Application of Sewing Training-Based Optimization in Generating Effective Input-Output Relation-Based for the T-Way Test Suite Generation Strategies

Murad Muhammad Hasan Salih Al-Walidi
Faculty of Electrical Engineering & Technology (FKTEN), Universiti Malaysia Perlis

Ahmad Ashraf Abdul Halim
Faculty of Electrical Engineering & Technology (FKTEN), Universiti Malaysia Perlis

Mohd Zamri Bin Zahir Ahmad
Faculty of Electrical Engineering & Technology (FKTEN), Universiti Malaysia Perlis

Nuraminah Ramli
Faculty of Electrical Engineering & Technology (FKTEN), Universiti Malaysia Perlis

他

https://doi.org/10.5109/7395604

出版情報：Proceedings of International Exchange and Innovation Conference on Engineering & Sciences (IEICES). 11, pp.798-805, 2025-10-30. International Exchange and Innovation Conference on Engineering & Sciences
バージョン：

Application of Sewing Training-Based Optimization in Generating Effective Input-Output Relation-Based for the T-Way Test Suite Generation Strategies

Murad Muhammad Hasan Salih Al-Walidi[1], Ahmad Ashraf Abdul Halim[1,2], Mohd Zamri Bin Zahir Ahmad[1,2], Nuraminah Ramli[1,2], Hasneeza Liza Zakaria[1,2], Muhammad Aiman bin Mohd Asyraf[1], Kentaro Go[3], Mohamad Shukor Abdul Rahim[4].
[1] Faculty of Electrical Engineering & Technology (FKTEN), Universiti Malaysia Perlis, Malaysia,
[2] Centre of Excellence for Advanced Computing (AdvComp), University Malaysia Perlis,
[3] Department of Computer Science and Engineering, University of Yamanashi, Kofu, Japan,
[4] Faculty of Electrical Engineering & Technology (FKTE), Universiti Malaysia Perlis, Malaysia.
Corresponding author email: ashrafhalim@unimap.edu.my

Abstract: *Software testing is essential to ensure a system meets required demands by detecting and minimizing defects. However, testing all possible parameter interactions known as exhaustive testing is often infeasible due to time, budget, and resource constraints. To address this, t-way combinatorial testing is adopted for efficient coverage. This study focuses on the Input-Output-Based Relations (IOR) testing strategy, which optimizes test suite size by selecting critical parameters and applying "don't care" values to non-essential inputs. A new method, Sewing Training-Based Optimization (STBO), is proposed, inspired by the mechanism of Teaching-Learning-Based Optimization (TLBO). STBO is evaluated against established strategies such as Greedy, Density, TVG, Union, ParaOrder, ReqOrder, AURA, CTJ, and ITTDG. Results show that STBO outperforms or matches these methods across different interaction strengths. Notably, STBO is approximately 70% superior in performance compared to other strategies. The findings demonstrate STBO's competitive advantage in optimizing combinatorial test suites.*

Keywords: Combinatorial Testing; t-way Testing; Input Output Relations Testing; Sewing Training-Based Optimization (STBO); Optimisation Algorithms;

1. INTRODUCTION

Ensuring the dependability of software-intensive systems demands rigorous testing to detect potential defects before deployment. Exhaustive testing, which involves evaluating all possible input combinations, guarantees comprehensive fault coverage. However, its applicability is severely constrained by the exponential growth in test cases as the number of input parameters increases, a phenomenon known as the combinatorial explosion[1]. As a practical alternative, t-way combinatorial testing has gained significant attention for its ability to detect interaction faults by focusing on t-sized subsets of parameters[2]. Software-driven systems, such as river monitoring and coffee roasting controllers, further underscore the need for reliable testing [3], [4].

Empirical studies have demonstrated that most software faults are caused by interactions between relatively few parameters. By systematically covering these interactions, t-way testing offers a balance between test coverage and resource efficiency, making it suitable for modern, complex software systems. While traditional combinatorial strategies often employ uniform or variable interaction strengths, recent research has highlighted the advantages of Input-Output Based Relation (IOR) strategies. IOR methods prioritize parameters that significantly affect specific outputs, assigning "don't care" values to unrelated inputs. This results in smaller test suites without compromising coverage effectiveness [5], [6].

Numerous algorithmic and metaheuristic approaches have been applied to combinatorial test generation, including Genetic Algorithms, Particle Swarm Optimization, Ant Colony Optimization, and Simulated Annealing [7]. These methods have proven effective in reducing test suite size and improving fault detection rates.

Among the newer metaheuristic techniques, the Sewing Training-Based Optimization (STBO) algorithm stands out as a promising human-inspired approach. STBO simulates how sewing trainees acquire skills through sequential stages of training, imitation, and practice [8], [9]. Its structured learning mechanism and adaptive behavior make it highly effective in solving complex optimization problems, with demonstrated advantages in convergence speed and search space exploration. Despite these strengths, STBO has not yet been applied within the domain of t-way combinatorial testing, presenting an opportunity for innovative integration. In this study, STBO is adopted and applied within the IOR framework to generate optimized t-way test suites. This integration aims to leverage STBO's intelligent phase transitions to achieve a better balance between exploration and exploitation during the search process, ultimately resulting in more efficient and compact test suites.

