Borneo Engineering & Advanced Multidisciplinary International Journal (BEAM)
Volume 4, Issue 1, May 2025, Pages 24-30

Advancements in t-way Interaction Test Generation Strategies: A Comprehensive Review (2020-2024)

Full Paper

Tengku Diyana Tengku Ibrahim[1]*

[1]Politeknik Muadzam Shah, Lebuhraya Tun Razak, 26700 Muadzam Shah, Pahang, Malaysia

*Corresponding author: tengkudiyana@pms.edu.my
Please provide an **official organisation email** of the corresponding author

Article history
Received
23 January 2025
Received in revised form
18 February 2025
Accepted
3 March 2025
Published online
1 May 2025

Abstract

T-way interaction testing is a critical method in software testing, aimed at covering all t-way combinations of input parameters to identify interaction faults effectively. In recent years, from 2020 to 2024, numerous advanced t-way test generation strategies have been proposed to enhance test coverage, efficiency, and scalability. This research review examines comprehensively these recent strategies, providing an in-depth analysis of their objectives, methodologies and outcomes. Through a systematic literature review, this study evaluates the most prominent algorithms, highlighting significant improvements in test generation time, fault detection capabilities and applicability to complex software systems. The findings reveal notable advancements in addressing key challenges in combinatorial testing, offering insights into the evolving landscape of t-way test generation. This review concludes by summarizing the contributions of these recent strategies and proposing directions for future research to further enhance the effectiveness of t-way interaction testing.

Keywords: - T-way interaction testing, software testing, test generation strategies, combinatorial testing, fault detection

1. Introduction

Software testing is an indispensable phase in the software development lifecycle, ensuring the delivery of reliable and high-quality software products. Among various testing methodologies, t-way interaction testing plays a vital role by systematically covering all possible t-way combinations of input parameters to uncover interaction faults. These faults, which arise due to specific interactions between input parameters, can lead to severe software failures if not detected early (Kuhn et al., 2017). Traditional testing methods often fail to address this issue comprehensively, necessitating the adoption of combinatorial testing techniques such as t-way interaction testing (Chen, 2023). The concept of t-way interaction testing has evolved significantly over the years, driven by the need for more efficient and effective testing strategies. Initially, the focus was on 2-way or pairwise testing, which covers all possible pairs of input parameter values (Lei et al., 2007). However, as software systems grew more complex, the limitations of pairwise testing became apparent, leading to the development of higher strength t-way interaction testing. These advanced strategies aim to cover combinations involving three, four or more parameters, thereby providing a more thorough examination of potential interaction faults (Othman et al., 2011).

Despite the theoretical benefits of t-way interaction testing, practical challenges remain, particularly in the generation of test cases. The number of possible t-way combinations can be exponentially large, making it computationally intensive to generate and execute all test cases (Zamli et al., 2011). This challenge has spurred significant research interest in developing efficient t-way test generation strategies that balance test coverage and computational feasibility. This research review aims to explore and analyse the recent advancements in t-way test generation strategies published between 2020 and 2024. The study focuses on various algorithms and techniques proposed during this period, examining their objectives, methodologies and outcomes. By providing a comprehensive overview of these recent developments, this review seeks to highlight the progress made in the field and identify potential areas for future research. The primary objectives of this research review are to provide a detailed overview of the state-of-the-art t-way test generation strategies developed between 2020 and 2024. It aims to analyse the methodologies and algorithms proposed in these strategies, with a focus on their effectiveness, efficiency and applicability. Additionally, the review seeks to identify key trends, advancements and gaps in the current research landscape of t-way interaction testing. Ultimately, the review offers insights and recommendations for future research directions to further enhance the effectiveness and practicality of t-way test generation strategies.

2. Literature Review

In the past five years, significant advancements have been made in the field of t-way interaction test generation. Researchers have proposed various algorithms and techniques, each addressing different aspects of the test generation process. This section reviews the existing works on t-way test generation strategies. Table 1 presents a sequential account of these algorithms, highlighting their specific features and the trajectory of research progress in this field published from 2020 to 2024.

