 2014.
50. K. Rabbi, Q. Mamun, and M. D. R. Islam, "An efficient particle swarm intelligence based strategy to generate optimum test data in t-way testing," in *Proceedings of the 2015 10th IEEE Conference on Industrial Electronics and Applications, ICIEA 2015*, 2015. doi: 10.1109/ICIEA.2015.7334096.
51. A. B. Nasser, A. R. A. Alsewari, and K. Z. Zamli, "Tuning of cuckoo search based strategy for T-way testing," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 19, 2015.
52. Y. A. Alsariera and K. Z. Zamli, "A Bat-inspired strategy for t-way interaction testing," *Adv Sci Lett*, vol. 21, no. 7, 2015, doi: 10.1166/asl.2015.6316.
53. Y. A. Alsariera, H. A. S. Ahmed, H. S. Alamri, M. A. Majid, and K. Z. Zamli, "A Bat-Inspired Testing Strategy for Generating Constraints Pairwise Test Suite," *Adv Sci Lett*, vol. 24, no. 10, 2018, doi: 10.1166/asl.2018.12922.
54. A. R. Alsewari, K. Z. Zamli, and B. Al-Kazemi, "Generating t-way test suite in the presence of constraints," 2015.
55. K. Rabbi and Q. Mamun, "An effective t-way test data generation strategy," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 2015. doi: 10.1007/978-3-319-28865-9_42.
56. A. B. Nasser, Y. A. Sariera, A. R. A. Alsewari, and K. Z. Zamli, "Assessing optimization based strategies for t-way test suite generation: The case for flower-based strategy," in *Proceedings - 5th IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2015*, 2016. doi: 10.1109/ICCSCE.2015.7482175.
57. H. L. Zakaria and K. Z. Zamli, "Migrating Birds Optimization based strategies for Pairwise testing," in *2015 9th Malaysian Software Engineering Conference, MySEC 2015*, 2016. doi: 10.1109/MySEC.2015.7475189.
58. H. L. Zakaria, K. Z. Zamli, and F. Din, "Hybrid Migrating Birds Optimization Strategy for t-way Test Suite Generation," in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1830/1/012013.
59. K. Z. Zamli, B. Y. Alkazemi, and G. Kendall, "A Tabu Search hyper-heuristic strategy for t-way test suite generation," *Applied Soft Computing Journal*, vol. 44, pp. 57–74, Jul. 2016, doi: 10.1016/j.asoc.2016.03.021.

60. A. K. Alazzawi, A. A. Ba Homaid, A. A. Alomoush, and A. R. A. Alsewari, "Artificial Bee Colony algorithm for pairwise test generation," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 1–2, 2017.
61. F. Din, A. R. A. Alsewari, and K. Z. Zamli, "A Parameter Free Choice Function Based Hyper-Heuristic Strategy for Pairwise Test Generation," in *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security Companion, QRS-C 2017*, 2017. doi: 10.1109/QRS-C.2017.22.
62. B. S. Ahmed, L. M. Gambardella, W. Afzal, and K. Z. Zamli, "Handling constraints in combinatorial interaction testing in the presence of multi objective particle swarm and multithreading," *Inf Softw Technol*, vol. 86, 2017, doi: 10.1016/j.infsof.2017.02.004.
63. A. B. Nasser, K. Z. Zamli, A. R. A. Alsewari, and B. S. Ahmed, "Hybrid flower pollination algorithm strategies for t-way test suite generation," *PLoS One*, vol. 13, no. 5, May 2018, doi: 10.1371/journal.pone.0195187.
64. A. Aminu Muazu and A. Aminu Muazu, "Design of a harmony search algorithm based on covering array t-way testing strategy.," in *1st International Conference on Information Technology in Education & Development (ITED)*, Information Technology in Education & Development (ITED), 2018, pp. 33–38.
65. A. Alsewari and K. Z. Zamli, "Design and implementation of a harmony-search-based variable-strength t-way testing strategy with constraints support," *Inf Softw Technol*, vol. 54, no. 6, pp. 553–568, Jun. 2012, doi: 10.1016/j.infsof.2012.01.002.
66. A. B. Nasser, A. Alsewari, and K. Z. Zamli, "Learning cuckoo search strategy for t-way test generation," in *Communications in Computer and Information Science*, Springer Verlag, 2018, pp. 97–110. doi: 10.1007/978-981-13-0755-3_8.
67. K. Rabbi, Q. Mamun, and M. R. Islam, "A novel swarm intelligence based strategy to generate optimum test data in T-Way testing," in *Advances in Intelligent Systems and Computing*, 2018. doi: 10.1007/978-3-319-67071-3_31.
68. A. Alsewari, A. A. Mu'aza, T. H. Rassem, N. M. Tairan, H. Shah, and K. Z. Zamli, "One-Parameter-at-a-Time Combinatorial Testing Strategy Based on Harmony Search Algorithm OPAT-HS," *Adv Sci Lett*, vol. 24, no. 10, 2018, doi: 10.1166/asl.2018.12927.