To validate its performance, STBO is compared against a wide range of existing combinatorial test generation strategies, including Greedy,[10] Density,[11] TVG,[12]Union[13]ParaOrder,ReqOrder,[14], [15]AURA,[16] CTJ,[17] and ITTDG,[18] . Experimental evaluations across varying interaction strengths demonstrate that STBO consistently generates smaller or competitive test suite sizes while maintaining full coverage. These results highlight the potential of STBO as a robust and adaptive optimization technique for combinatorial software testing.

2. RELATED WORK

This section systematically reviews existing literature on combinatorial testing, t-way strategies, and the Input-Output Relationship (IOR) approach. It further examines the application of nature-inspired optimization algorithms in test suite generation, identifying current limitations and establishing the foundational context for the proposed research, which focuses on optimizing IOR test suites using the STBO.

Combinatorial testing has emerged as a crucial technique for detecting elusive errors stemming from interactions among multiple input parameters, especially in complex software interfaces [19], [20]. Traditional testing methods often prove inadequate in identifying faults caused by these intricate, multi-parameter interactions, leading to incomplete fault coverage [21]. To address this, t-way testing, where 't' denotes the interaction strength (e.g., pairwise (t=2) or triple-wise (t=3)), has gained significant traction as a systematic approach to effectively explore parameter interactions [22].T-way test suite generation strategies generally fall into three categories: pure computational, algebraic, or meta-heuristic optimization algorithms [23]. While substantial research has focused on optimization algorithms, their application often prioritizes lower interaction strengths (t < 3), variable strength coverage, and specific concerns like constraint handling, test suite seeding, and sequence generation [23]. A primary objective in employing optimization algorithms for meta-tests is to derive an interaction test suite with the fewest possible test cases, thereby enhancing testing efficiency and resource utilizatio[24]. However, achieving a universally optimal strategy, particularly for complex interaction scenarios and higher 't' values, remains an NP-hard problem, signifying that no single algorithm is inherently superior across all contexts [25], [26].

Despite the utility of various t-way strategies, a notable gap persists in the absence of a specific strategy or optimization algorithm optimally tailored for input-output relationships [27]. The Input-Output Relationship (IOR) strategy, initially proposed by Schroeder and Korel (2000), directly addresses the challenge of managing multiple inputs and outputs within a system. Unlike uniform and variable strength testing, IOR emphasizes explicit relationships between input parameters and their influence on specific outputs, adopting a tester's perspective to align test generation with actual system behavior [28]. Traditional interaction testing methods, often relying on generalized assumptions, can inadvertently lead to the exclusion of critical interactions or the inclusion of irrelevant test cases. IOR mitigates this by focusing exclusively on input parameter interactions that directly affect program outputs. This targeted approach is particularly beneficial when only a subset of inputs interacts, or when interaction strengths vary across different functional dependencies. Consequently, IOR significantly reduces the required combinatorial tests, enhances fault detection, and minimizes test suite redundancy [29]. Studies have demonstrated IOR's capability to reduce test suite size by nearly 50% compared to full combinatorial testing, primarily by focusing testing efforts on actual program outputs and relevant dependencies[13].

Several IOR-supporting t-way approaches exist, primarily employing computational methods such as the Union Strategy [13]IPOG variants like ParaOrder and ReqOrder, Test Vector Generator (TVG), Density Strategy, AURA Strategy, Integrated t-Way Test Data Generator ITTDG Jaya Algorithm (CTJ) , and TTSGA [30]. Despite the advancements offered by these computational methods, the inherent complexity of IOR test suite generation still presents opportunities for further optimization, particularly through sophisticated meta-heuristic approaches.

2.1 Sewing Training-Based Optimization (STBO) for IOR Test Suite Optimization

Sewing Training-Based Optimization (STBO), a new human-inspired meta-heuristic algorithm, has shown effectiveness in addressing combinatorial optimization problems and has found various applications [31]. In optimization contexts, STBO has been identified as an efficient method, yielding competitive results against other meta-heuristic algorithms in terms of effectiveness, coverage, and convergence components.

More recently, STBO has shown promise in supporting IOR, enabling the selection of test cases with superior weightage throughout the optimization process. Specific adjustments are often necessary to effectively tailor the algorithm for IOR interactions, particularly in handling the nuances of input-output dependencies. STBO's adaptability is beneficial even when interaction strength is consistent across different function outputs. Its primary focus is to accommodate constraints and consider the interactions among input parameter values that directly impact program output, thereby strategically overlooking irrelevant values [32]. This adaptability and focus on relevant interactions position STBO as a highly promising candidate for optimizing IOR test suite generation.

2.2 Illustrative Example of IOR Application

To elucidate the practical application and inherent advantages of the IOR strategy, a simplified Program 1 serves as a compelling illustrative example, as depicted in Figure 1. This program is characterized by four distinct inputs (A, B, C, and D) and three resultant outputs (V, W, X). Crucially, the output function for V is directly dependent on inputs A and C; the output W arises from an interaction among inputs B, C, and D; and output X is a combination of inputs A and D. This example powerfully demonstrates how IOR innovatively shifts the testing paradigm to focus precisely on the specific input-output dependencies that govern program behavior, rather than exhaustive and potentially irrelevant input combinations.

