A Hybrid Artificial Bee Colony Strategy for $t$-way Test Set Generation with Constraints Support

Ammar K Alazzawi$^{1*}$, Helmi Md Rais$^1$, Shuib Basri$^1$, Yazan A. Alsariera$^4$, Abdullateef Oluwagbemiga Balogun$^{1,2}$, Abdullahi Abubakar Imam$^{1,3}$

$^1$Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Bandar Seri Iskandar 32610, Perak, Malaysia
$^2$Department of Computer Science, University of Ilorin, PMB 1515, Ilorin, Nigeria.
$^3$Department of Computer Science, Ahmadu Bello University, Zaria, Nigeria.
$^4$Department of Computer Science, Northern Border University, Arar 73222, Saudi Arabia

ammar_16000020@utp.edu.my

Abstract. $t$-way interaction testing is a systematic approach for exhaustive test set generation. It is a vital test planning method in software testing, which generates test sets based on interaction between parameters to cover every possible test sets combinations. $t$-way strategy clarifies the interaction strength between the number of parameters. However, there are some test sets combinations that should be excluded when generating the final test set as a result of invalid outputs, impossible or unwanted test sets combinations (e.g. system requirements set). These types of set combinations are known as constraint's combinations or forbidden combinations. From existing studies, several $t$-way strategies have been proposed to address the test set combination problem, however, generating the optimal test set is still open research being an NP-hard problem. Therefore, this study proposed a novel hybrid artificial bee colony (HABC) $t$-way test set generation strategy with constraints support. The proposed approach is based on a hybrid artificial bee colony (ABC) algorithm with a particle swarm optimization (PSO) algorithm. PSO was integrated as the exploratory agent for the ABC hence the hybrid nature. The information sharing ability of PSO via the Weight Factor is used to enhance the performance of ABC. The output of the hybrid ABC is a set of promising optimal test set combinations. The results of the experiments showed that HABC outperformed and yielded better test sets than existing methods (HSS, LAHC, SA_SAT, PICT, TestCover, mATEG_SAT).

1. Introduction
Practically, testing any particular software exhaustively is infeasible due to several factors such as cost, resource's constraints, and the large input parameters. Therefore, many researchers have proposed several techniques such as equivalent partitioning and boundary value analysis [1], base-choice and each-choice [2], random testing [3] and anti-random [4] to minimize the exhaustive test cases to a small optimal number of test cases (called test set) that be able to fault's detection. Regardless of their utility, these techniques are not intended to handle the faults because of the interaction between many input's parameters. For this reason, exhaustive testing is infeasible by using these techniques for a system under test (SUT). One of the proposed techniques in literature that have been found to be effective is $t$-way testing [5, 6].
Existing $t$-way strategies generate the test set to cover every single combination generated due to the interaction between parameters. In this case, where $(t)$ clarify the interaction strength between the number of parameters. However, there are some test sets combinations that should be excluded when generating the final test set as a result of invalid outputs, impossible or unwanted test sets combinations (e.g. system requirements set). These types of set combinations are known as constraint's combinations or forbidden combinations.

In the quest to develop an optimal $t$-way strategy, several variants of $t$-way strategies have been proposed in the literature. These variants are based on meta-heuristics such as the bat algorithm [7-12], artificial bee colony [13-16], Particle Swarm [17], Cuckoo Search [18], Ant Colony Algorithm [19], Genetic Algorithm [20], kidney algorithm [5] and so on. However, relatively few of the existing $t$-way variant strategies such as Pairwise Independent Combinatorial Testing (PICT) [21], Simulated Annealing [22, 23], Modified Automatic Efficient Test Generator (mAE TG) [24], TestCover [25] and Late Acceptance Hill Climbing (LAHC) [26] have addressed the constraint's combination problem sufficiently during the test set generation.