Table 1. Algorithms by year of publication

Year	Algorithm Name	Key Features
2020	Dynamic event order strategy (DEO)	Event-driven testing, dynamic event ordering, priority mechanism
2021	Elitist hybrid MBO-GA strategy (EMBO-GA)	Hybrid MBO-GA approach, elitist strategy, enhanced fault detection
2022	Tolerance structurally modified whale optimization (TSWOA)	Tolerance thresholds, modified search mechanism, high-dimensional handling
2023	Seeding and constraint support in IPOG (SCIPOG)	Seeding mechanism, constraint support, high-risk combination focus
2024	Improved asexual reproduction optimization (ImpARO)	Enhanced mutation and selection, self-adaptive approach, high fault detection

The review examines these algorithms in depth, providing a comprehensive analysis of their unique methodologies, key innovations and significant implications in the field of software testing. The paper delves into the intricate details of each algorithm, scrutinizing their underlying principles, strategies and the research advancements they represent. By highlighting the distinct features and contributions of these t-way interaction test generation approaches, the review offers valuable insights into the trajectory of progress in this critical area of software engineering.

2.1 2020: Dynamic Event Order Strategy (DEO)

The dynamic event order strategy (DEO) introduced in 2020 (Younis, 2020) focuses on dynamically adjusting the order of events during test case generation. DEO leverages event-driven testing techniques to identify critical t-way interactions and employs a priority mechanism to reorder events based on their impact on software behaviour. This strategy aims to reduce the number of test cases by eliminating non-essential event sequences, making it particularly suitable for event-driven software systems and applications.

2.2 2021: Elitist Hybrid MBO-GA Strategy (EMBO-GA)

In 2021, the elitist hybrid MBO-GA strategy (EMBO-GA) was proposed, combining the migrating birds optimization (MBO) algorithm with genetic algorithms (GA). EMBO-GA uses MBO for initial population generation and GA for optimization, implementing an elitist strategy to retain the best solutions across generations. This hybrid approach enhances fault detection and test coverage by leveraging the strengths of both algorithms, making it applicable to complex and large-scale software systems (Zakaria et al., 2021).

2.3 2022: Tolerance Structurally Modified Whale Optimization Algorithm (TSWOA)

The tolerance structurally modified whale optimization algorithm (TSWOA), introduced in 2022, adapts the whale optimization algorithm (WOA) with structural modifications for better performance (Hassan et al., 2022). TSWOA introduces tolerance thresholds to handle variability in software parameters and utilizes a modified search mechanism to explore the solution space more effectively. This algorithm is capable of handling high-dimensional and complex test scenarios, improving convergence speed and fault detection rates.

2.4 2023: Seeding and Constraint Support in In-Parameter-Order Generalized (SCIPOG)

In 2023, the seeding and constraint support in in-parameter-order generalized (SCIPOG) strategy enhanced the IPOG method with seeding and constraint support (Muazu et al., 2023). SCIPOG incorporates a seeding mechanism to initialize test cases with known critical values and supports complex constraints to ensure valid test case generation. This strategy reduces test case generation time by focusing on high-risk combinations, making it ideal for software systems with intricate parameter dependencies and constraints.

2.5 2024: Improved Asexual Reproduction Optimization (ImpARO)

The improved asexual reproduction optimization (ImpARO) algorithm, introduced in 2024, improves the asexual reproduction optimization (ARO) algorithm for test generation (Pira & Khodizadeh-Nahari, 2024). ImpARO introduces new mutation and selection mechanisms to enhance solution diversity and utilizes a self-adaptive approach to adjust algorithm parameters dynamically. This algorithm achieves high fault detection rates with fewer test cases and is suitable for diverse software systems, including those with dynamic behaviour.

3. Methodology

This research review employed a systematic literature review methodology to identify and evaluate recent advancements in t-way interaction test generation strategies published between 2020 and 2024. The methodology was adapted from established systematic review protocols, such as those proposed by Kitchenham et al. (2009), and tailored to fit the specific requirements of this study. The literature search focused on peer-reviewed journal articles, conference papers and significant technical reports relevant to t-way test generation. Multiple academic databases were used, including IEEE Xplore, ACM Digital Library and Google Scholar. The search terms included "t-way test generation," "combinatorial testing," "interaction testing algorithms," and specific names of the selected algorithms, such as "Dynamic Event Order Strategy" and "Elitist Hybrid MBO-GA Strategy." The inclusion criteria for selecting the literature were relevance to t-way interaction test generation strategies, publication date between 2020 and 2024, and high overall quality and citation impact.

