DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

Combinatorial Testing Approaches: A Systematic Review

Heba Mohammed Fadhil[1], Mohammed Najm Abdullah[2], Mohammed Issam Younis[3]
[1]*Department of Information and Communication, Al-Khwarizmi College of Engineering, University of Baghdad, Baghdad, Iraq*
[2]*Department of Computer Engineering, University of Technology, Baghdad, Iraq*
[3]*Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq*
[1]*ce.19.15@grad.uotechnology.edu.iq,* [2]*mohammed.n.abdullah@uotechnology.edu.iq,*
[3]*younismi@coeng.uobaghdad.edu.iq*

Abstract*— Testing is a vital phase in software development, and having the right amount of test data is an important aspect in speeding up the process. As a result of the integrationist optimization challenge, extensive testing may not always be practicable. There is also a shortage of resources, expenses, and schedules that impede the testing process. One way to explain combinational testing (CT) is as a basic strategy for creating new test cases. CT has been discussed by several scholars while establishing alternative tactics depending on the interactions between parameters. Thus, an investigation into current CT methods was started in order to better understand their capabilities and limitations. In this study, 97 publications were evaluated based on a variety of criteria, including the generation technology, test strategy method, supported interactions, mixed coverage ,and support constraints between parameters. CT analysis had a wide range of interaction assistance options available to researchers. Since 2010, a unified interaction has been the most common style of interaction between the two parties. The year 2018 was hailed as the most successful in terms of CT by researchers. Researchers should focus on one test at a time and metaheuristic search strategies for t-way CT. There has also been a significant increase in the popularity of other trends, such as deep learning (DL). CT appears to be a useful testing technique for balancing and fault detection capabilities for a variety of systems and applications, according to our research. Future research and software development may benefit from this information.*

Index Terms*— Combinatorial Testing, Test Case Generation, Optimization Algorithms, Software Testing, Artificial Intelligent.*

I. INTRODUCTION

So far, many big companies that collect huge amounts of data, especially banks and web-based communications, have gained popularity with the ability to work with big data sets due to big data technology. Currently, interested parties are evaluating how these data are stored and controlled. For example, customers' behavioral data can be converted into financial value through various types of critical imperatives. However, it is not feasible to create platforms capable of loading, storing, and executing large datasets. In addition, analyzing this huge amount of data and ensuring that it is free from errors requires a reliable testing agent; here, the role of software testing comes [1], [2] ,[3]. A service or product that has been rigorously tested is essential to revealing whether the customer is pleased. Configuration system testing may prove hazardous. Testing with a system with many configurations, each of many possible values, is impractical. While combinatorial testing (like select a susbset from the exghustive test systematicall reduced the amount and duration necessary to conduct testing, it also proved

Received 06/February/2022; Accepted 04/May/2022

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

the framework to be viable and functional. Testing techniques, such as combinatorial testing, are only as good as the test sets they are run against. This includes actual fault settings. When actual fault settings were not included in the test sets, defects were not detected [4].

A combinatorial interaction test, uses combinatorial interactions (t-way) to isolate and classify errors. In general, combinatorial tests appear to be either deterministic or non- deterministic. They are referred to as combinatorial interactions or combinatorial tests because each one forms a sequence on its own, without further interaction. The resulting test suites order of parameter-values are called either covering array (CA), sequence covering array (SCA), and multivalued sequence covering array (MVSCA)[5]. Within the test suite generation phase of computational strategies, two approaches are used: one-test-at-a-time and one-parameter-at-a-time. The prime concern involves the generation of all t-way interaction tuples, which is followed by an inquiry algorithm that produces a test suit that all t-way. The generated legal action was then added to the test suite. This procedure was repeated for each action until all the tuples were covered. This strategy was used to generate the CAs and SCAs. But at the other hand, with a one-parameter-at-a-time approach, the interaction elements within the generation are partitioned for each input parameter [6].

Combinatorial interaction testing has also been successfully implemented for the evaluation of a variety of configurable software and hardware systems. Numerous survey and research articles have been published on the subject, offering both tactical and strategic solutions [7],[8]. However, numerous strategies for testing combinatorial interactions have been developed. At this time, no current research exists in this field that goes into the topic of extensive and focused review. This has led to research on other means of generating combinatorial test suites. In addition, there are multiple systems and empirical evaluations that constitute a survey of combinatorial interaction research in the preceding decade. Its goal is to uncover what is essential, identify what is out of date, and quantify what is missing in the literature in order to serve as a guide for future studies. Over the last decade, research on this subject has been performed using a scientific methodology. The literature has been significantly influenced by various concerns. There are several papers on combinatorial testing, but none have focused on all aspects, including combinatorial interaction tests. Combinatorial testing is emerging as a promising area of research; it is interesting to take a look at this area. A research article on combinatorial testing may also help interested scholars quickly grasp and master the subject topic. The rest of the paper is organized as follows: Section II summarizes the combinatorial testing. Section III discusses the motivation for this study and provides an overview of the related work. Section IV discusses the methodology used to conduct the literature review. Section V shows the results. Section VI summarizes the findings of this study and Section VII discusses potential threats to validity. Finally, section VIII closes the work.

II. COMBINATORIAL TESTING

With the significance of testing has dramatically increased, an increasing number of system components, devices, and operating systems will interconnect and cause additional interactions, increasing the likelihood of defects. One solution to this problem is the plethora of interaction-based testing methodologies, known as t-way strategies. Combinatorial testing was used to generate test cases in accordance with the System Under Test (SUT). After examining the requirements, the parameters and values were used to describe the input to the system. A set of tests was conducted to cover all possible combinations of the specified parameters [9]. Following the analysis requirements, the parameters and values describe the system input, and every possible combination is tested. As all these mixes of parameters increase, the test suite size increases. In order to limit the amount of test cases, it is critical to have a test strategy; they formed each combinational testing notation. As the sample size increased, the ideal test suite dimensions increased. Earlier studies in combinatorial testing have shown that the creation of test cases is both NP-complete and NP-hard, as there is no unique method for

Received 06/February/2022; Accepted 04/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

covering tuples. generating the smallest possible sample size is difficult because no one answer exists [10]. Researchers have devised numerous approaches for combinatorial test suites. According to Kuhn *et al.*, it has been proven that some legitimate input combinations are infeasible in practice, and very few problems arise from odd combinations of circumstances.Combinatorial testing is effective and economical for reducing test suite size by up to a factor of 70% to 90% [11].

To better understand how the CA approach works, let us take the following example, taking a look at the example provided in for web-based systems. Table I implies four parameters, each with three possible values. Suppose that the complete experimental design includes (t = 4), which leads to $3^4 = 81$ probabilities; however, if the relations are simplified to t = 2 (pairwise), the number of test cases that can be constructed will be reduced to nine possibilities.

TABLE I. WEB-BASED SYSTEM PARAMETERS.

	Browser	OS	DBMS	Connections
0	Firefox	Windows 7	MySQL	ISDN
1	Chromium	Ubuntu 10.10	PostgreSQL	ADSL
2	Netscape	Red Hat 5	MaxDB	Cable

To map the CA for the web-based system, Table I shows that every potential value is given a unique row number, as Table II presents the pairwise test combination.

TABLE II. A COMPREHENSIVE TEST SUITE FOR TWO-WAY INTERACTIONS, CA (9; 2, 4, 3).

		Experiments		
1	Firefox	Windows 7	MySQL	ISDN
2	Chromium	Ubuntu 10.10	PostgreSQL	ADSL
3	Netscape	Red Hat 5	MaxDB	Cable
4	Chromium	Windows 7	PostgreSQL	Cable
5	Chromium	Ubuntu 10.10	MaxDB	ISDN
6	Chromium	Red Hat 5	MySQL	ADSL
7	Netscape	Windows 7	MaxDB	ADSL
8	Netscape	Ubuntu 10.10	MySQL	Cable
9	Netscape	Red Hat 5	PostgreSQL	ISDN

Mathematically, CA takes the parameters N, t, p, and v, which correspond to size and coverage strength, and the number of parameters is used in addition to having a uniform value. The parameter reaction number was determined using Eqs. (1).The number of t-way reaction tuples for each interaction, as indicated by Eqs. (2). Thus, the total number of groups was determined using Eqs. (3). The CA is an N * p matrix, where all the t-way groups consist of at least one row. It should be noted that these parameters are ordered contiguous [5]. The combinatorial test approach requires the extraction of test cases from the functional domain of the SUT. Larger input domains are favored when deriving tests from smaller output domains. The systems are put through the paces with safety domain-specific test cases to produce maximum outputs[12].

A mixed covering array (MCA) can be used to deal with situations in which the number of component values changes. An MCA (N;t,k, (v_1,v_2,v_k)) is an N*k array going on v values, in which the rows of each N * t sub-array cover and all t interactions of values from the t columns appear at least once. For additional flexibility in the notation, the array can be given by the MCA (N;t,$v_1^{k1} v_2^{k} .. v_k^{k}$). In response to the proliferation of diverse uses for CA and MCA, the variable strength covering array (VSCA) notation has been developed as a new variation of CA and MCA. A VSCA (N; t, k, a vector of the CA or MCA, and a subset of the k columns each with (v_1,v_2 …v_k), C) is an N × k CA or MCA of strength t containing C, a strength >t . Interactions among variables have varying effects on various software

Received 06/February/2022; Accepted 04/May/2022

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

systems. System failures can be caused by some interactions, whereas failures caused by other interactions may have hardly any effect on the system. Varying the intensity of the covering arrays can be used to detect these various interactions efficiently. This can provide various covering strengths to distinct groupings of variables, and thus can provide a realistic way to evaluate real applications [13] , [14].

$$\binom{p}{t} \stackrel{\text{def}}{=} \frac{p!}{t!(p-t)!} \tag{1}$$

$$v^t \tag{2}$$

$$\frac{p!}{t!(p-t)!}\ v^t \tag{3}$$

III. RELATED WORK AND MOTIVATION

Nie and Lueng [15] reviewed the primary of combinatorial testing and thier applications. This is about the essential concepts and construction methodologies of combinatorial testing. Kuliamin and Petukhov [16] also included different ways of creating combinatorial test cases. Ahmed and Zamli [17] applied research on the relationship between interaction testing and devices. Torres-Jimenez and Izquierdo-Marquez [18] gathered existing knowledge on various cover array construction approaches. The techniques have been grouped into three classes: greedy, metaheuristic, and algebraic. Wu and Nie [19] assessed the search and analysis of suite creation using search-based combinatorial testing (SBCT). Khalsa and Labiche [20] studied 75 research publications included algorithm criteria such as support for building of algorithms/applications, with priority for techniques that allow for combination.