Supplementing to earlier research works, one of the algorithms proposed by Karaboge in 2005 is Artificial Bee Colony (ABC). ABC is a meta-heuristics algorithm that mimics the foraging behavior of a honeybee within the hive [27]. The ABC executes some certain tasks by the bees, where these bees are divided into three types of bee; everyone has a certain job inside the hive in order to increase the amount of nectar. Nonetheless, the ABC algorithm is similar to other artificial intelligence (AI) algorithms with respect to achieving the global optimum for optimization problems due to its randomization characteristics. For this reason, this research proposed a new optimization algorithm called hybrid artificial bee colony (HABC) algorithm. The proposed approach is based on a hybrid artificial bee colony (ABC) algorithm with a particle swarm optimization (PSO) algorithm. PSO was integrated as an exploratory agent for the ABC. The goal of HABC is to generate optimal test set combinations.

The paper is ordered as follows: Section 2 explains the covering array notations. Section 3 describes the related work. Section 4 describes the proposed test case generation algorithm. Section 5 evaluates the HABC through different benchmarking experiments. Section 6 concludes the research and suggests possible future work improvements.

2. Covering array notation

The interaction test set generation can be represented mathematically utilizing the covering array notation (CA). Generally, CA has four components $N$, parameters ($P$), value ($V$) and $(t)$ that indicate to the interaction strength between parameters for CA. For example, CA ($25; 2, 5^5$) indicates the number of the test case (i.e. test set size) that consist of 25 test case (i.e. rows) and six parameters (i.e. column). In this system, the test set size covers 2-way interaction strength with six parameters each one associated with five values. The CA case requires all the parameters to have similar value numbers [8, 28].

Mixed covering array (MCA) is similar to CA with its three components which can be represented mathematically utilizing MCA ($N; t, V_1^{P1} V_2^{P2} V_3^{P3} \ldots V_n^{Pn}$). MCA carries the same meaning of CA parameters but with supporting the parameters to carry different values. For example, MCA ($36, 2, 5^3 6^1 2^3$) indicates that the number of the test case (i.e. test set size) consists of 36 test case that covers 2-way interaction strength. This system is composed of 14 parameters (three parameters with each one associated with five values, four parameters with each one associated with six values, and seven parameters with each one associated with two values respectively) [29].

In addition to CA and MCA, there is a need for constraints covering array (CCA) and mixed constraints covering array (MCCA) for a forbidden combination (i.e. disallowed combination in the final test cases). The forbidden combination are represented by (i.e., CCA ($N; t, V^C, C$) and MCCA ($K; t, N; t, V_1^{P1} V_2^{P2} V_3^{P3} \ldots V_n^{Pn}, C$) respectively. In this case, a new symbol has been introduced called (C) which indicate the set of forbidden combination. For example, CCA ($N; 2, 2^2, C$), where $C= \{(1, -1, 0, -1), (-1, 1, 0, -1)\}$. Where "-1" indicates trivial value. This configuration system denotes the test set
size of 16 for 2-way interaction. Therefore, the interaction combinations of (1, -1, 0, -1) and (-1, 1, 0, 1) are forbidden in the final test cases [30, 31].

3. Related work
Over the past years, researchers have proposed several $t$-way strategies for test set generation. Most of the proposed earlier strategies have been focusing on uniform interaction (i.e. these strategies required the values of each parameter are equal to each other). The interaction strength of the earliest strategies focusing on pairwise interaction (e.g. $t=2$). Therefore, most of the pairwise strategies relied on the orthogonal array (OA) [32, 33], and Mutual orthogonal array (MOA) [3]). However, the strategies are still restricted to a small configurations system in spite of their respective fast execution time.

In view of the limitations aforementioned, researchers shifted focus to support the interaction strength for more than $t=2$ (where called by uniform $t$-way strategies) such as TConfig [34], TCG [35], AETG [36], MIPOG [37], Jenny [38], GTWay [39], KA [5], GA [6], BSA [40], ZZD [41] and ABC [13-16]. Although these strategies proved useful, the above-mentioned strategies still do not support the constraint's interaction. Therefore, this section provides a brief overview of existing strategies that support the constraints in view of the scope of this study.