Data collection involved extracting detailed information about each identified algorithm. The extracted data included the algorithm name, year of publication, key features, innovations, theoretical foundation, implementation details and performance metrics such as test generation time, fault detection rate and computational complexity. This comprehensive data collection ensured a robust basis for subsequent analysis. A combination of qualitative and quantitative analysis methods was employed to evaluate the identified algorithms. The qualitative analysis involved a detailed examination of the theoretical foundation and implementation details of each algorithm, as well as an assessment of their strengths and weaknesses based on their design and reported performance. The quantitative analysis included a comparative evaluation based on key performance metrics, allowing for statistical analysis to identify significant differences and trends in the performance of different algorithms.

Each algorithm was analysed in detail to understand its effectiveness and efficiency. The comparative evaluation focused on how well each algorithm balances test coverage with computational feasibility, its adaptability to different testing scenarios, and its overall contribution to advancing the field of t-way interaction testing. By synthesizing these findings, the study aimed to provide a comprehensive overview of the advancements in t-way interaction test generation strategies. The final step involved synthesizing the findings to provide a comprehensive overview of the advancements in t-way interaction test generation strategies. The strengths and weaknesses of each algorithm were discussed and their applicability to various testing scenarios was evaluated. The review also highlighted key trends, gaps and potential areas for future research. By following this systematic approach, the study aims to provide a thorough and unbiased evaluation of recent advancements in t-way interaction test generation strategies, offering valuable insights for researchers and practitioners in the field.

4. Results and Findings

This section presents the results and findings of the review, highlighting the advancements and performance metrics of various t-way test generation strategies developed between 2020 and 2024. Each strategy is analysed for its unique contributions, improvements and effectiveness in generating optimal test suites and detecting faults in software systems. The following subsections detail the individual strategies and their performance.

4.1 Dynamic Event Order Strategy (DEO) - 2020

The dynamic event order strategy (DEO) is rooted in event-driven testing principles based on a novel SCA strategy, which focus on the sequence and interactions of events within software applications (Kuhn et al., 2012). Unlike static testing methods, DEO dynamically adjusts the order of events during test case generation. This approach helps identify critical interactions that may not be evident when events occur in a fixed sequence. By analysing how different event orders affect software behaviour, DEO aims to uncover hidden faults and interaction issues that traditional methods might overlook. This Fault Detection Rate makes DEO particularly effective for applications where the sequence of user actions or system events can significantly impact performance and functionality. DEO employs a priority mechanism to reorder events based on their impact on software behaviour. Initially, events are prioritized according to predefined criteria, such as their likelihood to trigger faults or their importance in the application's workflow. As test cases are generated, DEO continuously re-evaluates and adjusts the event order to focus on the most critical interactions (Younis, 2020).

This dynamic reordering helps eliminate non-essential event sequences, thereby reducing the total number of test cases needed. By focusing on high-priority events and their interactions, DEO ensures that the most significant potential faults are tested thoroughly while minimizing redundant testing efforts. The effectiveness of DEO is measured through its ability to reduce the number of test cases while improving fault detection rates. Empirical studies have shown that DEO can significantly decrease the total number of test cases compared to traditional static methods. This reduction is achieved without compromising the thoroughness of the testing process, as DEO's dynamic event ordering effectively targets critical interactions. Additionally, DEO has demonstrated improved fault detection rates in event-driven applications, identifying interaction faults that might go unnoticed in static testing scenarios (Younis, 2020). These performance metrics highlight DEO's potential to enhance the efficiency and effectiveness of software testing, particularly for complex, event-driven systems.

4.2 Elitist Hybrid Migrating Birds Optimization Strategy for t-way Test Suite Generation (EMBO-GA) – 2021

The elitist hybrid migrating birds optimization (EMBO-GA) strategy combines the principles of migrating birds optimization (MBO) with enhancements of genetic algorithm (GA) elitism tailored for t-way test suite generation. The genetic algorithm (GA) is a well-established optimization technique that mimics the process of natural selection and evolution. GA works by creating a population of candidate solutions, called chromosomes, and then iteratively applying genetic operators such as crossover and mutation to these chromosomes to generate new, potentially better, solutions (Esfandyari & Rafe, 2018). The algorithm is designed to gradually improve the quality of the solutions over successive generations, converging towards an optimal or near-optimal solution (Mirjalili & Lewis, 2016). By incorporating the principles of GA, the elitist hybrid migrating birds optimization strategy can leverage the strengths of both GA and migrating birds optimization to enhance the efficiency and effectiveness of t-way test suite generation. MBO is inspired by the V-formation flight pattern of migrating birds, which optimizes the group's overall energy expenditure (Duman et al., 2012). By applying this natural optimization method, EMBO-GA leverages the collaborative and adaptive behaviours of birds to explore the solution space efficiently. The hybrid approach integrates additional mechanisms to improve the balance between exploration and exploitation, crucial for generating diverse and effective test cases that can cover all t-way interactions within software parameters.