Recently, Lopez-Herrera *et al.* [21] performed the main body of work on a software product line (SPL). SPL testing enables various study designs to be considered in its approach and allows constraints to be considered. However, the research has only focused on constraints that could have arisen due to SPL implementation. Alsewari *et al.* [22] revised the test generation processes relating to character-based algorithms and procedures with respect to combinatorial tests to validate IOR, in which combinatorial input information processing was assessed using the IOR. Their results highlight existing algorithm strategies using an algorithm nature-inspired nation. Cobos-lozada and Torres-jimenez [23] studied metaheuristics, such as genetic algorithms, simulated annealing, particle swarm, and ant colony, which may be used to solve different problems, were applied to the problem, resulting in an outline of the key metaheuristic contributions that were made, including algorithmically based population particle harmonization and algorithmically directed adaptation. Ramli *et al.* [24] focused on the investigation of present tools or algorithms from 2010 to the primary quarter of 2017.They analyzed 20 algorithms and tools search techniques, supported strategy approach, supported interaction, and year of publication.

Zamli *et al.* [25] guided users and encouraged the adoption of this notion by examining the current state-of-the-art and outlining current industrial practices. The objective of thier research is to lay out many two-way approaches, including uniform interaction, variable interaction, and input-output-based relations. Ahmed *et al.* [26] looked at approximately 100 papers released between 2005 and 2016 and covered most of them. The questions in the study were studied from multiple perspectives. Their findings showed that responses to constricted interactions are on the rise. Researchers have also found that one way to remove or reduce constraints is to use a constraint solver. The investigation closes with a discussion on the matter and prospects for constrained testing.

Prakash *et al.* [9] offered a step-by-step testing function to generate reviews of a novel algorithm. A significant amount of effort has gone into understanding that number in the past, meaning that more work can be done in the future. Testing with PSO was shown to generate a combinatorial number of

Received 06/February/2022; Accepted 04/May/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

test cases. Li *et al.* [27] started a review with a summary of combinatorial testing and briefly mentioned its primary applications. They organized combinatorial applications according to two different classifications, including standard industry and IT. Altmemi *et al.* [28] summarized the prevailing strategies and identified the capabilities and weaknesses of ways to provide valuable and useful guidance for long-term approaches. Setiani *et al.* [29] focused on the latest and most significant issues in the entire generation research domain. Wu *et al.* [30] analyzed the conventions of CT scanning and the results of published studies between 1987 and 2018. Their study classifies the many ways in which the limitations of dealing with constraints as well as identifying the types of less effective ones and evaluating less important ones. Naseri and Koffas [31] used modern combinatorial methods to identify the methodological approaches and technological advancements that assist with their application.
Numerous papers including numerous validation sources were considered excellent data for the study because they are quite valuable as they offer relevant literature on interaction testing. As a result, this study expands on prior work on test data generation strategies. This paper could serve as a more in-depth review of previously published summaries on the subject of the zone because it covers new developments and research more comprehensively. However, none of the previous review papers are exhaustive. They were also not carried out as systematic literature studies specialized in search-based combinatorial testing.

IV. METHODOLOGY

This section describes the methodology used in this research. This study was in accordance with the suggested methodology standards [33], [34]. The methodology consists of the following four stages:

- **Stage I.** At this point that the research design are crystallized. Using such research approach to collect data. Only the most relevant scientific material was returned after a comprehensive investigation of the search query.
- **Stage II.** Filtering: As appropriate papers are selected and judged, irrelevant papers are filtered out based on the following criteria: titles, full texts, bookmarks, and assessments depending on the merits.
- **Stage III**. Analysis: Once all required information has been gathered from the beginning of the research studies, more in-depth research can begin (in this instance, 200 studies were involved). To provide a fuller picture of the results, the primary data of the original studies were further analyzed and classified.
- **Stage IV**. Writing: asking the right questions can help find new approaches to creating something new and creative; at this point, the challenges to validity are investigated as part of a search for deficiencies in the research.

Finally, the information is categorized and organized after all the other stages for literal depiction and organization, as illustrated in *Fig 1*.

Received 06/February/2022; Accepted 04/May/2022

 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

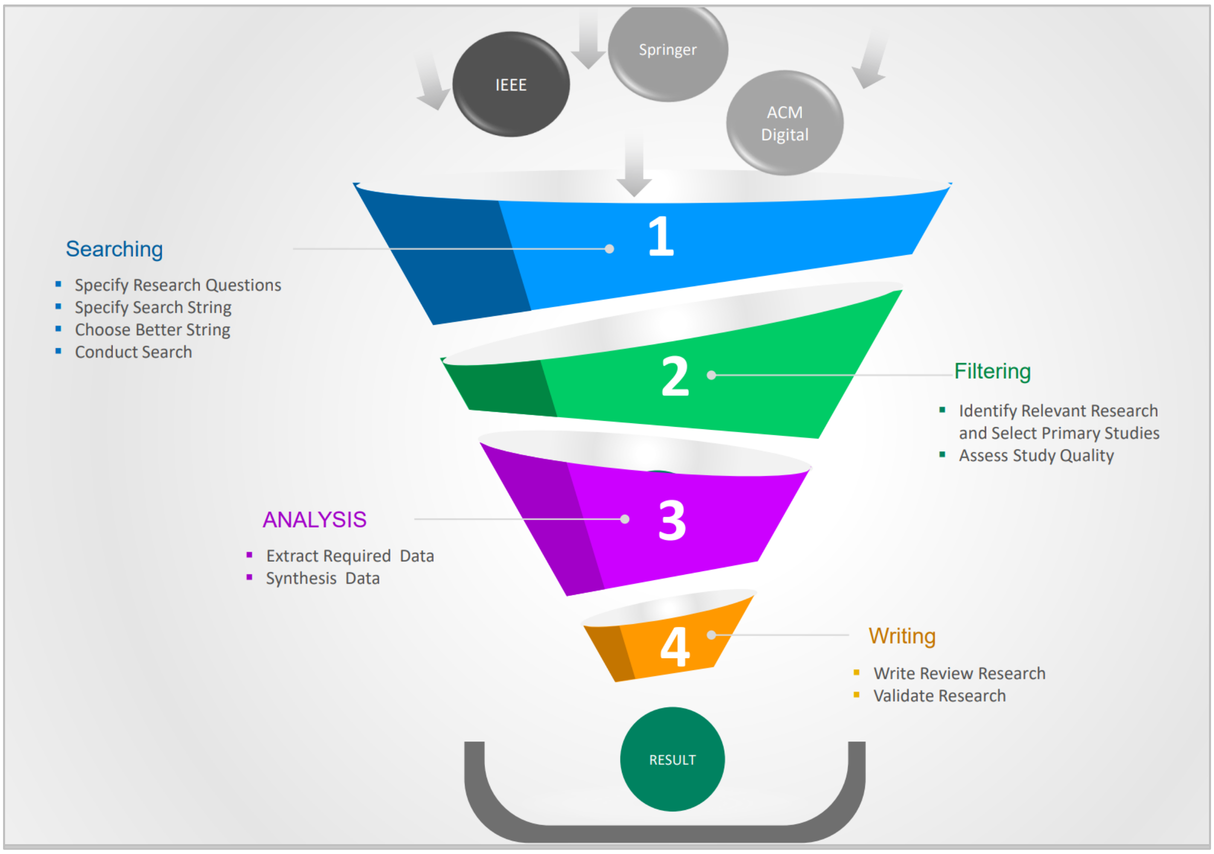

FIG. 1. THE GENERALIZED DETAIL LITERATURE PROCESS.

A. Search Strategy

The search for keywords in a complex textual collection is difficult. Kitchenham and Charters [33] divided the words into three sections: What interests users into interest types. It narrows the scope of combinatorial testing, which is described as "combinatorial interaction" making greater effort to search using a well-defined set of keywords. Some types of frameworks have been recommended. The preview, question, read, and summarize (PQRS) method helps researchers remain focused consistently and enables quick identification and information retrieval [34].

Researchers classified the research question keywords and PQRS criteria keywords into subsets because they were comparable. The papers were searched on IEEE Xplore, ACM, and Scopus. Databases were selected based on their long history [35]. The second keyword set is composed of combinatorial tests, which are built to yield software-related strategies. To widen the search, Boolean AND combines search keywords, while using alternative phrases is enhanced. The first draft of the search string was ready to go, and multiple tests were required to develop the final draft. Because of the combinatorial link with other interactions, these experiments are necessary. Search queries were judged based on how well they met the requirements. The final string was selected to restore these studies to the research topic. Table III lists four sets of search terms. The initial three sets of search strings yielded less relevant results; thus, they were removed from the selection process. According to these data, this general terminology in the majority of papers produced results in various ineffective documents. Furthermore, the meaning of “combinatorial” has been observed depending on the context of the sentence. Among other things, it was found that the search tested several word sets. For this reason, these two concepts, “combinatorial interaction “ and “ covering array “ are noted in many scientific texts and may belong to other areas. In order to guarantee the comprehensiveness of the study topics, additional terminology such as (“strategy”, “approach” , “technique" and “instrument") were included. They were designed using the criteria and guidance supplied by [32], [36] ,[35]. To comply with these

Received 06/February/2022; Accepted 04/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

criteria, the following sources were incorporated into the search results: there are two databases that are well-known: IEEE Xplore and ScienceDirect, notably "Springer "or "Scopus" in addition to the ACM Digital Library was also among the resources that are used in this research.