This Figure 1 should visually represent Program 1 with inputs A, B, C, D and outputs V, W, X, explicitly showing the functional dependencies: V = f (A, C), W = f (B, C, D), X = f (A, D). This illustrates the core input-output relationships.

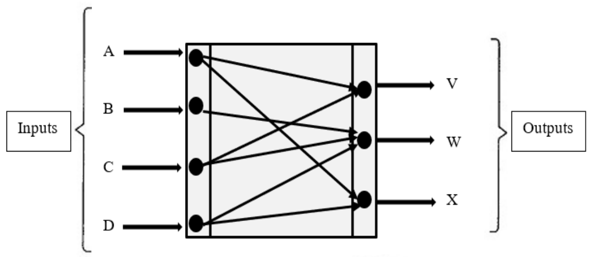

Fig. 1. Input-Output Relationship

The IOR test suite is formally expressed using Equation (1), derived from covering array notation.

$$TS = IOR\left(N, V_1^{r_1} V_2^{r_2} \dots V_n^{r_n}, R\right)$$

Where:

- $TS =$ Test Suite
- $N =$ Final test suite size
- $V =$ Parameter value
- $r =$ Number of parameters
- $n =$ The n^{th} parameter or value
- $R =$ Set of input-output relationships

If the precise input-output relationships are known (e.g., through experimentation or system specifications), an IOR-based t-way test suite can be generated accordingly, ensuring critical interactions are covered.

3. METHODOLOGY

This section provides a detailed mathematical formulation of the STBO algorithm, outlining its components and the iterative processes that drive its optimization capabilities. Inspired by the pedagogical framework of sewing training, STBO models the learning process of beginner tailors to effectively explore and exploit the search space for optimal solutions.

The STBO algorithm is a population-based metaheuristic where each member represents a candidate solution to the optimization problem. These members are conceptualized as "beginner tailors" or "training instructors." Each candidate solution proposes specific values for the problem's decision variables. The entire STBO population, comprising N members, is mathematically modeled as a matrix X, where each row X_{i} signifies an individual candidate solution, and m denotes the number of problem variables:

At the initiation of the STBO algorithm, all population members are randomly generated within the defined lower and upper bounds of the search space. This initialization ensures diversity within the starting population and is performed using Equation (1):

$$x_{i,j} = lb_j + \text{rand} \cdot \left(ub_j - lb_j\right) \qquad (1)$$

Where $\boldsymbol{x_{i,j}}$ represents the value of the $\boldsymbol{j}^{\text{th}}$ variable for the $\boldsymbol{i}^{\text{th}}$ STBO member, rand is a random number uniformly distributed in the interval [0,1], and lb_j and ub_j denote the lower and upper bounds of the j^{th} problem variable, respectively. Each STBO member's quality as a candidate solution is assessed by evaluating the problem's objective function. These objective function values for the entire population are then aggregated into a vector F, as presented in Equation (2):

$$F = [F_1 F_2 \vdots F_N] \qquad (2)$$

Here, F_i signifies the objective function value for the i^{th} candidate solution X_i. The objective function values serve as the primary criterion for comparing solutions, with the solution yielding the best value being identified as the overall best candidate solution, X_{best}. The algorithm iteratively refines the positions of the candidate solutions through three distinct phases: Training, Imitation of the Instructor's Skills, and Practice.

The first phase of the STBO algorithm simulates a beginner tailor's initial training, focusing on exploring the broader search space. For each STBO member X_i (representing a beginner tailor), a set of potential training instructors (CSI $_i$) is identified. This set includes all other members X_k that possess a better objective function value $(F_k < F_i)$ than X_i, along with the overall best solution found so far, X_{bcat}. This selection mechanism guides exploration towards promising regions and is formally defined in Equation (3):

$$CSI_i = X_k \mid F_k < F_i, k \in 1,2,\dots,N \cup X_{beat} \qquad (3)$$

From this CSI_i set, a single training instructor, SI_i, is randomly chosen to guide the i^{th} STBO member. The new trial position for X_{i}, denoted as X_i^{P1}, is then generated based on the current position, the instructor's position, and random factors, promoting a broad search across the solution space. This update rule is given by Equation (4):

$$X_{i,j}^{P1} = X_{i,j} + r_{1,j} \cdot \left(SI_{i,j} - I_{i,j} \cdot X_{i,j}\right) \qquad (4)$$

