

Binary Black Hole-Based Optimization For T-Way Testing

Hamsa N Nsaif¹ and Dayang Norhayati Abang Jawawi¹

¹School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM), 81310 UTM Skudai, Johor Bahru, Johor, Malaysia.

E-mail: hamsa6000@yahoo.com

Abstract. Software testing is an important process in software development life cycle, which aims to guarantee the quality of software and reduce the number of errors and bugs. In such a process, software inputs and parameters are used to create a set of testing cases. Nevertheless, the number of testing cases increases enormously when considering all combinations of those inputs. Although t-way testing can reduce the test cases, generating the minimum, yet representative t-way testing set is challenging due to the large search space, which renders finding the best solution computationally prohibitive. The extant solutions suffer from the sensitivity to the random initialization and the subjectivity to the local minima, which adversely affects the reproducibility of these algorithms and obstructs finding the optimal solution. To this end, this paper proposes a novel meta-heuristic searching algorithm called Binary Black Hole (BBH) optimization that formulates the t-way testing as a binary optimization problem. Experimental results show the superiority of BBH over the famous Binary Particle Swarm Optimization (BPSO) algorithm. The achieved improvement shows the capability of BBH in generating smaller covering arrays with the same t-strength compared to those generated by BPSO.

1. Introduction

In the recent years, software emerged rapidly and proved itself as a core enabler of the technologies in many fields in civilian and military applications [1]. Almost all systems in today's technologies are based on sophisticated software and programs. As such, reducing the bugs in software is essential for smooth and effective deployment of the applications in the real-world [2]. To achieve such a goal, software testing is a crucial step in Software Development Life Cycle (SDLC).

To ensure the high quality, software undergoes several tests. However, number of the tests need to be carried out may grow rapidly due to the enormous size of software inputs including parameters and their value dependencies, which renders the testing process impractical [3]. Consequently, the overall cost of software development increases proportional to the number of tests. For example, the software application with 10 inputs and five values for each input produces a list with $5^{10} = 9,765,625$ tests. If each test case requires one minute to be executed, the execution time will be 9,765,625 minutes, which is impractical to be applied in industry. To address such problem, the combinatorial software testing method; also called t-way testing; that involves the interaction of parameters was employed by several studies [2, 4, 5]. The letter t represents the interaction strength. Such interaction factor improves the testing efficiency as it can guarantee fault detection ability with small number of test cases [6]. Using the interaction testing (t-way) strategy, the number of test cases can be reduced considerably. However, the generation of the minimum t-way test set is challenging due to the large search space [4], which makes finding the optimal solution an NP-Complete (Non-deterministic Polynomial-time) problem.



Solving combinatorial optimization problems often requires finding one or more optimal solutions within a specified solution space. Different approaches, including pure computational-based and artificial intelligence-based approaches are adopted in the existing t-way strategies, among these approaches the nature-inspired algorithms, such as Particle Swarm Optimization (PSO) , Harmony Search Algorithm (HSA) [3], and Bat Algorithm (BA) [6]. Despite the advantages of both computational-based and AI-based approaches in achieving satisfactory outcomes, the AI-based approach seems promising when optimality is required. However, those methods rely on controlling parameters to balance between the global and local searches, which increases the complexity of the algorithm. In addition, these algorithms are susceptible to be trapped into local optima [7].

To address this issue, this paper focuses on special method from meta-heuristic searching optimization called black hole (BH) optimization [1]. Contrary to other metaheuristic algorithms, BH does not require any controlling parameters for balancing between the global and local search abilities of the algorithm. Therefore, it has a simple and easy-to-implement structure [1]. In addition, BH is not susceptible to local optima which makes it suitable when searching for optimal solution [7]. However, BH in its current form is not suitable to apply on the software combinatorial testing due to the binary nature of the coverage array used by the t-way testing. Therefore, this paper proposes the Binary Black Hole (BBH) algorithm that cope with such binary nature. The remaining of the article is organized as follows. Section 2 provides the related works. In section 3 presents the background about black hole concept and apply it as a meta-heuristic searching algorithm for optimization. Section 4 presents the methodology adapted by this study. Section 5 provides the experimental results and analysis. In Section 6, the conclusion, findings, and future work are given.