Some discrepancies in the indexing algorithms led to modest index changes while finding and indexing numerous references. "Mendeley" was characterized as a reference management program since it dealt with duplicates by integrating several file repositories into a single location. Checking off was performed manually to improve the accuracy. In addition, the growing research trends since 2010 must be acknowledged in the literature review. The origins of the study, which started from the third quarter of 2020, to keep the information in this research field updated until the first quarter of 2021, during the time frame of writing this research.

TABLE III. SEARCH STRINGS.

Keyword	Searching String
Set #1	((covering array) OR (t-way testing)) and (combinatorial testing)
Set #2	((covering array) OR (t-way testing)) and ((interaction testing) OR (combinatorial testing))
Set #3	((covering array) OR (t-way testing)) and ((combinatorial) OR (interaction testing) OR (combinatorial testing)) and (strategy OR technique OR method OR approach)
Set #4	((covering array) OR (t-way testing)) and ((combinatorial) OR (interaction testing) OR (combinatorial testing)) AND (testing) AND (strategy OR technique OR method OR approach)

B. Criteria for Paper Selection and Assurances of Quality

Previously, it has been said that articles ought to be considered through an abstract only, but so far, others have been refuted by full proof text. The esthetics of the papers were also well thought-out because they must be comfortable writing them. The first author revised the choices to make them more reliable. Included titles and abstracts were certain articles that had been accepted and rejected because of this result, and remember that this must be taken into consideration. Furthermore, most items required comprehensive reading before their choices were made. Final studies to exclude articles to be created were calculated and compiled, which were in the form of the initial study. A full-text search utilizing these studies was chosen to provide more information:

- Several complementary studies that have already been done have uncovered the importance of using covering arrays for combinatorial testing.
- Applications of the combinatorial approach to testing.
- Software engineering studies that have appeared in the literature over the past decade.

The research also needed to account for the exclusion criteria defined in the guidelines by [36], and it became apparent that articles that did not meet the inclusion/exclusion criteria may be eliminated. Backward scanning techniques are applied to unrecorded conversations to determine what was previously stated in [37], [36], and the study flow increased, leading to the inclusion of 27 studies. The following procedures were employed to eliminate studies that have previously appeared in the publications.

- Studies involving testing but lacking the support of combinatorial testing.
- The use of testing has been widely expanded in fields besides software engineering.
- Readings that have not been published in english
- Readings where no entire text is provided.
- Abstruse books and literature.
- Research obtained outside peer-reviewed journals.

Based on these principles, a smaller subset of papers was selected, helping to limit the study's subject matter to be concentrated on only those that have these qualities. As noted previously, a few of these papers have been published multiple times and are eventually removed. *Fig. 2* depicts the data

Received 06/February/2022; Accepted 04/May/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

processing required to choose papers from a number of databases that were found to be sufficiently broad in their scope to be suitable for study. This methodology has also been employed in other literary and theoretical studies, such as [36],[38]. First, associated papers were initiated in the chosen databases (i.e., IEEE Xplore, ScienceDirect, ACM Digital Library, Scopus, and SpringerLink). The paper number prediction previously stated that 6,869 papers were produced. It is important to note that these documents are shared by different databases. As a result, the "Exclude Duplicates" step was completed, removing all titles that had been present in all previous "Filter 1" iterations, as well as abstracts , presentations, titles, and introductions were studied using "Filter 2”. The previously described research criteria were employed to identify the publications that were selected for research. The researchers read all the papers as part of "Filter 3” some conference papers describe specific research findings, such as research study outcomes, which are rarely included in the papers themselves. Furthermore, their primary aim was software engineering rather than research. Finally, relevant papers from other publications were searched to ensure that any relevant investigations were omitted.

Additionally, the bibliography was searched for each paper picked, and several others were found to be on the subject. The automatic search does not identify objects that may have been missed. This time, finding more sources was made through an attempt to carefully look for the paper's title. However, this decision was made to include other sources only through reference seeking, because most of the material was of sufficient quality and did not provide information that could not be disclosed (see *Fig. 2*). Thus, another 17 pieces of paper have been tacked here, resulting in extra work. At last 97 papers resulted, as a result of which information was extracted from them.

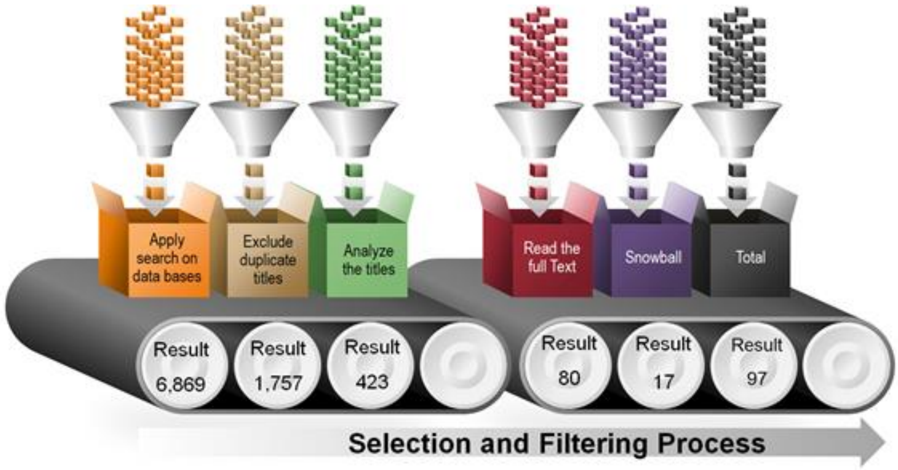

FIG. 2. THE NUMBER OF PAPERS SELECTED AND FILTERED.

C. Acquisition and Relevance of the Data

Once the studies were selected, the primary purpose was to extract the data and explore the questions. A spreadsheet was constructed from the collected data to perform this task. The sheet was built using a template developed by [38], [36]. With all these additional fields, the table has become quite vast. The design shown in Table IV utilizes a template. A remark like this regarding the document describes the information recorded: paper ID, paper name, publication year, author names, subject matter and countries, classification. However, taking into consideration the study's methodology, gathering and analyzing the data also helped the researcher comprehend each study's goals. In contrast, the frequencies were determined after the process was complete. The information in Table IV can indeed be found in each document; because the material for all publications was carefully extracted and tabulated, the results are accurate and reliable. The information was extracted and verified by a research team (names mentioned on the first page).

Received 06/February/2022; Accepted 04/May/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

TABLE IV. FORMULA OF DATA COLLECTION.

Data item	Value
Paper ID	Integer
Publication Title	The publication's title
Author Name	Authors' given names
Year of Publication	Year according to the publication
Country	Each author's country
Research Topic	The study's principal subject or concept
Classification	How were articles classified
Applied techniques	What techniques were applied
Contribution	The publication's primary contribution
Challenges	problems which are highlighted in the study
Evaluation process	Which standardization method was employed?

V. RESULTS

The primary aim of this part is to respond to all of the research approch mentioned previously, which evaluates each research approch from Section IV individually. Each part has a brief title that is generated from the research strategy.

A. Publications on a Regular Basis

Cited studies were reviewed throughout the last decade (2010-2021) to determine if there are significant differences in how often and in the degree of publishing numbers, as shown in *Fig. 3*. The increase in the number of publications during the past few years has had a significant effect on the average. To date, it has been established that 97 papers with an average publication lifespan of ten years.

The curiosity in combinatorial interaction testing increased moderately between 2010 and 2021, with a significant increase in research in 2017 and beyond. This increase in the number of publications reflects the growing interest in software engineering communities in research. Another probable explanation for this rise in combinatorial interaction test studies is the move away from theory to the application of combinatorial methods, with new studies exploring the numerous case studies on which combinatorial approaches have been tested. A publication venue is decided based on the type of information shown in *Fig 4*. 64% of the publications, were academic journals, 28% were conference papers, and 7% were workshop papers. Researchers should also emphasize that

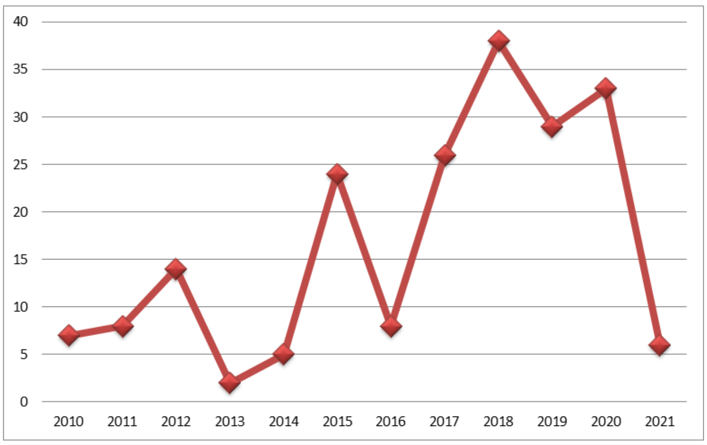

FIG. 3. TOTAL NUMBER OF DIFFERENT TOOLS OR METHODS IDENTIFIED DURING A PERIOD OF DECADE.