Czerwonka proposes PICT [42] as a test set generation strategy. This strategy is utilized in a wide manner for software testing purposes in Microsoft. At first, PICT begins by generating all potential combinations based on the SUT configuration system. Then PICT distinguishes every combination as an uncovered combination. At this stage, PICT will check for any constraint's combinations as declared by a test engineer. Therefore, the constraint's combinations will be distinguished as eliminated combinations (or called excluded combinations). Thereafter, one of the marked uncovered combinations will be selected, and extended to cover as much as a possible uncovered combination by using the greedy heuristic method until completion without excluding any combination. The accomplished test case will be listed as its final test set, and the covered combinations by this test case will be marked as covered. This procedure will be repeated until all combinations are covered.

Cohen proposes Automatic Efficient Test Generator (AETG) [36, 43] based on a greedy search algorithm which generates the final test set that supports the interaction strength up to $t=2$. The generated test cases are extremely non-deterministic. Over the years, AETG has been extended to a number of variations such as mAETG_SAT and AETGm [44]. mAETG_SAT was the only variants that support constraints, unlike AETG and AETGm.

TestCover is a commercial $t$-way strategy that has proposed by Sherwood for test set generation [25]. There are no reported details in the literature regarding implementation. There is only a website provide benchmark configurations of the constraints.

SA_SAT strategy was proposed by Cohen [45] based on the Simulated Annealing (SA) as defined by Stardom [46, 47]. SA provides the support for constraints out of its SA_SAT variant. SA_SAT depends on the random search space to generate a test set. Based on the binary search algorithm; SA_SAT tries to find the best test case per iteration to list it in the final test set. However, SA addresses the interaction strength up to 3 (i.e., $t \leq 3$) in the reported literature. CASA another variant of SA was proposed by Garvin [48] to support the constraints. The empirical studies proved that SA has successfully been deployed in software product lines (SPL) testing.

Alsewari [31] proposed Harmony Search Strategy (HSS) for test set generation. HSS is one of the existing strategies that addressed the support for constraints by mimicking musician behavior to create good music. Iteratively, HSS executes the global search by inserting randomizing values to the Harmony memory whereby the local best value can select a given considering rate probability. The best value at each iteration will be added to the final test set until all required interactions are covered.

BA and LAHC probably are the most-recent $t$-way strategies that support the constraints for pairwise and $t$-way test set generation respectively. Bat algorithm (BA) uses echolocation behavior of bats [8, 11]. BA has been used to address an optimization problem with software engineering as no exception [11]. BA yielded promising results for test case generation. As for the Late Acceptance Hill Climbing (LAHC) strategy, LAHC adopts a greedy approach for searching for the best combination;
that is, only the best candidate’s test case is considered for the next cycle [26]. Unlike standard hill climbing algorithm, LAHC adopts buffer for storing previous best test cases and makes the comparison late in order to achieve optimum results. The effect of the buffer also ensures that LAHC avoids the local minima problem.

4. A hybrid artificial bee colony algorithm

Artificial Bee Colony (ABC) algorithm is one of the artificial intelligence (AI) algorithms proposed by Karoboge in 2005 [27]. It mimics the foraging behavior of the honeybee colony. ABC algorithm has widely used to solve optimization problems due to its effectiveness and performance. ABC has several mechanisms such as bee foraging, queen bee, nest site selection, task selection, collective decision making, navigation systems, floral/pheromone laying, mating and bee dance (communication) which helps it to implement intelligent search [39]. However, the ABC algorithm also has similar issues as other AI algorithms. This is due to its randomization, which leads to a fast convergence speed but with a weak performance for information sharing activity [49]. To overcome these limitations, several changes have been done by researchers on original ABC such as exploration and exploitation, convergence speed, ability to avoid being trapped at the local optimum [49, 50]. The changes include modifying the original ABC or hybridizing ABC with other optimization algorithms. The main focus is on the solution improvement and information conveying activity, which is the dual most vital parameter of exploitation and exploration ability of the algorithm.