2. Related works

The literature of t-way testing contains numerous of approaches. The study conducted by [5] proposed genetic algorithm test suites. Their algorithm is combined of two parts: calculating the weight of test cases and generating the test suite using GA. Their chromosome encoding accepts integer numbers. Other researchers have aimed to improve the concept of covering array such as in [8]. In their work, [8] introduced the notion of Constrained Locating Arrays (CLAs), which are mathematical objects that can be used for fault localization in software testing.

The popularity of such algorithms has increased due the power of random searching when combined with heuristic guidance of the searching, especially with the increasing capability of the processing power nowadays. Some of the algorithms that were developed for solving t-way testing based on meta-heuristics are Genetic Algorithms GA, Ant Colony Optimization ACO, Simulated Annealing, Particle Swarm Optimization, and Cuckoo Search. All these algorithms preserved the formulation of t-way optimization and put much focus on the optimization encoding and procedure, which is essential to avoid local minima.

One of the recently developed meta-heuristic searching algorithms is the Black Hole (BH) algorithm [9]. BH is inspired by the black hole phenomenon and the behaviour of the stars when they interact with the black hole. If a star gets too close to the black hole, it will be swallowed by the black hole. In such a case, the algorithm will generate a new star randomly to represent a new solution and will place it in the search. This algorithm is a tuning-free, which is an attractive aspect that makes it attractive for the solving the t-way testing problem. That is, the algorithm does not need to tune numerous parameters like other meta-heuristic algorithms.

Although BH has been applied in various fields [10], it has not been used for solving the optimization problem in t-way testing. One obstacle toward using BH for solving the t-way testing problem is its continuous searching nature which is not compatible with the discrete (binary) nature of t-way testing. Thus, BH needs to be modified before using it in solving the t-way testing optimization problem.

3. The methodology

This section provides the methodology of optimization of t-way testing. It starts with the optimization framework in subsection 3.1. Next, the adaptation of BH algorithm to fit in the optimization framework

is done using new concept named as binary black hole optimization. This is presented in subsection 3.2. Then, the pseudocode of BBH presents in section 3.3.

3.1. Optimization framework of t-way testing

The goal of optimizing t-way testing is to obtain the minimum size of covering array CA with certain value of strength t , N is number of rows (test cases), k is number of variables, variable space is (v_1, v_2, \dots, v_k) . The approach is based on encoding the solution space and objective function for the BBH optimization. The input of the system is $CA(N, t, k, ((v_1, v_2, \dots, v_k)))$ then the output is an array CA with strength t and minimum size or number of rows. It has been encoded the solution space as a vector $x = (x_1 x_2 \dots x_N) \in \{0,1\}^N$ where $x_i = 1$ indicates that the corresponding row has been chosen to be added to the output matrix CA and $x_i = 0$ indicates that the corresponding row has not been chosen to be added to the output matrix CA. The objective function is the number of 1's in the solution. As mention that each solution has to meet the constraint of t-strength with respect to the variables space.as shown in figure 1.

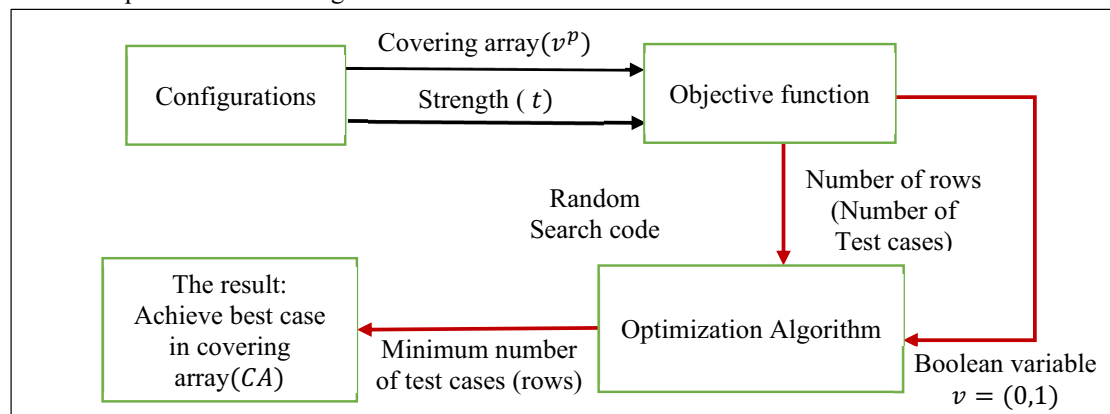


Figure 1. Optimization framework of t-way testing.