69. A. AbdulRahman, L. M. Xuan, and K. Z. Zamli, "Firefly combinatorial testing strategy," in *Advances in Intelligent Systems and Computing*, 2019. doi: 10.1007/978-3-030-01174-1_72.
70. A. M. Saleh, R. R. Othman, Y. M. Yacob, and J. M. Alkanaani, "Parameters tuning of adaptive firefly algorithm based strategy for t-way testing," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, 2019, doi: 10.35940/ijitee.A6111.119119.
71. P. Ramgouda and V. Chandraprakash, "Constraints handling in combinatorial interaction testing using multi-objective crow search and fruitfly optimization," *Soft comput*, vol. 23, no. 8, 2019, doi: 10.1007/s00500-019-03795-w.
72. F. Din and K. Z. Zamli, "Pairwise test suite generation using adaptive teaching learning-based optimization algorithm with remedial operator," in *Advances in Intelligent Systems and Computing*, 2019. doi: 10.1007/978-3-319-99007-1_18.
73. M. Lakshmi Prasad, A. Raja Sekhar Reddy, and J. K. R. Sastry, "GAPSO: Optimal test set generator for pairwise testing," *Int J Eng Adv Technol*, vol. 8, no. 6, 2019, doi: 10.35940/ijeat.F8645.088619.
74. A. R. A. Alsewari, R. Poston, K. Z. Zamli, M. Balfaqih, and K. S. Aloufi, "Combinatorial test list generation based on Harmony Search Algorithm," *J Ambient Intell Humaniz Comput*, 2020, doi: 10.1007/s12652-020-01696-7.
75. A. B. Nasser, F. Hujainah, A. A. Al-Sewari, and K. Z. Zamli, "An improved jaya algorithm-based strategy for t-way test suite generation," in *Advances in Intelligent Systems and Computing*, 2020. doi: 10.1007/978-3-030-33582-3_34.
76. A. A. Hassan, S. Abdullah, K. Z. Zamli, and R. Razali, "Combinatorial test suites generation strategy utilizing the whale optimization algorithm," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3032851.
77. A. B. Nasser, A. S. H. Abdul-Qawy, N. Abdullah, F. Hujainah, K. Z. Zamli, and W. A. H. M. Ghanem, "Latin Hypercube Sampling Jaya Algorithm based Strategy for T-way Test Suite Generation," in *ACM International Conference Proceeding Series*, 2020. doi: 10.1145/3384544.3384608.
78. A. Aminu Muazu and U. D. Maiwada, "PWiseHA: Application of Harmony Search Algorithm for Test Suites Generation using Pairwise Techniques," *International Journal of Computer and Information Technology*, vol. 9, no. 4, pp. 2279–0764, 2020, [Online]. Available: www.ijcit.com
79. M. S. A. Rashid Ali, R. R. Othman, Z. R. Yahya, and M. Z. Zahir, "A Modified Artificial Bee Colony Based Test Suite Generation Strategy for Uniform T-Way Testing," in *IOP Conference Series: Materials Science and Engineering*, 2020. doi: 10.1088/1757-899X/767/1/012020.
80. A. B. Nasser, K. Z. Zamli, N. W. B. M. Nasir, W. A. H. M. Ghanem, and F. Din, "T-way Test Suite Generation Based on Hybrid Flower Pollination Algorithm and Hill Climbing," in *ACM International Conference Proceeding Series*, 2021. doi: 10.1145/3457784.3457822.

81. K. M. Htay, R. R. Othman, A. Amir, H. L. Zakaria, and N. Ramli, “A Pairwise T-Way Test Suite Generation Strategy Using Gravitational Search Algorithm,” in *ICAICST 2021 - 2021 International Conference on Artificial Intelligence and Computer Science Technology*, 2021. doi: 10.1109/ICAICST53116.2021.9497823.
82. H. N. Nsaif and D. Norhayati Abang Jawawi, “Binary Black Hole-Based Optimization for T-Way Testing,” in *IOP Conference Series: Materials Science and Engineering*, 2020. doi: 10.1088/1757-899X/864/1/012073.
83. C. Luo *et al.*, “AutoCCAG: An Automated Approach to Constrained Covering Array Generation,” in *IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, Institute of Electrical and Electronics Engineers (IEEE), May 2021, pp. 201–212. doi: 10.1109/icse43902.2021.00030.
84. Y. A. Alsariera, Y. Sanjalawe, A. H. Al Omari, M. A. Albawaleez, Y. K. Sanjalawe, and K. Z. Zamli, “Hybridized BA & PSO t-way Algorithm for Test Case Generation Cloud Computing Security View project Detection DDoS attack approaches against SDN View project Hybridized BA & PSO t-way Algorithm for Test Case Generation,” *IJCSNS International Journal of Computer Science and Network Security*, vol. 21, no. 10, p. 343, 2021, doi: 10.22937/IJCSNS.2021.21.10.48.
85. H. M. Fadhil, M. N. Abdullah, and M. I. Younis, “TWGH: A Tripartite Whale–Gray Wolf–Harmony Algorithm to Minimize Combinatorial Test Suite Problem,” *Electronics (Switzerland)*, vol. 11, no. 18, 2022, doi: 10.3390/electronics11182885.