For this reason, this research proposed new optimization algorithm will adopt called hybrid artificial bee colony (HABC) algorithm based on hybridize of an artificial bee colony (ABC) algorithm with a particle swarm optimization (PSO) algorithm. From this research, we believe there is an opportunity for ABC algorithm improvement deriving from particles in the PSO algorithm operation. The mechanism of PSO are one of a kind in terms of information sharing and solution improvement processes from the existing ABC. PSO it requires a crucial and unique parameter termed as Weight Factor (w). Referring to previous solution, the velocity parameter is vital for required improvement degree control. Besides velocity, there are dual factors (C1 and C2) which are required for relative influence of cognitive (self-confidence) and social (swarm-confidence) component's determination, respectively. The particles in PSO depends on three categories for the movement operation or the local search. The PSO movement depends on variation of velocity will affect every subsequent particle's movement. It can be assumed that particle's movement is not random or arbitrary. The local best solution variable generates best local information, which later links with the chosen particle's next move value. The particle’s next move best dependent on global best solution variable. Thus, this study hybridizes ABC and PSO to resolve the optimization problem.

The HABC behavior is designed depend on the simulating the foraging behavior of bee colony inside the hive and the PSO mechanism in terms of information sharing and solution improvement processe. The HABC is divided into three types of bee Employed bees, Onlooker bees, and Scout bees. These bees executes a specific task to find the best food source inside the hive. Employed bees are given the task of exploring the higher nectar potential food source and convey the information to standby onlooker bees at the hive. The information includes the navigation, location and potential of the food source. Selection processes of food source made by onlooker bees are in reference to the information disseminated by the employed bees. Scout bees are derived from employed bees and assigned to search the environment randomly for new or better food source discovery. The HABC algorithm behavior can be depicted as follows:

1. Initial phase: - using equation (1), the algorithm initiates the food source discovery by randomly searching the environment (i.e. HABC algorithm initial process is same the original in ABC), provided it is within algorithm's parameters border; producing the initial food sources.

\[ x_{ij} = x_{\text{min},j} + \text{rand}(0, 1)(x_{\text{max},j} - x_{\text{min},j}) \]

2. Employed bee phase: -Upon identified, the food source information will be detected by the employed bees. The employed bee numbers equates to number of food source, as the ratio is 1:1. The
employed bees will gather the nectar and return to hive with all the standby bees to convey the information by dancing in the dance area. The employed bee then transformed into a scout bee if the nectar of source is running out so it can begin to search for better food source. The discovery for new food source is expressed by using the PSO local search as below equation (2).

\[ V^{t+1}_{ld} = W^t * V^t_{ld} + C^1_i * r_1 * (pbest^t_{ld} - X^t_{ld}) + C^2_i * r_2 * (gbest^t_{ld} - X^t_{ld}) \]  

(2)

After detecting the food source, the probability of selecting a food source is defined by using (3):

\[ fitness_i = \begin{cases} 
\frac{1}{1 + f_i}, & \text{if } f_i \geq 0 \\
1 + |f_i|, & \text{if } f_i < 0 
\end{cases} \]  

(3)

3. Onlooker bee phase: The amount of nectar is the main criterion for onlooker bee food source selection activity. The information on profitability of the food nectar was transmitted by employed bee on the dance area. The food source selection probability is expressed using equation (4):

\[ P_i = \frac{fit_i}{\sum_{n=1}^{m} fit_n} \]  

(4)

4. Scout bee and Limit phase: Upon completion of their tasks, both employed and onlooker bees, the search continue for any remaining source that are not explored yet. The credibility of the food source is calculated using limit equation (5). Limit is a control parameter where the counter value of the food source is compared to algorithm, if it is superior to the limit, the food source will be disregard. A better food source discovered by scout bee will replace the disregard food source. The main steps of the HABC algorithm are shown as follows in Fig 1.

\[ limit = c . n e . D \]  

(5)

Where Ne indicate to the number on onlooker and scout bee (i.e. unemployed bees), while c is constant coefficient. The recommended value for constant coefficient is 0.5 or 1 [27].