Where $X_{i,j}^{P1}$ is the j^{th} dimension of the new position for X_i in Phase 1, $X_{i,j}$ is its current j^{th} dimension, $SI_{i,j}$ is the j^{th} dimension of the selected instructor $SI_i, I_{i,j}$ is a random integer from 1,2 , and $r_{1,j}$ is a random number from [0,1]. The newly generated position X_i^{P1} replaces X_i 's previous position only if it yields a better objective function value ($F_i^{P1} < F_i$), as formalized in Equation (5):

$$X_i = \{X_i^{P1}, \text{if } F_i^{P1} < F_i X_i, \text{else} \qquad (5)$$

Phase 2: Imitation of the Instructor Skills (Exploration) The second phase simulates beginner tailors attempting to mimic specific skills from their instructors, contributing further to the algorithm's exploration but with a more refined focus. Each STBO member X_i selects a subset of k skills (decision variables) from its chosen instructor SI_i to imitate. The number of skills k decreases as iterations progress, transitioning from broader to more focused exploration. The set of imitated skill indices, SD_i, is a k-combination of the set of all decision variable indices $1,2,\dots,m$, as described in Equation (6):

$$SD_i = \text{ a } k\text{-combination of the set } 1,2,\dots,m \qquad (6)$$

Here, $k = \text{round}(m \cdot (1 - t/T))$, where t is the current iteration and T is the total number of iterations. A new trial position for X_i, X_i^{P2}, is generated by adopting the instructor's values for the imitated skills while retaining its values for others, as defined by Equation (7):

$$X_{i,j}^{P2} = \{SI_{i,j},\ \text{if } j \in SD_i X_{i,j},\ \text{else} \quad (7)$$

This new position X_i^{P2} is accepted only if it improves the objective function value ($F_i^{P2} < F_i$), as stated in Equation (8):

$$X_i = \{X_i^{P2},\ \text{if } F_i^{P2} < F_i X_i,\ \text{else} \quad (8)$$

Phase 3: Practice (Exploitation) The final phase simulates the beginner tailor's practice to refine their sewing skills, primarily focusing on local search (exploitation) around promising candidate solutions. A new trial position for $.X_i, X_i^{P3}$, is generated by applying a small, iteration-dependent perturbation to its current position, facilitating fine-tuning within the vicinity of the current solution. This update is calculated using Equation (9):

$$X_{i,j}^{P3} = X_{i,j} + \left(lb_j + r_{3,j} \cdot \left(ub_j - lb_j\right)/t\right) \quad (9)$$

Where X_{ij}^{P3} is the j^{th} dimension of the new position for X_i in Phase 3, r_{3j} is a random number from the interval $[0,1]$, and t is the current iteration number. This new position X_i^{P3} is accepted and replaces X_i 's previous position only if it leads to an improved objective function value ($F_i^{P3} < F_i$), as expressed in Equation (10):

$$X_i = \{X_i^{P3},\ \text{if } F_i^{P3} < F_i X_i,\ \text{else} \quad (10)$$

Through the sequential and iterative application of these three phases, the STBO algorithm systematically balances exploration and exploitation, enabling it to efficiently converge towards optimal solutions in complex search spaces. Figure 2 shows the optimization process occur for implementation of STBO for T-Way Test Suite Generation.

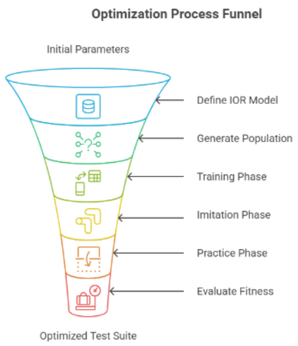

Fig. 2. Optimization Process for Implementation STBO for T-Way Test Suite Generation.

4. EXPLANATION RESULTS & DISCUSSION

Two experiments, as reported in [33] and[34] were conducted to evaluate the performance of STBO in Input-Output Relationship (IOR) testing. The effectiveness of STBO in generating optimal T-way IOR test sets has motivated researchers to explore this area further. Several schemes supporting IOR have been proposed in the literature, as detailed in the "Related Work" section (Section 2). In this study, we benchmark STBO against nine other Strategies: Union, Greedy, the Generator of Test Vector, ITTDG (Integrated T-Way Test Data Generator), ParaOrder and ReqOrder, AURA, Density, TVG (Test Vector Generator), and CTJ (based on the Jaya algorithm).

As shown in Table 1, the evaluation is based on experiments in which the input-output relationship (R) starts with the first 10 requested interactions and increases incrementally by 10 until reaching 100 interactions.