Received 06/February/2022; Accepted 04/May/2022

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

although certain conferences produced as publications have made it into chapters in books, those books were based on the original venues that they first appeared in. These results demand close attention because combinatorial combinative work is likely to become the most preferred peer-reviewed field in the research community. According to *Fig. 5*, publications have been published that are relevant in the field of active-learning versus incumbent-learning publications. Looking specifically at journal publications, it is clear that ‘‘Information and Software Technology" ‘‘IEEE Access" ‘‘Communications in Computer and Information Science" and ‘‘PLoS ONE,” ‘‘Advanced Science Letters,” ”Applied Soft Computing Journal” and “Engineering Science and Technology, an International Journal” The leading and most-publishing journals account for almost 25% of the total journal publications. However, numerous papers have appeared in conference publications, such as academic and community ‘‘IEEE International Conference on Software Testing, Verification, and Validation Workshops, ICSTW,” ‘‘IEEE International Conference on Software Quality, Reliability, and Security Companion" and ” ACM International Conference. ” Beyond the mainstream press, academicians write academic papers for research grant agencies, international forums, conferences, journals, and committee proceedings books. However, these findings show that when one looks at conferences with a number of papers above this upper limit, 36% of the papers published at academic conferences per year are cited twice. The remaining papers were presented at 15 distinct conferences, denoted as others in *Fig. 6*.

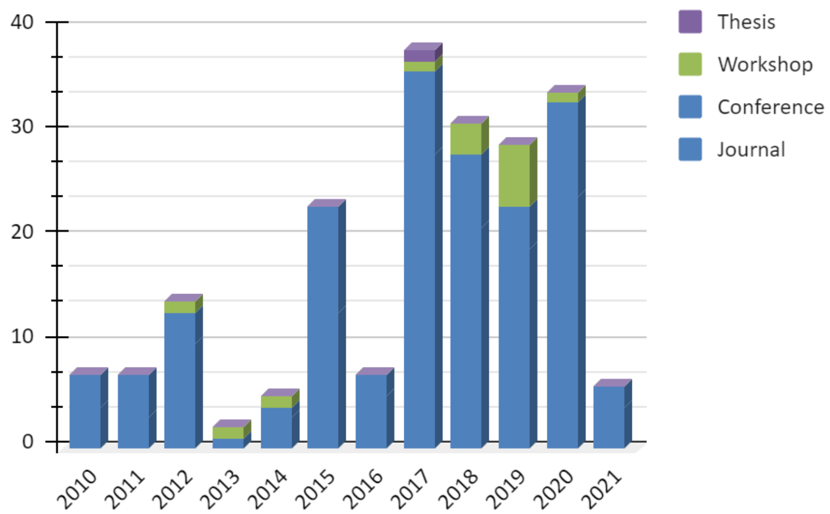

FIG. 4. NUMEROUS TOOLS/ALGORITHMS IDENTIFIED OVER SEVERAL YEARS AND PRESENTED IN FOUR DIFFERENT VENUES.

Received 06/February/2022; Accepted 04/May/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. 22, No. 4, December 2022

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

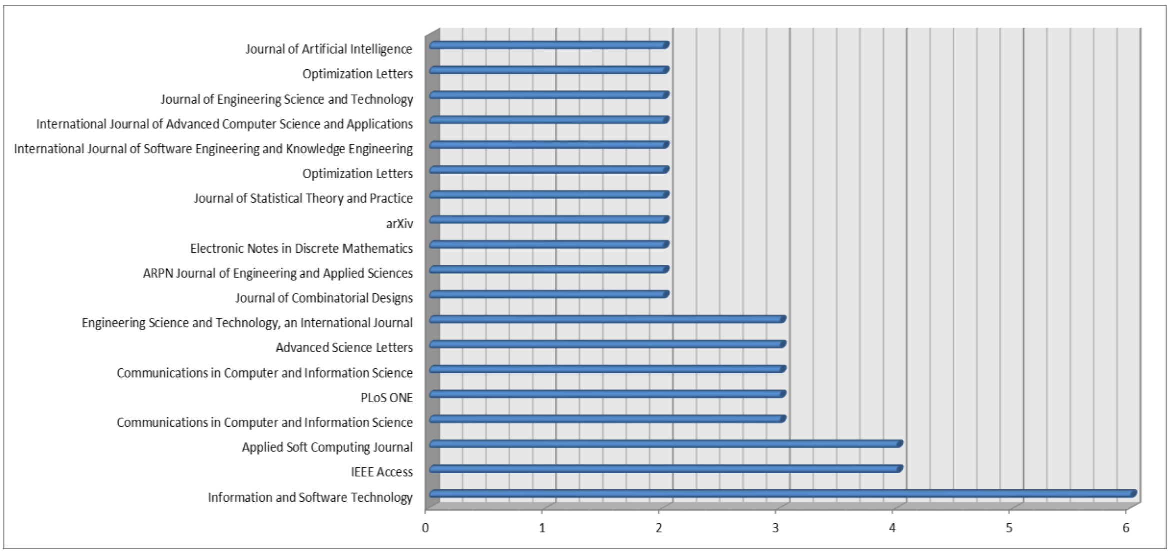

FIG. 5. PAPER NUMBER VS. JOURNAL NAME.

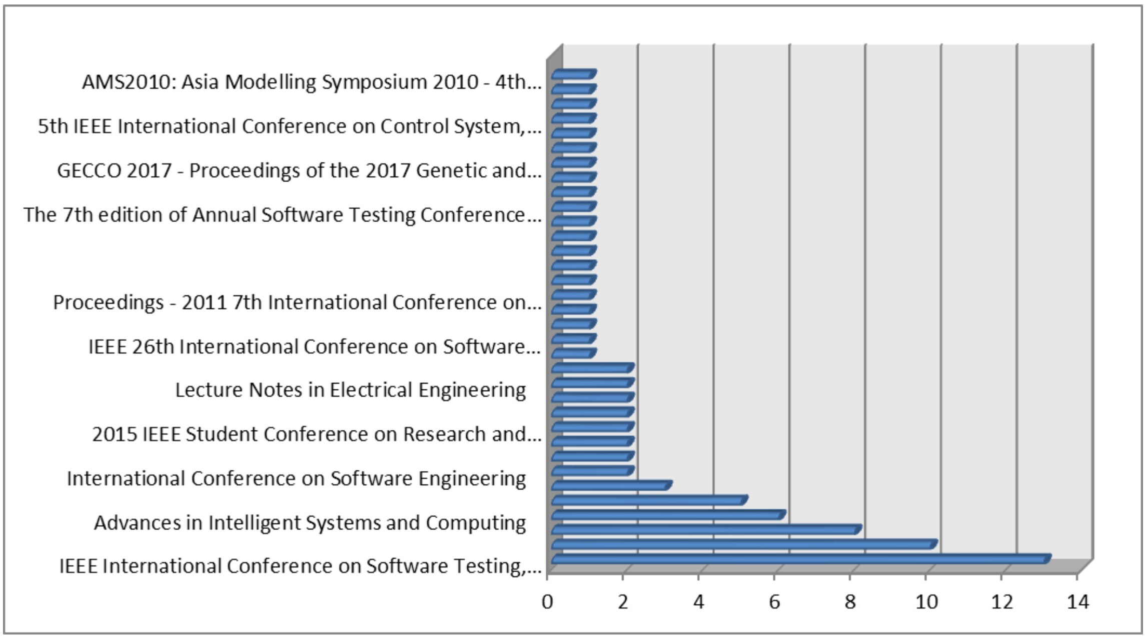

FIG. 6. AMOUNT OF PUBLICATIONS VS. VENUES OF CONFERENCE.

B. Individuals and Countries That are Active

The results of academic databases show that combinatorial interaction testing has been extensively researched. Furthermore, it is evident that numerous researchers have collaborated in this study. Researchers who wrote more than three publications were the most productive, and almost (47%) of all publications were created by scholars from the disciplines. A visual representation of these researchers' representations is shown in *Fig. 7*. According to the survey results, the leaders are the four best of ‘‘Zamli, Kamal Z.” ‘‘Alsewari, Abdul Rahman A.” ” Younis, Mohammed I.” and ‘‘Ahmed, Bestoun S." with 37, 19, and 16 published papers, respectively. More than one-quarter of all publications were created by these four authors. *Fig. 8* shows the combinatorial testing methodology, which refers to countries that publish the most papers on the subject. As seen in the lineup, the bar shows the author's nation of affiliation in the articles. Malaysia, the United States, India, and Iraq are clearly the top four

Received 06/February/2022; Accepted 04/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

countries in terms of publications, with 37, 20, 19, and 17 publications, respectively. For example, 37 of the Malaysian papers were co-authored by ‘‘Zamli, Kamal Z.” ‘‘Alsewari, Abdul Rahman A." and ‘‘Ahmed, Bestoun S.” ” Nasser Abdullah,” “Alazzawi, Ammar K.” The United States is the second-best organization that has participated most in combinatorial testing. Three researchers collaborated on this, from the United States “Kuhn, D. Richard,” “Kacker, Raghu N.,” and “Lei, Yu,” as well as other groups. Both India and Iraq opted for combinatorial testing in the third and fourth places, respectively. This is a partnership between researchers, based on an experiment that showed an enormous advantage of teamwork between "Sabharwal, Sangeeta,” "Mittal, Nitish” "Bansal, Priti" and " Younis, Mohammed I.” ” Hasan, Imad H." and "Potrus, Moayad Y." Combinatorial interaction testing is one of critical and tremendous significance, yet few academics contribute to the papers on it. In this scenario, 60 current active authors are directly responsible for two-thirds of the publications, and also contribute to more than 43% of all papers concerned.