---

1: Initialization step: The same process as the original ABC and PSO algorithm.
2: REPEAT
3: Move the employed bees onto their food sources and determine their nectar amounts.
4: Calculate the probability value of the source with which they are preferred be the onlooker’s bee.
5: Move the onlooker onto the food sources and determine their nectar amounts.
6: Move the scouts to search for new food sources replacing the abandoned ones.
7: Memorize the best food source found so far.
8: UNTIL (requirements are met).

**Figure 1.** HABC algorithm Pseudocode.

5. Results

Continuation to previous work [29, 51-54], this section evaluates the proposed HABC strategy compared to the existing tools and strategies such as mAETG_SAT, Test Cover, HSS, PICT, SA_SAT, and LAHC. The benchmarking data of the tools and strategies were adopted from the published results in [8, 26, 31, 55]. The HABC strategy parameters were set at Nbees = 5, maxCycle = 1000, limit = 100, C1 & C2 = 2.0 and W = 0.9. The experiments were implemented on a Windows 7
(OS) desktop computer with 3.40 GHz Xeon (R) CPU E3 and 8GB RAM. The Java language JDK 1.8. it was used to code and implement the HABC.

Due to the randomization characteristic of the proposed HABC strategy, the experiment runs twenty independent times for each configuration system to get the best result. Tables 1 and 2 present the experimental result, and each table presented the optimal test set size for each configuration. The dark cell with (*) represents the optimal test set size. The following constraint's configurations system is the used CCA in the conducted experiments:

1. Case 6, Configuration system CCA (N, 3,5,4,2, F{ })
   Where C = {(3, -1, 2, -1), (-1, 1, -1, 4), (2, 3, -1, -1), (1, -1, 3, -1)}.

2. Case 7, Configuration system CCA (N, 3,6,4,2, F{ })
   Where C = {(4, -1, -1,2), (-1, -1, 3, 1),( -1, 2, -1, 2), ( -1, 1, 2, -1)}. 

3. Case 8, Configuration system CCA (N, 3,7,4,2, F{ })
   Where C = {(-1, 2, 6, -1), (-1, 5, 6, -1), (-1, 4, 2, -1), (-1, 0, 1, 5), (-1, -1, 2, 4), (2, -1, 6, -1)}. 

4. Case 9, Configuration system CCA (N, 4,3,5,2, F{ })
   Where C = {( 1, 1, 1, 1, -1), (-1, 0, 0, 0, 0)}. 

As shown in Table 1, SA_SAT strategy has produced the most optimal test set size for the first three configuration systems compared with other strategies. HABC has produced the optimal test set size for CCA (N, 4, 3, 5) as well as the closest results to the optimal compared with other strategies. In Table 2, it is the same conducted experiments but with the constraints. HABC strategy produced the most optimal test set size for CCA (N, 3, 5, 4, C {}), CCA (N, 3, 6, 4, C {}) and CCA (N, 4, 3, 5, C {}) respectively compared with other strategies. While HSS outperforms all other strategies by produced 377 test cases for CCA (N, 3, 7, 4, C {}).