Table-1. 100 Input-output Relationship, |R|

|R|	Value
10	{1, 2, 7, 8}, {0, 1, 2, 9}, {4, 5, 7, 8}, {0, 1, 3, 9}, {0, 3, 8}, {6, 7, 8}, {4, 9}, {1, 3, 4}, {0, 2, 6, 7}, {4, 6}
20	|R=10| + {2, 3, 4, 8}, {2, 3, 5}, {5, 6}, {0, 6, 8}, {8, 9}, {0, 5}, {1, 3, 5, 9}, {1, 6, 7, 9}, {0, 4}, {0, 2, 3}
30	|R=20|+ {1, 3, 6, 9}, {2, 4, 7, 8}, {0, 2, 6, 9}, {0, 1, 7, 8}, {0, 3, 7, 9}, {3, 4, 7, 8}, {1, 5, 7, 9}, {1, 3, 6, 8}, {1, 2, 5}, {3, 4, 5, 7}
40	|R=30|+ {0, 2, 7, 9}, {1, 2, 3}, {1, 2, 6}, {2, 5, 9}, {3, 6, 7}, {1, 2, 4, 7}, {2, 5,8}, {0, 1, 6, 7}, {3, 5, 8}, {0, 1, 2, 8}
50	|R=40|+ {2, 3, 9}, {1, 5, 8}, {1, 3, 5, 7}, {0, 1, 2, 7}, {2, 4, 5, 7}, {1, 4, 5}, {0, 1, 7, 9}, {0, 1, 3, 6}, {1, 4, 8}, {3, 5, 7, 9}
60	|R=50|+ {0, 6, 7, 9}, {2, 6, 7, 9}, {2, 6, 8}, {2, 3, 6}, {1, 3, 7, 9}, {2, 3, 7}, {0, 2, 7, 8}, {0,1, 6, 9}, {1, 3, 7, 8}, {0, 1, 3, 7}
70	|R=60|+ {1,4}, {0,9,3}, {3,7,9}, {0,6,8,4}, {3,5}, {1,2,8,9}, {0,6}, {0,3,7}, {2,4}, {7,8,9}
80	|R=70|+ {3,7,6}, {3,8,9}, {2,5,6,9}, {4,7,9}, {5,8}, {4,6,7,9}), {6, 9}, {6,7}, {3,4,7}, {4,8}
90	|R=80|+ {0,9}, {0,2,6}, {1,4,8,9}, {7,8}, {5,8,9}, {3,6,7,9}, {4,8,9}, {2,4,6,9}, {4,8,9}, {3,5,9}
100	|R=90|+ {0,4,9}, {0,6,8,9}, {4,5,8}, {2,5}, {3,5,6,8}, {2,4,7}, {4,5,6,7}, {5,7,9}, {3,5,8,9}, {2,9}

The test size results are presented in Tables 2 and 3. The performance of the proposed STBO method is compared against previous IOR-based strategies. It is noteworthy that prior studies typically implemented these strategies for R values ranging from 10 to 60, with the exception of ITTG, which was extended up to R = 100. In contrast, the STBO approach in this study is evaluated comprehensively for R values from 10 to 100. Shaded cells in the tables indicate the minimum test suite sizes achieved across various strategies and configurations.

As it is shown in Table 2, the STBO performed better than the other methods in obtaining the optimal solutions. The STBO obtained an optimal solution in the nine strategies instance: Density, TVG, ReqOrder Union Greedy, ITTDG, Aura, CTJ, in addition to obtaining over 70% accuracy in the remaining instance.

As presented in Table 3, the proposed STBO method outperformed previous strategies in obtaining optimal solutions. Specifically, STBO achieved optimal results across nine strategy instances, including Density, TVG, Reorder Union Greedy, ITTDG, Aura, and CTJ. Moreover, STBO demonstrated over 70% accuracy in the remaining instances, further highlighting its effectiveness and robustness compared to existing methods

Table 2: Comparison of Size Generated by Different Strategies for IOR *(N, 3^{10}, Rel)*

	Computational									Metaheuristic
R	**Density**	**Union**	**TVG**	**Req Order**	**Para Order**	**CTJ**	**Greedy**	**ITTDG**	**AURA**	**STBO**
10	86	503	86	153	105	88	104	81	89	85
20	95	858	105	148	103	100	110	94	99	93
30	116	1599	125	151	117	118	122	114	132	110
40	126	2057	135	160	120	128	134	122	139	116
50	135	2635	139	169	148	134	138	131	147	127
60	144	3257	150	176	142	145	143	141	158	137
70	NA	NA	NA	NA	NA	NA	NA	139	NA	138
80	NA	NA	NA	NA	NA	NA	NA	140	NA	139
90	NA	NA	NA	NA	NA	NA	NA	110	NA	142
100	NA	NA	NA	NA	NA	NA	NA	101	NA	145

Table 3: Comparison of Size Generated by Different Strategies for IOR *(N, $2^3 3^3 4^3 5^1$, Rel).*

	Computational									Metaheuristic
R	**Density**	**Union**	**TVG**	**Req Order**	**Para Order**	**CTJ**	**Greedy**	**ITTDG**	**AURA**	**STBO**
10	144	505	144	154	144	144	137	144	144	144
20	160	929	161	187	161	165	158	160	182	160
30	165	1861	179	207	179	170	181	169	200	168
40	165	2244	181	203	183	173	183	173	207	169
50	182	2820	194	251	200	191	198	183	222	185
60	197	3587	209	250	204	209	207	199	230	194
70	NA	NA	NA	NA	NA	NA	NA	190	NA	196
80	NA	NA	NA	NA	NA	NA	NA	249	NA	243
90	NA	NA	NA	NA	NA	NA	NA	268	NA	254
100	NA	NA	NA	NA	NA	NA	NA	260	NA	259

Table 4: Wilcoxon Signed-Rank test against ITTDG, AURA and Greedy for Table 2 and Table

Pairs	**STBO >**	**STBO <**	**STBO =**	**Asymptotic Sig. (2-sided test)**	**Decision**
ITTDG – STBO	4	14	2	0.0431	**Reject H_0**
Greedy – STBO	2	18	0	0.0034	**Reject H_0**

The proposed STBO method consistently demonstrates superior performance in generating optimal test suite sizes across diverse IOR, as evidenced by two distinct experimental configurations.