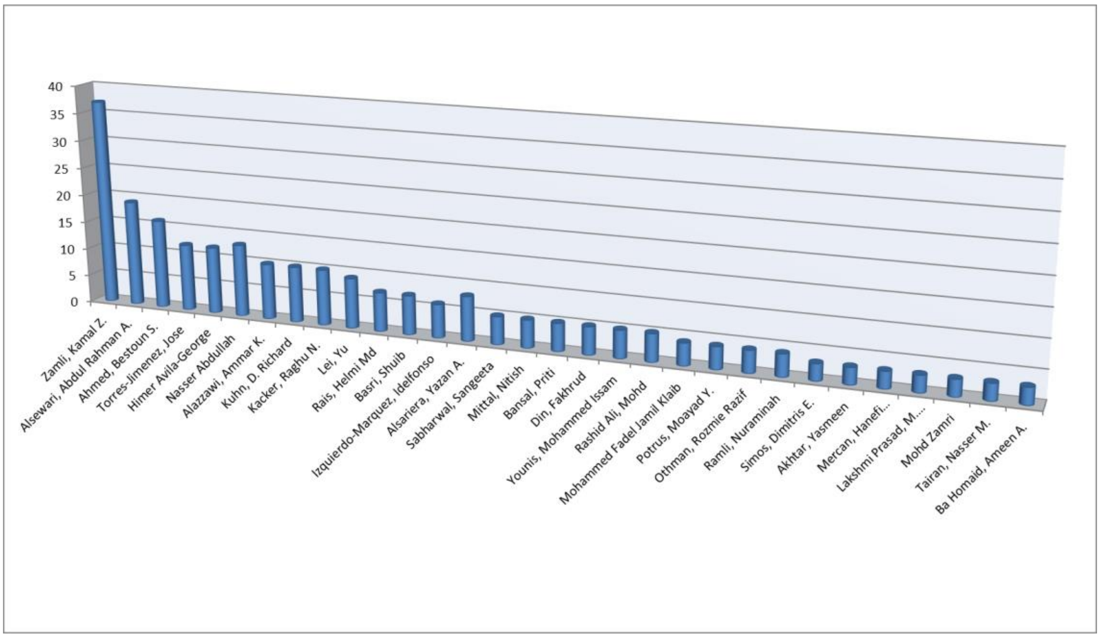

FIG. 7. ACTIVE RESEARCHERS.

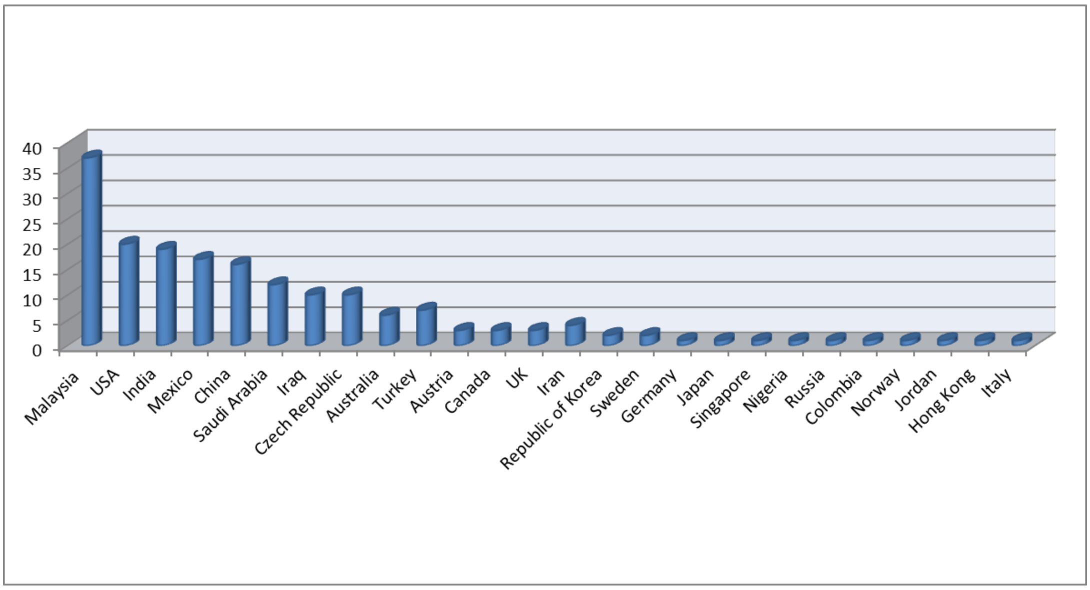

FIG. 8. ACTIVE COUNTRIES.

Received 06/February/2022; Accepted 04/May/2022

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

VI. EVALUATION AND DISCUSSION

The discussion included a variety of themes and subjects. Despite the varied subjects, this analysis focuses on four main aspects. Categories ranging from conception highlight the article topic of the various publications, which are published in the categorized sections, as shown in *Fig. 9*. These subsections describe other categories and their relevant publications in further detail. The study results were distinct based on analysis of the study findings.

A. Studies on Test Case Generation

Several methods for generating a combinatorial interaction test suite have been discussed in detail. Regarding unique use cases, papers may provide personalized techniques. However, many studies have used various methods of generation. It is imperative to grasp the processes and techniques used to handle and cope with limits; it is equally important to recognize ways and approaches for generating ideas. Table V. examines methods for generating case scenarios and assesses strategies.

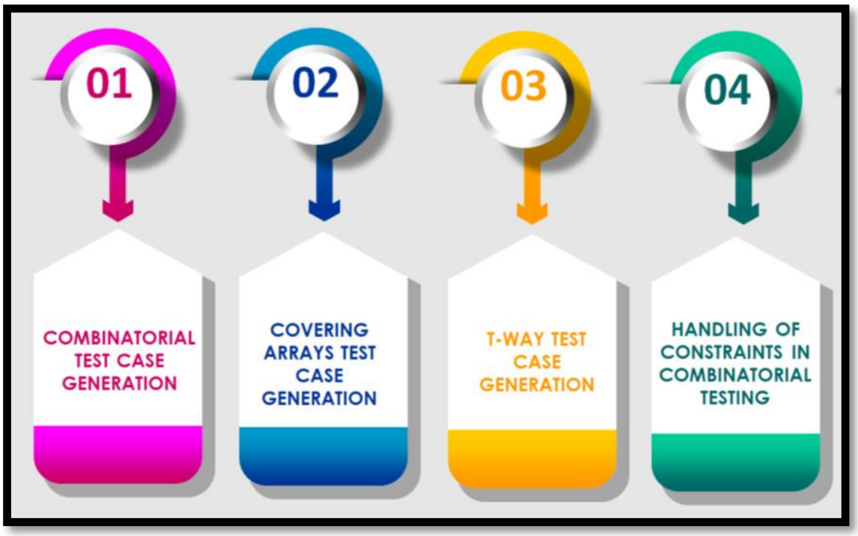

FIG. 9. CATEGORIES OF TEST CASE GENERATION.

TABLE V. PERFORMANCE COMPARISON OF DIFFERENT CLASSIFIERS

CATEGORIES	PAPERS
COMBINATORIAL TEST CASE GENERATION	[39], [40], [41], [42], [43], [44], [45], [46], [47]
COVERING ARRAYS TEST CASE GENERATION	, [17] ,[48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [67], [66], [67], [68], [69], [70], [71], [72], [73], [74]
T-WAY TEST CASE ENERATION	[44] ,[57] ,[75], [76], [79], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87]
CONSTRAINTS HANDLING IN COMBINATORIAL TESTING	[61] ,[88], [89], [90], [91]

B. Existing Strategies, and Methods of Fault Localization

Debugging software requires fault localization, which is costly and time intensive. The right fault localization method is needed to assist programmers in locating errors with minimum manual intervention. Owing to this requirement, many fault localization techniques have been developed. Fault localization is an essential avenue for increasing the capacity of combinatorial interaction testing. The fundamental task in CT is to locate the failure-inducing interactions.

Received 06/February/2022; Accepted 04/May/2022

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

Combinatorial interaction fault localization attempts to separate one or more t-way interaction faults from pass/fail data. According to the interdependencies between new test cases and running outcomes, fault localization approaches studied in combinatorial testing can be classified as adaptive or non-adaptive methods [92]. Fault localization in combinatorial testing is briefly discussed in the next section. Li *et al*. [93] discussed fault localization utilizing the findings of combinatorial testing and a technique based on the delta-debugging concept of isolation. The fault localization tool BEN, developed by Ghandehari *et al.* [94] , is based on computed tomography. BEN's primary concept is built on three distinct forms of suspicion: component, combination, and environmental. It operates on the premise that such a combinatorial test set has indeed been run, and the status of each test case's execution has been recorded. New tests can be developed to provide extra information needed for combinatorial fault localization to identify failure-inducing interaction faults in an adaptable manner. A variety of methodologies and methods for combinatorial fault localization are discussed in the following publications [95], [96], [97], [98], [99], [100], [101].

C. Support for Mixed Covering Arrays

Fig. 10 shows the tools/algorithms that work with regular arrays, as well as those that specifically have a design that exhibits support for mixed arrays. Researchers have found that 19% of the tools provide mixed covering arrays, meaning that 42% of the parameters are free to be supplied, and have various numbers of values, while 3% require exactly one to be provided, if any at all. This visualization further explains the use of different construction techniques that can be used after expansion.

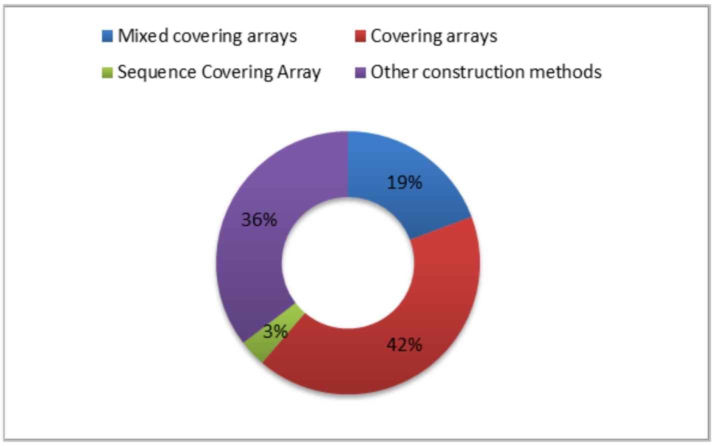

FIG. 10. SUPPORT FOR MIXED COVERING ARRAYS.