86. J. B. Odili, A. B. Nasser, A. Noraziah, M. H. A. Wahab, and M. Ahmed, “African Buffalo Optimization Algorithm Based T-Way Test Suite Generation Strategy for Electronic-Payment Transactions,” in *Lecture Notes in Networks and Systems*, 2022. doi: 10.1007/978-3-030-82616-1_15.
87. Rozmie R. Othman, Norazlina Khamis, and Kamal Z.Zamli, “Variable Strength T Way Test Suite Generator with Constraints Support,” *Malaysian Journal of Computer Science*, vol. 27, no. 3, 2014.
88. S. A. C. Abdullah, Z. H. C. Soh, and K. Z. Zamli, “Variable-strength interaction for t-way test generation strategy,” *International Journal of Advances in Soft Computing and its Applications*, vol. 5, no. SPECIALISSUE.3, 2014.
89. K. Z. Zamli, F. Din, S. Baharom, and B. S. Ahmed, “Fuzzy adaptive teaching learning-based optimization strategy for the problem of generating mixed strength t-way test suites,” *Eng Appl Artif Intell*, vol. 59, pp. 35–50, Mar. 2017, doi: 10.1016/J.ENGAPPAI.2016.12.014.
90. A. B. Nasser and K. Z. Zamli, “A new variable strength t-way strategy based on the cuckoo search algorithm,” in *Lecture Notes in Networks and Systems*, vol. 67, 2019. doi: 10.1007/978-981-13-6031-2_17.
91. A. A. B. A. Homaid, A. R. A. Alsewari, K. Z. Zamli, and Y. A. Alsariera, “Adapting the elitism on greedy algorithm for variable strength combinatorial test cases generation,” *IET Software*, vol. 13, no. 4, pp. 286–294, Aug. 2019, doi: 10.1049/iet-sen.2018.5005.
92. N. Ramli, R. R. Othman, Z. I. A. Khalib, M. Z. Z. Ahmad, and S. S. M. Fauzi, “Ant colony algorithm to generate t-way test suite with constraints,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jun. 2020. doi: 10.1088/1742-6596/1529/4/042103.
93. J. M. Altmemi, R. R. Othman, and R. Ahmad, “SCAVS: Implement Sine Cosine Algorithm for generating Variable t-way test suite,” in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, Sep. 2020. doi: 10.1088/1757-899X/917/1/012011.
94. A. K. Alazzawi and S. Basri, “HABC: Hybrid artificial bee colony for generating variable t-way test sets,” 2020.
95. M. Younis, “GAMIPOG: A Deterministic Genetic Multi-Parameter-Order Strategy for the Generation of Variable Strength Covering Arrays,” 2020. [Online]. Available: https://www.researchgate.net/publication/344599430
96. A. K. Alazzawi *et al.*, “HABCSm: A Hamming Based t -way Strategy based on Hybrid Artificial Bee Colony for Variable Strength Test Sets Generation,” *International Journal of Computers, Communications and Control*, vol. 16, no. 5, 2021, doi: 10.15837/ijccc.2021.5.4308.
97. N. Ramli, R. R. Othman, R. Hendradi, and I. Iszaidy, “T-way Test Suite Generation Strategy based on Ant Colony Algorithm to Support T-way Variable Strength,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1755/1/012034.
98. M. Z. Zahir Ahmad, R. R. Othman, N. Ramli, and M. S. A. Rashid Ali, “VS-TACO: A Tuned Version of Ant Colony Optimization for Generating Variable Strength Interaction in T-Way Testing Strategy,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Feb. 2022, pp. 48–54. doi: 10.1145/3524304.3524311.
99. M. I. Younis, A. R. A. Alsewari, N. Y. Khang, and K. Z. Zamli, “CTJ: Input-output based relation combinatorial testing strategy using jaya algorithm,” *Baghdad Science Journal*, vol. 17, no. 3, pp. 1002–1009, Sep. 2020, doi: 10.21123/BSJ.2020.17.3(SUPPL.).1002.
100. A. S. M. Ali, R. R. Othman, Y. M. Yacob, and H. S. A. Ben Abdelmula, “An Efficient Combinatorial Input Output-Based Using Adaptive Firefly Algorithm with Elitism Relations Testing,” *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 4, pp. 223–232, Jul. 2021, doi: 10.25046/aj060426.
101. R. N. Pagani, J. L. Kovaleski, and L. M. Resende, “Methodi Ordinatio: a proposed methodology to select and rank relevant scientific papers encompassing the impact factor, number of citation, and year of publication,” *Scientometrics*, vol. 105, no. 3, 2015, doi: 10.1007/s11192-015-1744-x.

102. J. M. Sharif, K. Z. Zamli, A. A. Bakar, S. Abdullah, I. S. Isa, and I. R. M. Noordin, "A non-deterministic T-way strategy with seeding and constraints support," in *SHUSER 2012 - 2012 IEEE Symposium on Humanities, Science and Engineering Research*, 2012. doi: 10.1109/SHUSER.2012.6268795.