In EMBO-GA, the initial population of solutions is generated using the standard MBO approach, which emulates the formation and movement of birds in flight. This initial step ensures a wide and diverse search across the solution space. To enhance the optimization process, EMBO-GA incorporates supplementary mechanisms such as adaptive weight adjustments and local search enhancements. These mechanisms allow the algorithm to dynamically balance exploration of new areas and exploitation of known good solutions. By combining these elements, EMBO-GA can effectively refine test suites to ensure comprehensive coverage of t-way interactions, improving overall test quality and efficiency. EMBO-GA has demonstrated significant improvements in both fault detection and test coverage. The strategy's ability to adaptively balance exploration and exploitation results in more effective identification of critical interactions within the software being tested. Empirical evaluations have shown that EMBO-GA can achieve higher fault detection rates compared to traditional MBO and other hybrid optimization techniques (Zakaria et al., 2021). Additionally, the strategy's efficient search process reduces the overall number of test cases required, thereby lowering computational costs and time. These performance metrics underscore EMBO-GA effectiveness in generating high-quality t-way test suites for complex software systems (Zakaria et al., 2021).

4.3 Tolerance Structurally Modified Whale Optimization Algorithm (TSWOA) - 2022

The tolerance structurally modified whale optimization algorithm (TSWOA) is an adaptation of the original whale optimization algorithm (WOA), designed to improve performance in test generation. WOA is inspired by the hunting behaviour of humpback whales, which use bubble-net feeding to trap schools of fish (Mirjalili & Lewis, 2016). TSWOA introduces structural modifications to enhance this natural optimization process, making it more suitable for software testing. Specifically, TSWOA incorporates tolerance thresholds to handle variability in software parameters, allowing it to manage high-dimensional and complex test scenarios more effectively. This theoretical foundation aims to create a more robust and adaptable optimization algorithm. TSWOA introduces several key modifications to the original WOA. One significant change is the implementation of tolerance thresholds, which allow the algorithm to handle variability in the software parameters being tested. These thresholds help TSWOA to focus on significant interactions while ignoring minor variations that may not impact software behaviour.

Additionally, TSWOA employs a modified search mechanism to explore the solution space more effectively. This mechanism includes adjustments to the mathematical models that guide the whales' movements, enhancing the algorithm's ability to converge on optimal solutions quickly. These modifications make TSWOA more adaptable and effective for complex test generation tasks. TSWOA excels in high-dimensional and complex test scenarios, showing significant improvements in convergence speed and fault detection rates. The algorithm's ability to handle variability through tolerance thresholds allows it to focus on critical interactions, improving its efficiency. Empirical studies have demonstrated that TSWOA can achieve faster convergence compared to traditional WOA and other optimization algorithms. Additionally, TSWOA's modified search mechanism enhances its fault detection capabilities, identifying more interaction faults in fewer test cases (Hassan et al., 2022). These performance metrics highlight TSWOA's potential to improve the effectiveness and efficiency of t-way interaction testing in complex software systems.

4.4 Seeding and Constraint Support in IPOG (SCIPOG) - 2023

The seeding and constraint support in in-parameter-order generalized (SCIPOG) strategy builds upon the traditional IPOG method by incorporating seeding and constraint support to improve test case generation efficiency. IPOG is a well-established technique for generating t-way interaction test cases, but it can be limited by its inability to handle complex constraints and its need for extensive computation (Lei et al., 2007). SCIPOG addresses these limitations by introducing a seeding mechanism, which uses known critical values to initialize test cases and by supporting complex constraints to ensure that generated test cases are valid. This theoretical foundation aims to enhance the efficiency and applicability of the IPOG method. SCIPOG introduces a seeding mechanism to initialize test cases with known critical values, which are identified based on prior knowledge or specific criteria. This approach helps to focus the test generation process on high-risk combinations, reducing the number of test cases needed.