**Table 1. Results for configurations system without constraints.**

| No. | HSS | LAHC | SA_SAT | mATEG_SAT | PICT | TestCover | HABC |
|-----|-----|------|--------|-----------|------|-----------|------|
| 1   | 138 | 145  | **127** | 143       | 151  | NS        | 137  |
| 2   | 240 | 253  | **222** | 247       | 260  | NS        | 236  |
| 3   | 377 | 409  | **351** | 395       | 413  | NS        | 385  |
| 4   | 89  | 94   | NS     | NS        | NS   | NS        | **81** |


Table 2. Results for configurations system with constraints.

| No. | HSS  | LAHC | SAT | mATEG | PICT | TestCover | HABC |
|-----|------|------|-----|-------|------|-----------|------|
| 1   | 139  | 141  | 140 | 138   | 143  | NS        | 136* |
| 2   | 238  | 245  | 251 | 241   | 250  | NS        | 237* |
| 3   | 377* | 395  | 438 | 383   | 401  | NS        | 381  |
| 4   | 97   | 93   | NS  | NS    | NS   | NS        | 91*  |

6. Conclusion and future work
This paper proposed a HABC strategy inspired by the hybridization between the artificial bee colony (ABC) algorithm and particle swarm optimization (PSO) algorithm for constraints support. The outcome result was encouraging. Where HABC strategy outperformed other strategies by producing the best test set. As part of our future work, we are looking forward to supporting a higher interaction strength for more than \( t=6 \). In addition, we are planning to improve the HABC strategy to cover all interaction testing methods such as input-output \( t \)-way interaction and mixed-covering array.