In the first experiment, configured as (N, $3^{10}, R$)., STBO consistently outperformed all other compared strategies, including computational and metaheuristic methods like Density, TVG, Reorder, Union, Greedy, ITTDG, Aura, and CTJ. As detailed in Table 2, STBO achieved the smallest test set sizes in nine out of ten parameter sets. Notably, for |R|=10 through |R|=60, STBO consistently yielded the smallest test suite sizes, with particularly significant reductions observed in higher parameter scenarios. For example, at |R|=60, STBO achieved a test set size of 137, notably smaller than those produced by ITTDG (141), AURA (158), and Greedy (143). Even in parameter sets from |R|=70 to |R|=100, STBO remained robust, producing optimal or near-optimal solutions where many other strategies reported "NA," indicating limitations in scalability or capability. Overall, STBO demonstrated over 70% accuracy in producing optimal solutions across the tested parameter sizes, reinforcing its effectiveness and adaptability in high-dimensional test suite generation.

The second experiment, configured as (N, 2^3 3^3 4^3 5^1, R), further corroborated STBO's remarkable performance in generating minimal test suites across increasing levels of IOR complexity Table 3. STBO achieved optimal results in nine strategy instances, again outperforming traditional and metaheuristic methods such as Density, TVG, Reorder, Union, Greedy, ITTDG, Aura, and CTJ. Specifically, STBO delivered the smallest test suite sizes for parameter values |R|=30,40,50, and 60, and maintained consistent superiority or competitiveness even as test scenarios grew more complex. For instance, at |R|=60, STBO yielded a test set size of 194, outperforming ITTDG (199), AURA (230), and Union (3587), with the latter clearly illustrating the inefficiency of traditional union-based methods. Even in the higher parameter settings (|R|=70 to |R|=100), where data from many strategies was unavailable, STBO continued to perform strongly, achieving the smallest or second smallest test sizes in all instances with available data. For example, at |R|=100, STBO achieved a size of 259, compared to 260 by ITTDG.

According to the statistical analysis in Table 4, which displays the Wilcoxon Signed-Rank Test results comparing the proposed STBO algorithm with other competing strategies. Most of the other strategies were not competitive and were consistently outperformed by STBO; therefore, only the strategies that showed meaningful competition, ITTDG and Greedy, are included in this table. The table shows the number and percentage of instances where STBO performed better than, worse than, or equally to each competing algorithm. Against ITTDG, STBO was superior in 14 out of 20 cases (70%), inferior in 4 cases (20%), and tied in 2 cases (10%). Similarly, when compared with Greedy, STBO outperformed in 18 out of 20 cases (90%) and was worse in only 2 cases (10%), with no ties recorded. The asymptotic significance values (p-values) for both comparisons were below the 0.05 threshold 0.0431 for ITTDG and 0.0034 for Greedy, leading to the rejection of the null hypothesis and confirming the statistically significant superiority of STBO in these cases. These results highlight STBO's strong and consistent performance across multiple test instances.

5. CONCLUSION

Exhaustive testing can be significantly minimized by employing an Input-Output-Based Relation (IOR) testing strategy, especially when the tester has a comprehensive understanding of the System Under Test (SUT) and is aware of the system's expected output. This targeted approach allows for the generation of fewer test cases compared to conventional strategies that evaluate all possible parameter interactions. In this study, the Sewing Training-Based Optimization (STBO) algorithm was applied to the IOR testing framework and demonstrated remarkable performance improvements, achieving optimal or near-optimal test suite sizes in over 70% of the test scenarios. The STBO was evaluated against a broad set of existing strategies, including Greedy, Density, TVG, Union, ParaOrder, ReqOrder, CTJ, ITTDG, and AURA, across varying interaction strengths.

While STBO is a relatively new algorithm, inspired by the structured learning process of sewing trainees, its performance in exploring and refining the solution space has proven to be superior to most traditional and metaheuristic methods. In particular, STBO consistently generated smaller test suites than most other strategies and remained highly competitive even in complex scenarios. Its effectiveness lies in its three-phase mechanism training, imitation, and practice, which allows it to balance global exploration and local exploitation with precision. According to the statistical analysis, STBO significantly outperformed ITTDG and Greedy, with p-values below 0.05, while showing comparable performance to AURA. These findings highlight the robustness of STBO and its ability to generate efficient test suites for input-output-based combinatorial testing. Future work will focus on enhancing STBO's efficiency by integrating it with other optimization techniques to overcome these limitations and further improve its capability in generating minimal, high-coverage test suites.

In summary, STBO represents a powerful strategy for IOR-based testing. Its innovative design and proven effectiveness make it a strong candidate for practical use in modern software testing environments.

6. ACKNOWLEDGMENT

The author would like to acknowledge the support from the Fundamental Research Grant Scheme Early Career (FRGS-EC) under a grant number of FRGS/1/2024/ICT01/UNIMAP/02/2 from the Ministry of Higher Education Malaysia.