It has been found that the solution space of t-way testing is not continues; rather, it is a binary value because the optimization algorithm has to select which rows are chosen from the covering array CA to combine the t-strength CA. The binary nature of t-way testing makes it not straight forward to apply the BH algorithm, which is a non-binary algorithm. Thus, to solve this problem it needs to upgrade BH Algorithm BH to become Binary Black Hole Algorithm (BBH) to apply it in t-way testing.

3.2. Upgrade black hole to binary black hole algorithm BBH

Observing the equation (1) of moving the stars toward the black hole, as follows:

$$x_i(t+1) = x_i(t) + rand \times (x_{BH} - x_i(t)), i = 1, 2 \dots N \quad (1)$$

The problem with the equation (1) is that it does not work for binary values because the addition of the second term to a binary value might cause leaving the binary constraint which is 0 or 1. In order to overcome this problem, first, define a parameter called pulling rate pr .

After that generate a random number r_d between 0 and 1, where d denotes the index of the dimension. Then check its value and change the value of $x_i(t)^d$ to $x_i(t+1)^d$ according to the condition below in the Equation (2).

$$x_i(t+1)^d = \begin{cases} BH(t)^d & \text{if } r < pr \\ x_i(t)^d & \text{otherwise} \end{cases} \quad (2)$$

It is preferred to choose a small value of pulling rate pr in order to do careful searching in the solution space. In order to show how the Equation (2) works, to give the following example:

Assume that one star has the value $x_t = [1 \ 0 \ 0 \ 1]$ while the black hole $BH_t = [0 \ 0 \ 0 \ 0]$, $pr = 0.2$, then generate for each element of x_t one random number r^d . Let $r = [r^1 \ r^2 \ r^3 \ r^4] =$

[0.1 0.2 0.04 0.6], then based on the equation (2), the value of $x_i(t + 1) = [0 0 0 1]$, then see that the distance between the star and the black hole was $(\sqrt{2})$, after moving the star using the equation (2), the distance becomes one.

3.3. Pseudocode of BBH

The pseudocode of BBH is similar to BH with the difference in the moveStars function. In this function, it is needed to assure that the values of the solutions are binary values belongs to $\{0,1\}$. In order to do so, generate random number r and compare it with the pulling rate. The pseudocode of BBH as shows in figure 2.

Pseudo Code of BBH Algorithm

```

1. Start
2.  $P = \text{moveStars}(P, BH, Pr)$  // move the solutions toward the BH
3. For each star  $S$ 
4.   For each dimension  $d$  in star  $S$ 
5.      $r = \text{rand}(0,1)$ 
6.     If ( $r < Pr$ )
7.       make  $S(d)$  equal to  $BH(d)$ 
8.     Else
9.       do not change  $S(d)$ 
10. End

```

Figure 2. Pseudocode of binary black hole algorithm (BBH).

4. Experimental results and analysis

In order to evaluate our developed BBH approach for t-way testing, binary particle swarm optimization Algorithm (BPSO) were used for comparison. A BPSO was selected because it is considered as one of the most powerful methods for resolving the optimization problems [11]. In BPSO, solutions are represented as particles moving in the solution space according to the calculated velocity. Also, it conducted the smart mobile system problem in sub-section 4.1. The problem was tested using t-strength ranges from 2 until 4.