Additionally, SCIPOG supports complex constraints, ensuring that generated test cases are valid and relevant to the software being tested. Constraints are applied during the test generation process to filter out invalid combinations, streamlining the generation of effective test cases. These implementation details make SCIPOG particularly suitable for applications with intricate parameter dependencies and constraints. SCIPOG has demonstrated significant reductions in test case generation time and improved focus on high-risk combinations. By incorporating seeding and constraint support, SCIPOG streamlines the test generation process, making it more efficient than traditional IPOG. Empirical studies have shown that SCIPOG can generate fewer test cases while maintaining or improving fault detection rates compared to standard IPOG. The ability to handle complex constraints ensures that the generated test cases are relevant and valid, further enhancing the overall effectiveness of the testing process (Muazu et al., 2023). These performance improvements make SCIPOG a valuable strategy for efficient and effective t-way interaction testing.

4.5 Improved Asexual Reproduction Optimization (ImpARO) - 2024

The improved asexual reproduction optimization (ImpARO) algorithm enhances the original asexual reproduction optimization (ARO) algorithm, focusing on improving solution diversity and fault detection. ARO is inspired by the asexual reproduction process, where offspring are generated without genetic recombination, leading to rapid population growth (Farasat et al., 2010). ImpARO introduces new mechanisms to enhance this process, making it more suitable for generating diverse and effective test cases. By improving solution diversity, ImpARO aims to cover a broader range of interactions and increase the likelihood of detecting faults in software systems. ImpARO introduces new mutation and selection mechanisms to enhance the diversity of solutions generated during the test case generation process. These mechanisms ensure that the generated test cases are varied and cover a wide range of potential interactions.

Additionally, ImpARO employs a self-adaptive approach to dynamically adjust algorithm parameters based on the current state of the population. This self-adaptive mechanism allows ImpARO to optimize its performance continuously, adapting to different testing scenarios and improving its efficiency. These implementation details make ImpARO a robust and flexible algorithm for t-way interaction test generation. ImpARO achieves high fault detection rates with fewer test cases, making it an efficient and effective strategy for t-way interaction testing. The enhanced mutation and selection mechanisms ensure that the generated test cases are diverse and comprehensive, improving the algorithm's ability to detect faults. Empirical studies have shown that ImpARO can achieve higher fault detection rates compared to traditional ARO and other optimization algorithms. Additionally, the self-adaptive approach allows ImpARO to perform well across different testing scenarios, further enhancing its applicability (Pira & Khodizadeh-Nahari, 2024). These performance metrics highlight ImpARO's potential to improve the efficiency and effectiveness of t-way interaction testing.

5. Analysis and Discussion

The analysis of t-way test generation strategies over the past five years reveals significant advancements. The Dynamic Event Order Strategy (DEO) in 2020 reduced test cases and improved fault detection in event-driven applications by dynamically adjusting event order. The Elitist Hybrid MBO-GA Strategy (EMBO-GA) in 2021 combined migrating birds optimization with genetic algorithms, enhancing fault detection and test coverage in complex systems through an elitist strategy. In 2022, the Tolerance Structurally Modified Whale Optimization Algorithm (TSWOA) improved convergence speed and fault detection rates in high-dimensional scenarios by adapting the whale optimization algorithm with structural modifications and tolerance thresholds. The Seeding and Constraint Support in In-Parameter-Order Generalized (SCIPOG) strategy in 2023 enhanced the IPOG method with seeding and constraint support, reducing test generation time and focusing on high-risk combinations for intricate software systems. Lastly, the Improved Asexual Reproduction Optimization (ImpARO) algorithm in 2024 introduced new mutation and selection mechanisms and a self-adaptive approach for dynamic parameter adjustment, achieving high fault detection rates with fewer test cases. Comparative analysis highlighted each algorithm's strengths and limitations: DEO for event-driven applications, EMBO-GA for complex systems, TSWOA for high-dimensional scenarios, SCIPOG for systems with intricate parameter dependencies and ImpARO for diverse software systems. Despite advancements addressing computational complexity and test coverage, gaps remain in handling dynamic software behaviors and reducing test generation time. Future research should focus on these areas, leveraging current algorithms' strengths and exploring new techniques to enhance t-way interaction testing's effectiveness and practicality.