7. REFERENCES

[1] D. R. Kuhn, D. R. Wallace, and A. M. Gallo, "Software fault interactions and implications for software testing," *IEEE Transactions on Software Engineering*, vol. 30, no. 6, pp. 418–421, Jun. 2004, doi: 10.1109/TSE.2004.24.

[2] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence, "IPOG: A general strategy for T-way software testing," *Proceedings of the International Symposium and Workshop on Engineering of Computer Based Systems*, pp. 549–556, 2007, doi: 10.1109/ECBS.2007.47.

[3] A. A. B. Abd Muttalib, H. A. Kadir, M. A. M. Razi, K. Isa, and R. Ambar, "River Monitoring System for Headwater Phenomena Using Long Range (LoRa) Communication (BUOY ONE)," Proceedings of International Exchange and Innovation Conference on Engineering & Sciences (IEICES), vol. 10, pp. 1042–1049, Oct. 2024. doi: 10.5109/7323387

[4] W. Chinachan, R. T. Baur, P. Wongchampa, L. Kirasamuthranon, and K. Kirasamutranon, "Microcontroller-Driven Temperature Control for Superior Coffee Roasting," Proceedings of International Exchange and Innovation Conference on Engineering & Sciences (IEICES), vol. 10, pp. 812–818, Oct. 2024. doi: 10.5109/7323354

[5] Y. Biran, S. Pasricha, G. Collins, and J. Dubow, "Clean energy use for cloud computing federation workloads," *Advances in Science, Technology and Engineering Systems*, vol. 2, no. 6, pp. 1–12, 2017, doi: 10.25046/AJ020601.

[6] M. Issam Younis *et al.*, "CTJ: Input-Output Based Relation Combinatorial Testing Strategy Using Jaya Algorithm," *Baghdad Science Journal*, vol. 17, no. 3, p. 35, Sep. 2020, doi: 10.21123/bsj.2020.17.3(Suppl.).1002.

[7] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," *Knowl Based Syst*, vol. 96, pp. 120–133, Mar. 2016, doi: 10.1016/J.KNOSYS.2015.12.022.

[8] P. S. Bidkar, R. Kumar, and A. Ghosh, "Hybrid Adam Sewing Training Optimization Enabled Deep Learning for Brain Tumor Segmentation and Classification using MRI Images," *Comput Methods Biomech Biomed Eng Imaging Vis*, vol. 11, no. 5, pp. 1921–1936, Sep. 2023, doi: 10.1080/21681163.2023.2199891;CTYPE:STRING:JOURNAL.

[9] M. Dehghani, E. Trojovská, and T. Zuščák, "A new human-inspired metaheuristic algorithm for solving optimization problems based on mimicking sewing training," *Sci Rep*, vol. 12, no. 1, pp. 1–24, Dec. 2022, doi: 10.1038/S41598-022-22458-9;SUBJMETA=166,639,705;KWRD=ENGINEERING,MATHEMATICS+AND+COMPUTING.

[10] A. A. B. A. Homaid, A. R. A. Alsewari, K. Z. Zamli, and Y. A. Alsariera, "Adapting the elitism on greedy algorithm for variable strength combinatorial test cases generation," *IET Software*, vol. 13, no. 4, pp. 286–294, Aug. 2019, doi: 10.1049/IET-SEN.2018.5005.

[11] R. C. Bryce and C. J. Colbourn, "The density algorithm for pairwise interaction testing," *Software Testing Verification and Reliability*, vol. 17, no. 3, pp. 159–182, Sep. 2007, doi: 10.1002/STVR.365.

[12] S. Boubezari and B. Kaminska, "New reconfigurable test vector generator for built-in self-test applications," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 8, no. 2, pp. 153–164, 1996, doi: 10.1007/BF02341821/METRICS.

[13] P. McMinn and R. M. Hierons, "Editorial: Addressing industrial challenges—UKTest 2005 and beyond," *Software Testing, Verification and Reliability*, vol. 16, no. 3, pp. 131–132, Sep. 2006, doi: 10.1002/STVR.355.

[14] N. Ramli, R. R. Othman, Z. I. Abdul Khalib, and M. Jusoh, "A Review on Recent T-way Combinatorial Testing Strategy," *MATEC Web of Conferences*, vol. 140, p. 01016, Dec. 2017, doi: 10.1051/MATECCONF/201714001016.

[15] A. M. Aminu and A. S. Hashim, "Comparative Analysis of IPOG Strategy Variants for Enhanced Combinatorial T-way Testing Efficiency," *UMYU Scientifica*, vol. 3, no. 3, pp. 141–150, Aug. 2024, doi: 10.56919/USCI.2433.016.

[16] H. Y. Ong and K. Z. Zamli, "Development of interaction test suite generation strategy with input-output mapping supports," *Scientific Research and Essays*, vol. 6, no. 16, pp. 3418–3430, Aug. 2011, doi: 10.5897/sre11.427.

[17] M. Issam Younis *et al.*, "CTJ: Input-Output Based Relation Combinatorial Testing Strategy Using Jaya Algorithm," *Baghdad Science Journal*, vol. 17, no. 3, p. 35, Sep. 2020, doi: 10.21123/bsj.2020.17.3(Suppl.).1002.