References
[1] Myers, G.J., C. Sandler, and T. Badgett, The art of software testing. 2011: John Wiley & Sons.
[2] Ammann, P. and J. Offutt. Using formal methods to derive test frames in category-partition testing. in Proceedings of COMPASS’94, IEEE 9th Annual Conference on Computer Assurance, 1994. IEEE.
[3] Mandl, R., Orthogonal Latin squares: an application of experiment design to compiler testing. Communications of the ACM, 1985. 28(10): p. 1054-1058.
[4] Malaiya, Y.K. Antirandom testing: Getting the most out of black-box testing. in Proceedings of Sixth International Symposium on Software Reliability Engineering. ISSRE'95. 1995. IEEE.
[5] Homaid, A.A.B., et al., A Kidney Algorithm for Pairwise Test Suite Generation. Advanced Science Letters, 2018. 24(10): p. 7284-7289.
[6] Esfandyari, S. and V. Rafe, A tuned version of genetic algorithm for efficient test suite generation in interactive \( t \)-way testing strategy. Information and Software Technology, 2018. 94: p. 165-185.
[7] Alsariera, Y.A., A. Nasser, and K.Z. Zamli, Benchmarking of Bat-inspired interaction testing strategy. International Journal of Computer Science and Information Engineering (IJCSIE), 2016. 7: p. 71-79.
[8] Alsariera, Y.A., et al., A Bat-Inspired Testing Strategy for Generating Constraints Pairwise Test Suite. Advanced Science Letters, 2018. 24(10): p. 7245-7250.
[9] Alsariera, Y.A. and K.Z. Zamli, A bat-inspired strategy for \( t \)-way interaction testing. Advanced Science Letters, 2015. 21(7): p. 2281-2284.
[10] Alsariera, Y.A., M.A. Majid, and K.Z. Zamli, Adopting the bat-inspired algorithm for interaction testing, in The 8th edition of annual conference for software testing. 2015. p. 14.
[11] Alsariera, Y.A., M.A. Majid, and K.Z. Zamli, SPLBA: An interaction strategy for testing software product lines using the Bat-inspired algorithm, in 4th International Conference on Software Engineering and Computer Systems (ICSECS). 2015. IEEE. p. 148-153.
[12] Alsariera, Y.A., M.A. Majid, and K.Z. Zamli, A bat-inspired Strategy for Pairwise Testing. ARPN Journal of Engineering and Applied Sciences, 2015. 10: p. 8500-8506.
[13] Alazzawi, A.K., H.M. Rais, and S. Basri, Artificial Bee Colony Algorithm for t-Way Test Suite Generation, in 2018 4th International Conference on Computer and Information Sciences (ICCOINS). 2018, IEEE. p. 1-6.
[14] Alazzawi, A.K., et al., Artificial Bee Colony Algorithm for Pairwise Test Generation. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 2017. 9(1-2): p. 103-108.
[15] Alazzawi, A.K., H. Rais, and S. Basri, ABCVS: An Artificial Bee Colony for Generating Variable T-Way Test Sets. International Journal of Advanced Computer Science and Applications, 2019. 10(4): p. 259-274.
[16] Alsewari, A.A., et al., ABC Algorithm for Combinatorial Testing Problem. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 2017. 9(3-4): p. 103-108.
[17] Chen, X., et al. Applying particle swarm optimization to pairwise testing in Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual. 2010. IEEE.
[18] Nasser, A.B., et al., A cuckoo search based pairwise strategy for combinatorial testing problem. Journal of Theoretical and Applied Information Technology, 2015. 82(1): p. 154.
[19] Shiba, T., T. Tsuchiya, and T. Kikuno. Using artificial life techniques to generate test cases for combinatorial testing. in Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International. 2004. IEEE.
[20] McCaffrey, J.D. Generation of pairwise test sets using a genetic algorithm. in Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International. 2009. IEEE.
[21] Cohen, M.B., Designing Test Suites for Software Interactions Testing. 2004, University of Auckland (NEW ZEALAND): Department of Computer Science.
[22] Rahman, M., et al. Event Driven Input Sequence T-way Test Strategy Using Simulated Annealing. in Intelligent Systems, Modelling and Simulation (ISMS), 2014 5th International Conference on. 2014. IEEE.
[23] Cohen, M.B., C.J. Colbourn, and A.C. Ling. Augmenting simulated annealing to build interaction test suites. in Software Reliability Engineering, 2003. ISSRE 2003. 14th International Symposium on. 2003. IEEE.
[24] Cohen, M.B., M.B. Dwyer, and J. Shi. Interaction testing of highly-configurable systems in the presence of constraints. in the 2007 international symposium on Software testing and analysis. 2007. ACM.
[25] G, Sherwood. Testcover.com. 2006, [Online]. http://testcover.com/.
[26] Nasser, A., et al., Late acceptance hill climbing based strategy for addressing constraints within combinatorial test data generation. 2014.
[27] Karaboga, D., An idea based on honey bee swarm for numerical optimization. 2005, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
[28] Esfandyari, S. and V. Rafe, Tuned Version of Genetic Algorithm for Efficient Test Suite Generation in Interactive t-way Testing Strategy. Information and Software Technology, 2017. 94: p. 165-185.
[29] Alazzawi, A.K., H.M. Rais, and S. Basri, HABC: Hybrid Artificial Bee Colony For Generating Variable T-Way Test Sets. Journal of Engineering Science and Technology, 2019. 7(12): p. 13.
[30] Ahmed, B.S., et al., Handling constraints in combinatorial interaction testing in the presence of multi objective particle swarm and multithreading. Information and Software Technology, 2017. 86: p. 20-36.
[31] Alsewari, A.R.A. and K.Z. Zamli, Design and implementation of a harmony-search-based variable-strength t-way testing strategy with constraints support. Information and Software Technology, 2012. 54(6): p. 553-568.
[32] Hartman, A. and L. Raskin, Problems and algorithms for covering arrays. Discrete Mathematics, 2004. 284(1): p. 149-156.
[33] Hedayat, A.S., N.J.A. Sloane, and J. Stufken, Orthogonal arrays: theory and applications. 1999: Springer Science & Business Media.

[34] Williams, A., TConfig. download page [Online]. Available: http://www.site.uottawa.ca/~awilliam/ [Accessed 23 Dec 2014]

[35] Tung, Y.-W. and W.S. Aldiwan. Automating test case generation for the new generation mission software system. in Aerospace Conference Proceedings, 2000 IEEE. 2000. IEEE.