6. Conclusion

This research review provides a comprehensive analysis of t-way test generation strategies from 2020 to 2024, highlighting methodologies, innovations and implications in software testing. It examines several meta-heuristic algorithms, including DEO, EMBO-GA, TSWOA, SCIPOG, ImpARO and their effectiveness in generating optimal test cases. Findings indicate significant improvements in test suite reduction and fault detection, with each strategy showing unique strengths in different testing challenges. However, ongoing challenges such as parameter tuning and complex interactions persist. Future research should focus on enhancing t-way test generation strategies, leveraging meta-heuristic advancements and exploring new techniques to improve computational efficiency, test coverage and fault detection. This review serves as a valuable resource, providing insights into the current state-of-the-art in t-way interaction test generation and guiding future research directions. The conclusion emphasizes recent advancements, such as improved fault detection rates and reduced test generation time and proposes future research to further enhance t-way interaction testing's effectiveness and practicality.

Acknowledgement: This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Author Contributions: The research study was carried out successfully with contributions from all authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

Chen, E. (2023). *Development of Hazard and Reliability Methodologies for Digital Instrumentation and Control Systems*. North Carolina State University. https://doi.org/10.1145/3652155.

Duman, E., Uysal, M., & Alkaya, A. F. (2012). Migrating birds optimization: a new metaheuristic approach and its performance on quadratic assignment problem. *Information Sciences*, *217*, 65-77. https://doi.org/10.1016/j.ins.2012.06.032.

Esfandyari, S. and Rafe V. (2018). A tuned version of genetic algorithm for efficient test suite generation in interactive t-way testing strategy Inform. *Software Tech*. *94*, 165-185. https://doi.org/10.1016/j.infsof.2017.10.007.

Farasat, A., Menhaj, M. B., Mansouri, T., & Moghadam, M. R. S. (2010). ARO: A new model-free optimization algorithm inspired from asexual reproduction. *Applied Soft Computing*, *10*(4), 1284-1292. https://doi.org/10.1016/j.asoc.2010.05.011.

Hassan, A. A., Abdullah, S., Zamli, K. Z., & Razali, R. (2022). Whale Optimization Algorithm Strategies for Higher Interaction Strength T-Way Testing. *Computers, Materials & Continua*, *73*(1). http://doi.org/10.32604/cmc.2022.026310.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, *51*(1), 7-15. https://doi.org/10.1016/j.infsof.2008.09.009.

Kuhn, D. R., Higdon, J. M., Lawrence, J. F., Kacker, R. N., & Lei, Y. (2012, April). Combinatorial methods for event sequence testing. In *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation* (pp. 601-609). IEEE. http://doi.org/10.1109/ICST.2012.1.

Kuhn, D. R., Kacker, R. N., & Lei, Y. (2017, March). A Model for T-Way Fault Profile Evolution during Testing. In *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 162-170). IEEE. http://doi.org/10.1109/ICSTW.2017.35.

Lei, Y., Kacker, R., Kuhn, D. R., Okun, V., & Lawrence, J. (2007, March). IPOG: A general strategy for t-way software testing. In *14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07)* (pp. 549-556). IEEE. http://doi.org/10.1109/ECBS.2007.47.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, *95*, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

Muazu, A. A., Hashim, A. S., Sarlan, A., & Abdullahi, M. (2023). 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*, *35*(1), 185-201. https://doi.org/10.1016/j.jksuci.2022.11.010.

Othman, R. R., & Zamli, K. Z. (2011). T-way strategies and its applications for combinatorial testing. *International Journal of New Computer Architectures and their Applications*, *1*(2), 459-73.

Pira, E., & Khodizadeh-Nahari, M. (2024). Combinatorial t-way test suite generation using an improved asexual reproduction optimization algorithm. *Applied Soft Computing*, *150*, 111070. https://doi.org/10.1016/j.asoc.2023.111070.

Younis, M. I. (2020). DEO: A dynamic event order strategy for T-way sequence covering array test data generation. *Baghdad Science Journal*, *17*(2), 0575-0575. *http://doi.org/10.21123/bsj.2020.17.2.0575*.

Zakaria, H. L., Zamli, K. Z., & Din, F. (2021, April). Hybrid migrating birds optimization strategy for t-way test suite generation. In *Journal of Physics: Conference Series* (Vol. 1830, No. 1, p. 012013). IOP Publishing. http://doi.org/10.1088/1742-6596/1830/1/012013.

Zamli, K. Z., Klaib, M. F., Younis, M. I., Isa, N. A. M., & Abdullah, R. (2011). Design and implementation of a t-way test data generation strategy with automated execution tool support. *Information Sciences*, *181*(9), 1741-1758.