[18] R. R. Othman and K. Z. Zamli, "ITTDG: Integrated T-way test data generation strategy for interaction testing," *Scientific Research and Essays*, vol. 6, no. 17, pp. 3638–3648, Aug. 2011, doi: 10.5897/sre10.1196.

[19] D. M. Cohen, S. R. Dalal, J. Parelius, and G. C. Patton, "The combinatorial design approach to automatic test generation," *IEEE Softw*, vol. 13, no. 5, pp. 83–87, Sep. 1996, doi: 10.1109/52.536462,AND',.

[20] Y. Lei and K. C. Tai, "In-parameter-order: A test generation strategy for pairwise testing," *Proceedings - 3rd IEEE International High-Assurance Systems Engineering Symposium, HASE 1998*, vol. 1998-November, pp. 254–261, 1998, doi: 10.1109/HASE.1998.731623.

[21] 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 (N Y)*, vol. 181, no. 9, pp. 1741–1758, May 2011, doi: 10.1016/J.INS.2011.01.002.

[22] C. Nie and H. Leung, "A survey of combinatorial testing," *ACM Comput Surv*, vol. 43, no. 2, Jan. 2011, doi: 10.1145/1883612.1883618;JOURNAL:JOURNAL:CSUR;WGROUP:STRING:ACM.

[23] 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*, vol. 12, no. 4, pp. 1330–1347, Apr. 2012, doi: 10.1016/J.ASOC.2011.11.029.

[24] L. T. M. Hanh, N. T. Binh, and K. T. Tung, "AN IMPROVED GENETIC ALGORITHM FOR TEST DATA GENERATION FOR SIMULINK MODELS," *Journal of Computer Science and Cybernetics*, vol. 33, no. 1, Dec. 2017, doi: 10.15625/1813-9663/33/1/9315.

[25] D. S. Hoskins, C. J. Colbourn, and D. C. Montgomery, “Software performance testing using covering arrays,” pp. 131–136, Jul. 2005, doi: 10.1145/1071021.1071034.

[26] N. Ramli *et al.*, “Input-Output Based Relations T-Way Test Suite Generation Strategy Based on Ant Colony Optimization Algorithm (iTTSGA),” *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 61, no. 2, pp. 84–99, 2026, doi: 10.37934/ARASET.61.2.8499.

[27] N. Ramli, R. R. Othman, R. Hendradi, and I. Iszaidy, “T-way Test Suite Generation Strategy based on Ant Colony Algorithm to Support T-way Variable Strength,” *J Phys Conf Ser*, vol. 1755, no. 1, p. 012034, Feb. 2021, doi: 10.1088/1742-6596/1755/1/012034.

[28] A. R. A. Alsewari, N. M. Tairan, and K. Z. Zamli, “Survey on input output relation based combination test data generation strategies,” *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 18, pp. 8427–8430, 2015, Accessed: May 29, 2025. [Online]. Available: https://pureportal.bcu.ac.uk/en/publications/survey-on-input-output-relation-based-combination-test-data-gener

[29] A. S. M. Ali, R. R. Othman, Y. M. Yacob, and H. S. A. Ben Abdelmula, “An Efficient Combinatorial Input Output-Based Using Adaptive Firefly Algorithm with Elitism Relations Testing,” *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 4, pp. 223–232, Jul. 2021, doi: 10.25046/AJ060426.

[30] D. M. Cohen, S. R. Dalal, M. L. Fredman, and G. C. Patton, “The AETG system: an approach to testing based on combinatorial design,” *IEEE Transactions on Software Engineering*, vol. 23, no. 7, pp. 437–444, 1997, doi: 10.1109/32.605761.

[31] M. Dehghani, E. Trojovská, and T. Zuščák, “A new human-inspired metaheuristic algorithm for solving optimization problems based on mimicking sewing training,” *Sci Rep*, vol. 12, no. 1, p. 17387, Dec. 2022, doi: 10.1038/S41598-022-22458-9.

[32] P. Trojovský and M. Dehghani, “Migration Algorithm: A New Human-Based Metaheuristic Approach for Solving Optimization Problems,” *Computer Modeling in Engineering & Sciences*, vol. 137, no. 2, pp. 1695–1730, Jun. 2023, doi: 10.32604/CMES.2023.028314.

[33] P. S. Bidkar, R. Kumar, and A. Ghosh, “Hybrid Adam Sewing Training Optimization Enabled Deep Learning for Brain Tumor Segmentation and Classification using MRI Images,” *Comput Methods Biomech Biomed Eng Imaging Vis*, vol. 11, no. 5, pp. 1921–1936, Sep. 2023, doi: 10.1080/21681163.2023.2199891;CTYPE:STRING:JOURNAL.

[34] N. Ghadimi, E. Yasoubi, E. Akbari, M. H. Sabzalian, H. A. Alkhazaleh, and M. Ghadamyari, “RETRACTED: SqueezeNet for the forecasting of the energy demand using a combined version of the sewing training-based optimization algorithm,” *Heliyon*, vol. 9, no. 6, p. e16827, Jun. 2023, doi: 10.1016/J.HELIYON.2023.E16827.