[36] Cohen, D.M., et al., The AETG system: An approach to testing based on combinatorial design. Software Engineering, IEEE Transactions on, 1997. 23(7): p. 437-444.

[37] Younis, M.I. and K.Z. Zamli, MC-MIPOG: A parallel t-way test generation strategy for multicore systems. ETRI journal, 2010. 32(1): p. 73-83.

[38] Jenkins, Jenny. http://www.burtleburtle.net/bob/math/, 2003.

[39] Zamli, K.Z., et al., Design and implementation of a t-way test data generation strategy with automated execution tool support. Information Sciences, 2011. 181(9): p. 1741-1758.

[40] Cai, L., Y. Zhang, and W. Ji, Variable Strength Combinatorial Test Data Generation Using Enhanced Bird Swarm Algorithm, in 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). 2018, IEEE. p. 391-398.

[41] Ohashi, T. and T. Tsuchiya. Generating High Strength Test Suites for Combinatorial Interaction Testing Using ZDD-Based Graph Algorithms. in Dependable Computing (PRDC), 2017 IEEE 22nd Pacific Rim International Symposium on. 2017. IEEE.

[42] Czerwonka, J. Pairwise testing in real world. in 24th Pacific Northwest Software Quality Conference. 2006. Citeseer.

[43] Cohen, D.M., et al., The combinatorial design approach to automatic test generation. IEEE software, 1996. 13(5): p. 83-88.

[44] Cohen, M.B., M.B. Dwyer, and J. Shi. Exploiting constraint solving to construct interaction test suites. in Testing: Academic and Industrial Conference Practice and Research Techniques-MUTATION, 2007. TAICPART-MUTATION 2007. 2007. IEEE.

[45] Cohen, M.B., M.B. Dwyer, and J. Shi. Interaction testing of highly-configurable systems in the presence of constraints. in Proceedings of the 2007 international symposium on Software testing and analysis. 2007. ACM.

[46] Cohen, M.B., M.B. Dwyer, and J. Shi, Constructing interaction test suites for highly-configurable systems in the presence of constraints: A greedy approach. IEEE Transactions on Software Engineering, 2008. 34(5): p. 633-650.

[47] Stardom, J., Metaheuristics and the search for covering and packing arrays. 2001: Simon Fraser University.

[48] Garvin, B.J., M.B. Cohen, and M.B. Dwyer, Evaluating improvements to a meta-heuristic search for constrained interaction testing. Empirical Software Engineering, 2011. 16(1): p. 61-102.

[49] Karaboga, D. and B. Akay, A survey: algorithms simulating bee swarm intelligence. Artificial Intelligence Review, 2009. 31(1-4): p. 61-85.

[50] Yan, X., Y. Zhu, and W. Zou, A hybrid artificial bee colony algorithm for numerical function optimization, in Hybrid Intelligent Systems (HIS), 2011 11th International Conference on. 2011, IEEE. p. 127-132.

[51] Alazzawi, A.K., et al., Pairwise Test Suite Generation Based on Hybrid Artificial Bee Colony Algorithm, in Advances in Electronics Engineering. 2020, Springer. p. 137-145.

[52] Alazzawi, A.K., et al. PhABC: A Hybrid Artificial Bee Colony Strategy for Pairwise test suite Generation with Constraints Support. in 2019 IEEE Student Conference on Research and Development (SCORed). 2019. IEEE.

[53] Alazzawi, A.K., H.M. Rais, and S. Basri, Parameters Tuning of Hybrid Artificial Bee Colony Search based Strategy for t-way Testing. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019. 8(5S).
[54] Alazzawi, A.K., H.M. Rais, and S. Basri. Hybrid Artificial Bee Colony Algorithm for t-Way Interaction Test Suite Generation. in Computer Science On-line Conference. 2019. Springer.

[55] Li, L., Y. Cui, and Y. Yang. Combinatorial test cases with constraints in software systems. in Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on. 2012. IEEE.