VII. THREATENINGS TO VALIDITY

This investigation confronts certain inherent difficulties, similar to any other type of investigation. The widespread application of well-known suggestions and guidance for literature research has made it easy to eliminate a number of hazards. To begin, as indicated in the paper, numerous search strings were used to provide maximal cross-paraphrase correlation in order to obtain similar literature. However, even though most related publications have been cited, this effort does not aim to provide thorough coverage of the papers. Although the search terms discovered some documents, numerous pilots agreed to follow these procedures, including a preliminary search for more than a handful of papers were executed in order to protect the integrity of the research, and a snowball search for other articles was performed to decrease the chance of discovery. Second, in research investigations that employ a single-author approach, individual author bias during data extraction can arise. To reduce this

Received 06/February/2022; Accepted 04/May/2022

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

threat, all authors checked and reputed their work as individuals before it was submitted for peer review. In addition, the spreadsheet utilized automatic data extraction and verification techniques to double-check the conclusions of the data extraction. Third, several studies were not included in this study.

The selection criteria are stated to validate the outcomes of the data extraction procedure described in Section III. In addition, a number of studies were omitted from this study. In addition, certain research may be eliminated because of the limitations of the study. The study in this field was dedicated to combinatorial interaction tests. This means that articles that do not focus on combinatorial interactions are overlooked by combinatorial testing scientists. Various studies have been conducted on these documents [8,9]. This was also discussed because of the significance of combinatorial interaction testing for the implementation of the constraint control flow. Unexpected and non-electronic contributions were ignored, while only electronic contributions were sought; thus, a combinatorial interaction researcher is likely to be interested in only work in this area.

VIII. CONCLUSIONS

This paper presents a detailed study of research studies performed from 2010 to 2021, focusing on 97 studies. The results of research investigations have revealed the growing use of combinatorial testing in the literature. A wide range of combinatorial studies have been referenced to present organizations and active researchers in the field, as well as presenting classification categories, such as combinatorial studies and model validation. Rather, in testing out the various techniques, the researchers considered all possible test matrices, which included the combinatorial test. In recent years, research into combinatorial testing applications has received considerable attention. The purpose of this study was to showcase CT and report interaction testing for current software applications. The research begins by illustrating many forms, notations, and constructions. Next, briefly discuss several CT algorithms. For the foreseeable future, this field of study appears to be an active, density-based path. To summarize, this article discusses many forms of interactions that may be utilized for interaction testing. Additionally, this article evaluates a variety of existing methods in terms of the types of interactions they support. Finally, this study discusses a practical example in which many forms of interactions are exhibited inside a single SUT. Consequently, it may be stated that all forms of contact were successful. Other studies are also called for in the fields of test identification and maintenance, which are both difficult to achieve. Future studies will focus on conducting deeper research on the influence of the t-way in CT and designing larger-scale CT test generators capable of handling DL systems. Additionally, we would like to further investigate the topic of a combination of optimization strategies, specifically with regard to the limited dataset, where both shortening runtime and achieving equivalent behavior is the key.

REFERENCES

[1] S. A. Dheyab, M. N. Abdullah, and B. F. Abed, "A novel approach for big data processing using message passing interface based on memory mapping," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0275-3.

[2] W. H. Abdulsalam, R. S. Alhamdani, and M. N. Abdullah, "Emotion recognition system based on hybrid techniques," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 4, 2019, doi: 10.18178/ijmlc.2019.9.4.831.

[3] W. H. Abdulsalam, R. S. Alhamdani, and M. N. Abdullah, "Speech emotion recognition using minimum extracted features," in *Proceedings - 2018 1st Annual International Conference on Information and Sciences, AiCIS 2018*, 2019, pp. 58–61, doi: 10.1109/AiCIS.2018.00023.

[4] D. R. Kuhn, R. N. Kacker, and Y. Lei, "Measuring and specifying combinatorial coverage of test input configurations," *Innov. Syst. Softw. Eng.*, vol. 12, no. 4, pp. 249–261, 2016, doi: 10.1007/s11334-015-0266-2.

[5] M. I. Younis, "MVSCA: Multi-Valued Sequence Covering Array," *J. Eng.*, vol. 25, no. 11, pp. 82–91, 2019, doi: 10.31026/j.eng.2019.11.07.

[6] I. H. Hasan and M. Y. Potrus, "Improved Mixed Neighborhood Tabu Search by Random Selection for Combinatorial Interaction Testing," *Zanco J. Pure Appl. Sci.*, vol. 32, no. 5, pp. 1–19, 2020, doi:

Received 06/February/2022; Accepted 04/May/2022

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

10.21271/zjpas.32.5.1.

[7] S. Anand *et al.*, "An orchestrated survey of methodologies for automated software test case generation," *J. Syst. Softw.*, vol. 86, no. 8, pp. 1978–2001, 2013, doi: 10.1016/j.jss.2013.02.061.

[8] A. A. Al-Sewari and K. Z. Zamli, "An orchestrated survey on T-way test case generation strategies based on optimization algorithms," in *Lecture Notes in Electrical Engineering*, 2014, vol. 291 LNEE, doi: 10.1007/978-981-4585-42-2_30.

[9] V. Chandra Prakash, S. Tatale, V. Kondhalkar, and L. Bewoor, "A critical review on automated test case generation for conducting combinatorial testing using particle swarm optimization," *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 22–28, 2018, doi: 10.14419/ijet.v7i3.8.15212.

[10] J. Charles, Y. Meng, and J. Zhou, "Sequence covering arrays," 2013.

[11] D. R. Kuhn, R. N. Kacker, and Y. Lei, "Practical combinatorial testing," *NIST Spec. Publ.*, vol. 800, 2010.

[12] C. P. Vudatha, S. Nalliboena, S. K. R. Jammalamadaka, B. K. K. Duvvuri, and L. S. S. Reddy, "Automated generation of test cases from output domain of an embedded system using Genetic algorithms," in *ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology*, 2011, vol. 5, doi: 10.1109/ICECTECH.2011.5941989.

[13] K. Z. Zamli, B. S. Ahmed, T. Mahmoud, and W. Afzal, "Fuzzy adaptive tuning of a particle swarm optimization algorithm for variable-strength combinatorial test suite generation," *arXiv*. 2018, doi: 10.1049/pbce119h_ch22.

[14] R. N. Kacker, D. R. Kuhn, Y. Lei, and D. E. Simos, "Factorials Experiments, Covering Arrays, and Combinatorial Testing," *Math. Comput. Sci.*, 2021, doi: 10.1007/s11786-021-00502-7.

[15] C. Nie and H. Leung, "A survey of combinatorial testing," *ACM Comput. Surv.*, vol. 43, no. 2, pp. 1–29, 2011, doi: 10.1145/1883612.1883618.

[16] V. V. Kuliamin and A. A. Petukhov, "A survey of methods for constructing covering arrays," *Program. Comput. Softw.*, vol. 37, no. 3, 2011, doi: 10.1134/S0361768811030029.

[17] B. S. Ahmed and K. Z. Zamli, "A review of covering arrays and their application to software testing," *J. Comput. Sci.*, vol. 7, no. 9, 2011, doi: 10.3844/jcssp.2011.1375.1385.

[18] J. Torres-Jimenez and I. Izquierdo-Marquez, "Survey of covering arrays," *Proc. - 15th Int. Symp. Symb. Numer. Algorithms Sci. Comput. SYNASC 2013*, no. 1, pp. 20–27, 2013, doi: 10.1109/SYNASC.2013.10.

[19] H. Wu and C. Nie, "An overview of search based combinatorial testing," *7th Int. Work. Search-Based Softw. Testing, SBST 2014 - Proc.*, pp. 27–30, 2014, doi: 10.1145/2593833.2593839.

[20] S. K. Khalsa and Y. Labiche, "An orchestrated survey of available algorithms and tools for combinatorial testing," *Proc. - Int. Symp. Softw. Reliab. Eng. ISSRE*, no. February 2017, pp. 323–334, 2014, doi: 10.1109/ISSRE.2014.15.

[21] R. E. Lopez-Herrejon, S. Fischer, R. Ramler, and A. Egyed, "A first systematic mapping study on combinatorial interaction testing for software product lines," 2015, doi: 10.1109/ICSTW.2015.7107435.

[22] A. R. A. Alsewari, N. M. Tairan, and K. Z. Zamli, "Survey on input output relation based combination test data generation strategies," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 18, pp. 8427–8430, 2015.

[23] C. A. Cobos-lozada and J. Torres-jimenez, "Metaheuristic algorithms for building Covering Arrays : A review Algoritmos metaheurísticos para construir Covering Arrays : Revisión Algoritmos metaheurísticos para construir Covering Arrays :," *Rev. Fac. Ing.*, vol. 25, no. 43, pp. 31–45, 2016.

[24] N. Ramli, R. R. Othman, and M. S. A. R. Ali, "Optimizing combinatorial input-output based relations testing using Ant Colony algorithm," *2016 3rd Int. Conf. Electron. Des. ICED 2016*, pp. 586–590, 2017, doi: 10.1109/ICED.2016.7804713.

[25] K. Z. Zamli, A. R. A. Alsewari, and M. I. Younis, "T-way testing strategies: Issues, challenges, and practices," in *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications*, 2017.

[26] B. S. Ahmed, K. Z. Zamli, W. Afzal, and M. Bures, "Constrained interaction testing: A systematic literature study," *IEEE Access*, vol. 5, pp. 25706–25730, 2017, doi: 10.1109/ACCESS.2017.2771562.

[27] Z. Li, Y. Chen, G. Gong, D. Li, K. Lv, and P. Chen, "A Survey of the Application of Combinatorial Testing," *Proc. - Companion 19th IEEE Int. Conf. Softw. Qual. Reliab. Secur. QRS-C 2019*, no. 2018, pp. 512–513, 2019, doi: 10.1109/QRS-C.2019.00100.

[28] J. M. Altmemi, R. R. Othman, and R. Ahmad, "Review of combinatorial testing strategy," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 6, pp. 2877–2881, 2019, doi: 10.30534/ijatcse/2019/31862019.

[29] N. Setiani, R. Ferdiana, P. I. Santosa, and R. Hartanto, "Literature review on test case generation approach," in *ACM International Conference Proceeding Series*, Jan. 2019, pp. 91–95, doi: 10.1145/3305160.3305186.

[30] H. Wu, C. Nie, J. Petke, Y. Jia, and M. Harman, "A survey of constrained combinatorial testing," *arXiv*,

Received 06/February/2022; Accepted 04/May/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

pp. 1–16, 2019.

[31] G. Naseri and M. A. G. Koffas, "Application of combinatorial optimization strategies in synthetic biology," *Nat. Commun.*, vol. 11, no. 1, pp. 1–14, 2020, doi: 10.1038/s41467-020-16175-y.

[32] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, 2007, doi: 10.1016/j.jss.2006.07.009.

[33] B. Kitchenham and S. Charters, "B. Kitchenham and S. Charters, 'Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3,'" *Engineering*, vol. 45, no. 4ve, 2007.

[34] P. Cronin, F. Ryan, and M. Coughlan, "Undertaking a literature review: a step-by-step approach.," *British journal of nursing (Mark Allen Publishing)*, vol. 17, no. 1. 2008, doi: 10.12968/bjon.2008.17.1.28059.

[35] T. Dybå, T. Dingsøyr, and G. K. Hanssen, "Applying systematic reviews to diverse study types: An experience report," 2007, doi: 10.1109/ESEM.2007.21.

[36] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, 2015, doi: 10.1016/j.infsof.2015.03.007.

[37] S. Jalali and C. Wohlin, "Systematic literature studies: Database searches vs. backward snowballing," 2012, doi: 10.1145/2372251.2372257.

[38] F. Q. B. Da Silva, A. L. M. Santos, S. Soares, A. C. C. Frana, C. V. F. Monteiro, and F. F. MacIel, "Six years of systematic literature reviews in software engineering: An updated tertiary study," *Inf. Softw. Technol.*, vol. 53, no. 9, pp. 899–913, 2011, doi: 10.1016/j.infsof.2011.04.004.

[39] X. Pan and H. Chen, "Using organizational evolutionary particle swarm techniques to generate test cases for combinatorial testing," *Proc. - 2011 7th Int. Conf. Comput. Intell. Secur. CIS 2011*, pp. 1580–1583, 2011, doi: 10.1109/CIS.2011.354.

[40] J. Wang and S. Jiang, "An improved algorithm for test data generation based on particle swarm optimization," *Proc. - 2011 Int. Conf. Instrumentation, Meas. Comput. Commun. Control. IMCCC 2011*, pp. 404–407, 2011, doi: 10.1109/IMCCC.2011.108.

[41] A. B. Nasser, Y. A. Alsariera, K. Z. Zamli, and B. Al-Kazemi, "Late Acceptance Hill Climbing Based Strategy for Addressing Constraints Within Combinatorial Test Data Generation," *Unpublished*, no. September, pp. 1–6, 2014, doi: 10.13140/rg.2.1.4986.5681.

[42] A. Kalaee and V. Rafe, "An Optimal Solution for Test Case Generation Using ROBDD Graph and PSO Algorithm," *Qual. Reliab. Eng. Int.*, vol. 32, no. 7, 2016, doi: 10.1002/qre.1934.

[43] O. Sahin and B. Akay, "Comparisons of metaheuristic algorithms and fitness functions on software test data generation," *Appl. Soft Comput. J.*, vol. 49, pp. 1202–1214, 2016, doi: 10.1016/j.asoc.2016.09.045.

[44] M. I. Younis, "DEO: A dynamic event order strategy for T-way sequence covering array test data generation," *Baghdad Sci. J.*, vol. 17, no. 2, pp. 575–582, 2020, doi: 10.21123/bsj.2020.17.2.0575.

[45] K. Z. Bin Zamli, M. F. Jamil Klaib, M. I. Younis, O. H. Yeh, and R. Klaib, M.F.J., Zamli, K.Z., Isa, N.A.M., Younis, M.I. and Abdullah, "G2WAY: A PAIRWISE TEST DATA GENERATION STRATEGY WITH AUTOMATED EXECUTION SUPPORT," *J. ICT, 9, pp 59 – 85*, pp. 59–85, 2010, doi: 10.32890/jict.9.2010.8100.

[46] L. Cai, Y. Zhang, and W. Ji, "Variable strength combinatorial test data generation using enhanced bird swarm algorithm," 2018, doi: 10.1109/SNPD.2018.8441104.

[47] W. Zhang and Q. Zhou, "Software test data generation technology based on polymorphic particle swarm evolutionary algorithm," *J. Intell. Fuzzy Syst.*, pp. 1–13, Mar. 2021, doi: 10.3233/jifs-189811.

[48] K. Z. Zamli and M. I. Younis, "Interaction testing: From pairwise to variable strength interaction," 2010, doi: 10.1109/AMS.2010.15.

[49] K. Z. Zamli, R. R. Othman, M. I. Younis, and M. H. Mohamed Zabil, "Practical adoptions of T-way strategies for interaction testing," *Commun. Comput. Inf. Sci.*, vol. 181 CCIS, no. PART 3, pp. 1–14, 2011, doi: 10.1007/978-3-642-22203-0_1.

[50] and K. Z. Z. AbdulRahman A. Alsewari, Mohammed I. Younis, "Generation of Pairwise Test Sets using a Harmony Search Algorithm," *Int. J. Comput. Sci. Lett.*, vol. 3, p. 283, 2011.

[51] B. S. Ahmed, K. Z. Zamli, and C. P. Lim, "Application of Particle Swarm Optimization to uniform and variable strength covering array construction," *Appl. Soft Comput. J.*, vol. 12, no. 4, pp. 1330–1347, 2012, doi: 10.1016/j.asoc.2011.11.029.

[52] B. S. Ahmed and K. Z. Zamli, "PSTG: A t-way strategy adopting particle Swarm Optimization," *AMS2010 Asia Model. Symp. 2010 - 4th Int. Conf. Math. Model. Comput. Simul.*, pp. 1–5, 2010, doi: 10.1109/AMS.2010.14.

[53] A. Rodriguez-Cristerna and J. Torres-Jimenez, "A Simulated Annealing with Variable Neighborhood Search Approach to Construct Mixed Covering Arrays," *Electron. Notes Discret. Math.*, vol. 39, pp.

Received 06/February/2022; Accepted 04/May/2022

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

249–256, Dec. 2012, doi: 10.1016/j.endm.2012.10.033.

[54] H. Wu, C. Nie, F. C. Kuo, H. Leung, and C. J. Colbourn, “A Discrete Particle Swarm Optimization for Covering Array Generation,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 575–591, 2015, doi: 10.1109/TEVC.2014.2362532.

[55] T. Mahmoud and B. S. Ahmed, “An efficient strategy for covering array construction with fuzzy logic-based adaptive swarm optimization for software testing use,” *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8753–8765, 2015, doi: 10.1016/j.eswa.2015.07.029.

[56] H. Avila-George, J. Torres-Jimenez, V. Hernández, and L. Gonzalez-Hernandez, “Simulated annealing for constructing mixed covering arrays,” in *Advances in Intelligent and Soft Computing*, 2012, vol. 151 AISC, pp. 657–664, doi: 10.1007/978-3-642-28765-7_79.

[57] S. Sabharwal, P. Bansal, and N. Mittal, “Construction of t-way covering arrays using genetic algorithm,” *Int. J. Syst. Assur. Eng. Manag.*, vol. 8, no. 2, pp. 264–274, 2017, doi: 10.1007/s13198-016-0430-6.

[58] P. Galinier, S. Kpodjedo, and G. Antoniol, “A penalty-based Tabu search for constrained covering arrays,” 2017, doi: 10.1145/3071178.3071324.

[59] L. Gonzalez-Hernandez and J. Torres-Jimenez, “MiTS: A new approach of tabu search for constructing mixed covering arrays,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6438 LNAI, no. PART 2, pp. 382–393, doi: 10.1007/978-3-642-16773-7_33.

[60] L. Gonzalez-Hernandez, N. Rangel-Valdez, and J. Torres-Jimenez, “CONSTRUCTION of MIXED COVERING ARRAYS of STRENGTHS 2 THROUGH 6 USING A TABU SEARCH APPROACH,” *Discret. Math. Algorithms Appl.*, vol. 4, no. 3, Sep. 2012, doi: 10.1142/S1793830912500334.

[61] H. Liu, F. Ma, and J. Zhang, “Generating covering arrays with pseudo-boolean constraint solving and balancing heuristic,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9810 LNCS, doi: 10.1007/978-3-319-42911-3_22.

[62] B. Garn and D. E. Simos, “Algebraic Techniques for Covering Arrays and related Structures,” *Electron. Notes Discret. Math.*, vol. 70, pp. 49–54, 2018, doi: 10.1016/j.endm.2018.11.008.

[63] Y. Akhtar, S. Maity, and R. C. Chandrasekharan, “Covering arrays of strength four and software testing,” *Springer Proc. Math. Stat.*, vol. 139, no. January, pp. 391–398, 2015, doi: 10.1007/978-81-322-2452-5_26.

[64] S. Maity, Y. Akhtar, R. C. Chandrasekharan, and C. J. Colbourn, “Improved Strength Four Covering Arrays with Three Symbols,” *Graphs Comb.*, vol. 34, no. 1, pp. 223–239, 2018, doi: 10.1007/s00373-017-1861-9.

[65] J. M. Balera and V. A. de Santiago Júnior, “T-tuple reallocation: An algorithm to create mixed-level covering arrays to support software test case generation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9158, doi: 10.1007/978-3-319-21410-839.

[66] S. Raaphorst, L. Moura, and B. Stevens, “Variable strength covering arrays,” *J. Comb. Des.*, vol. 26, no. 9, pp. 417–438, Sep. 2018, doi: 10.1145/3305160.3305186.

[67] J. Torres-Jimenez, I. Izquierdo-Marquez, A. Gonzalez-Gomez, and H. Avila-George, “A branch & bound algorithm to derive a direct construction for binary covering arrays,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9413, no. October, pp. 158–177, 2015, doi: 10.1007/978-3-319-27060-9_13.

[68] M. I. Younis, “Gamipog: A deterministic genetic multi-parameter-order strategy for the generation of variabLE STRENGTH COVERING ARRAYS,” *J. Eng. Sci. Technol.*, vol. 15, no. 5, pp. 3142–3162, 2020.

[69] J. Timaná, C. Cobos, and J. Torres-Jimenez, “Memetic algorithm for constructing covering arrays of variable strength based on global-best harmony search and simulated annealing,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11288 LNAI, doi: 10.1007/978-3-030-04491-6_2.

[70] X. Chen, Q. Gu, J. Qi, and D. Chen, “Applying particle swarm optimization to pairwise testing,” *Proc. - Int. Comput. Softw. Appl. Conf.*, pp. 107–116, 2010, doi: 10.1109/COMPSAC.2010.17.

[71] R. J. Zha, D. P. Zhang, C. H. Nie, and B. W. Xu, “Test data generation algorithms of combinatorial testing and comparison based on cross-entropy and particle swarm optimization method,” *Jisuanji Xuebao/Chinese J. Comput.*, vol. 33, no. 10, 2010, doi: 10.3724/SP.J.1016.2010.01896.

[72] Y. A. Alsariera, M. A. Majid, and K. Z. Zamli, “A Bat-inspired strategy for pairwise testing,” *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 18, pp. 8500–8506, 2015.

[73] M. F. J. Klaib, S. Muthuraman, N. Ahmad, and R. Sidek, “Tree based test case generation and cost

Received 06/February/2022; Accepted 04/May/2022

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

calculation strategy for uniform parametric pairwise testing," *J. Comput. Sci.*, vol. 6, no. 5, pp. 542–547, 2010, doi: 10.3844/jcssp.2010.542.547.

[74] 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, pp. 1–24, 2018, doi: 10.1371/journal.pone.0195187.

[75] Z. H. C. Soh, S. A. C. Abdullah, K. Z. Zamli, and M. I. Younis, "Distributed T-way test suite data generation using exhaustive search method with map and reduce framework," 2010, doi: 10.1109/ISIEA.2010.5679443.

[76] K. Z. Zamli, M. F. J. Klaib, M. I. Younis, N. A. M. Isa, and R. Abdullah, "Design and implementation of a t-way test data generation strategy with automated execution tool support," *Inf. Sci. (Ny).*, vol. 181, no. 9, 2011, doi: 10.1016/j.ins.2011.01.002.

[77] B. S. Ahmed, K. Z. Zamli, and C. P. Lim, "Application of Particle Swarm Optimization to uniform and variable strength covering array construction," *Appl. Soft Comput. J.*, vol. 12, no. 4, pp. 1330–1347, 2012, doi: 10.1016/j.asoc.2011.11.029.

[78] B. S. Ahmed, K. Z. Zamli, and C. P. Lim, "Constructing a t-way interaction test suite using the Particle Swarm Optimization approach," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 1 A, pp. 431–451, 2012.

[79] Priyanka, I. Chana, and A. Rana, "An effective approach to build optimal T-way interaction test suites over cloud using particle swarm optimization," *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng.*, vol. 108 LNICST, pp. 193–198, 2012, doi: 10.1007/978-3-642-35615-5_28.

[80] Z. Zhang, J. Yan, Y. Zhao, and J. Zhang, "Generating combinatorial test suite using combinatorial optimization," *J. Syst. Softw.*, vol. 98, pp. 191–207, Dec. 2014, doi: 10.1016/j.jss.2014.09.001.

[81] 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," *Proc. 2015 10th IEEE Conf. Ind. Electron. Appl. ICIEA 2015*, pp. 123–128, 2015, doi: 10.1109/ICIEA.2015.7334096.

[82] K. Z. Zamli, F. Din, G. Kendall, and B. S. Ahmed, "An experimental study of hyper-heuristic selection and acceptance mechanism for combinatorial t-way test suite generation," *Inf. Sci. (Ny).*, vol. 399, 2017, doi: 10.1016/j.ins.2017.03.007.

[83] 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, vol. 1073, doi: 10.1007/978-3-030-33582-3_34.

[84] 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," *ACM Int. Conf. Proceeding Ser.*, no. February, pp. 105–109, 2020, doi: 10.1145/3384544.3384608.

[85] A. K. Alazzawi, H. M. Rais, and S. Basri, *Hybrid Artificial Bee Colony Algorithm for t-Way Interaction Test Suite Generation*, vol. 984. Springer International Publishing, 2019.

[86] K. Z. Zamli, B. Y. Alkazemi, and G. Kendall, "A Tabu Search hyper-heuristic strategy for t-way test suite generation," *Appl. Soft Comput. J.*, vol. 44, pp. 57–74, 2016, doi: 10.1016/j.asoc.2016.03.021.

[87] A. B. Nasser, F. Hujainah, A. A. Alsewari, and K. Z. Zamli, "Sequence and sequence-less T-way test suite generation strategy based on flower pollination algorithm," *2015 IEEE Student Conf. Res. Dev. SCOReD 2015*, pp. 676–680, 2015, doi: 10.1109/SCORED.2015.7449424.

[88] 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, pp. 20–36, Jun. 2017, doi: 10.1016/j.infsof.2017.02.004.

[89] Y. Sheng, C. Wei, and S. Jiang, "Constraint Test Cases Generation Based on Particle Swarm Optimization," *Int. J. Reliab. Qual. Saf. Eng.*, vol. 24, no. 5, Oct. 2017, doi: 10.1142/S0218539317500218.

[90] L. Yu, F. Duan, Y. Lei, R. N. Kacker, and D. R. Kuhn, "Constraint handling in combinatorial test generation using forbidden tuples," *2015 IEEE 8th Int. Conf. Softw. Testing, Verif. Valid. Work. ICSTW 2015 - Proc.*, 2015, doi: 10.1109/ICSTW.2015.7107441.

[91] F. Duan, Y. Lei, R. N. Kacker, and D. R. Kuhn, "An approach to T-Way test sequence generation with constraints," *Proc. - 2019 IEEE 12th Int. Conf. Softw. Testing, Verif. Valid. Work. ICSTW 2019*, pp. 241–250, 2019, doi: 10.1109/ICSTW.2019.00059.

[92] R. Jayaram and R. Krishnan, "Approaches to fault localization in combinatorial testing: A survey," in *Smart Innovation, Systems and Technologies*, 2018, vol. 78, doi: 10.1007/978-981-10-5547-8_55.

[93] J. Li and C. Nie, "Improved Delta Debugging Based on Combinatorial Testing," pp. 1–4, 2012, doi: 10.1109/QSIC.2012.28.

[94] L. S. Ghandehari, J. Chandrasekaran, Y. Lei, R. Kacker, and D. R. Kuhn, "BEN : A Combinatorial Testing-Based Fault Localization Tool," no. Iwct, pp. 4–7, 2015.

[95] A. Testing, "Characterizing Failure-Causing Parameter Interactions by Categories and Subject

Received 06/February/2022; Accepted 04/May/2022

DOI: https://doi.org/10.33103/uot.ijccce.22.4.6

Descriptors," pp. 331–341.

[96] C. J. Colbourn and V. R. Syrotiuk, "Coverage , Location , Detection , and Measurement," 2016, doi: 10.1109/ICSTW.2016.38.

[97] S. A. Seidel, K. Sarkar, C. J. Colbourn, and V. R. Syrotiuk, "Separating interaction effects using locating and detecting arrays," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 10979 LNCS, doi: 10.1007/978-3-319-94667-2_29.

[98] L. S. Ghandehari, Y. Lei, R. Kacker, R. Kuhn, T. Xie, and D. Kung, "A Combinatorial Testing-Based Approach to Fault Localization," vol. 5589, no. c, pp. 1–31, 2018, doi: 10.1109/TSE.2018.2865935.

[99] X. Niu, C. Nie, J. Y. Lei, H. Leung, and X. Wang, "Identifying Failure-Causing Schemas in the Presence of Multiple Faults," *IEEE Trans. Softw. Eng.*, vol. 46, no. 2, pp. 141–162, Feb. 2020, doi: 10.1109/TSE.2018.2844259.

[100] X. Niu *et al.*, "An Interleaving Approach to Combinatorial Testing and Failure-Inducing Interaction Identification," *IEEE Trans. Softw. Eng.*, vol. 46, no. 6, pp. 584–615, 2020, doi: 10.1109/TSE.2018.2865772.

[101] H. Jin and T. Tsuchiya, "Constrained locating arrays for combinatorial interaction testing," *J. Syst. Softw.*, vol. 170, p. 110771, 2020, doi: 10.1016/j.jss.2020.110771.

Received 06/February/2022; Accepted 04/May/2022