4.1. Benchmark Smart Mobile System Case Study

This problem was also used as a case study by [6]. It was selected because it offers wide factors and levels which makes it suitable for software testing. Exhaustively, there are possible $(1^8 \times 2^{10} = 1024)$ configurations that can be generated for the smart mobile system. In the results, as observed in table 1, BBH was able to reduce the full size from (1024 to 109) for $t=2$, $t=3$, and $t=4$. However, this was not achieved for BPSO where it was able to reduce the full size from (1024 to 223) only for $t=2$ and $t=3$. This is also observed in figure 3.

Table 1. Smart mobile system BBH vs. BPSO.

Algorithm	Strength t			Full Size (covering array)
	t= 2	t= 3	t= 4	
BPSO	223	223	1024	1024
BBH	109	109	109	1024

The comparison results, BBH is superior over BPSO when applied for the t-way testing. Due to the non-linear mobility nature of BBH compared to BPSO whose mobility is the linear. Another aspect in BBH is the event horizon which represents a region around the black hole to eliminate solutions that approaches the black hole and generate new solutions in other areas. These two factors are essential to boost the exploring power of BBH to find optimal solution in the search space more than BPSO.

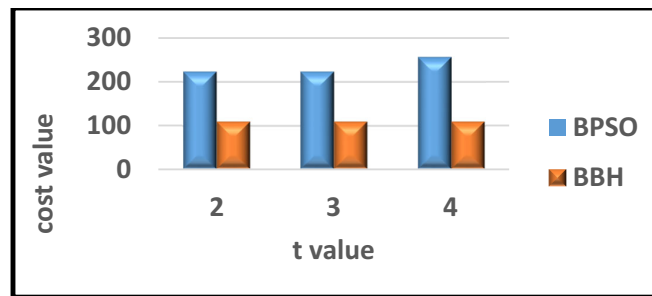


Figure 3. Smart mobile system BBH vs. BPSO.

5. Conclusion

This article has presented a novel optimization approach for solving the problem of t-way testing. The optimization approach is based on black hole optimization algorithm BH. Before applying it in t-way testing, a framework of optimization based on given problem configuration was provided. Next, an adaptation of BH to function in binary space was performed, it is named as binary black hole optimization BBH. For evaluating BBH, one problem was taken from the literature. Results reveal the superiority of BBH over BPSO in terms of the capability of reducing the size of covering array given certain strength. Future work is to boost the performance of BBH through changing it from one black hole to multi-black hole algorithm.

6. Acknowledgement

The authors fully acknowledge Universiti Teknologi Malaysia for UTM-TDR Grant Vot No. 06G23, and Ministry of Higher Education (MOHE) for FRGS Grant Vot No. 5F117, which have made this research endeavor possible.

References

- [1] Wu H, Nie C, Petke J, Jia Y and Harman M 2019 A Survey of Constrained Combinatorial Testing *arXiv preprint arXiv:1908.02480*
- [2] Rahman MS, Rivera E, Khomh F, Guéhéneuc Y-G and Lehnert B 2019 Machine Learning Software Engineering in Practice: An Industrial Case Study *arXiv preprint arXiv:1906.07154*
- [3] Alsewari ARA and Zamli KZ 2012 A harmony search based pairwise sampling strategy for combinatorial testing *Int. J. Phys. Sci.* **7** 1062-72
- [4] Al-sewari AA, Zamli KZ and Lin MX 2019 Firefly combinatorial testing strategy *Computing Conference 2018* (London, United Kingdom) p 936-44
- [5] Eksioğlu B, Vural AV and Reisman A 2009 The vehicle routing problem: A taxonomic review *Comput. Ind. Eng.* **57** 1472-83
- [6] Alsariera YA, Majid MA and Zamli KZ 2015 SPLBA: an interaction strategy for testing software product lines using the bat-inspired algorithm 2015 *4th International Conference on Software Engineering and Computer Systems (ICSECS)* (Kuantan, Malaysia) p 148-53
- [7] Shaheen AM, Spea SR, Farrag SM and Abido MA 2018 A review of meta-heuristic algorithms for reactive power planning problem *Ain Shams Engineering Journal* **9** 215-31
- [8] Kuhn R, Kacker R, Lei Y and Hunter J 2009 Combinatorial software testing *Computer* **42** 94-6
- [9] Farahmandian M and Hatamlou A 2015 Solving optimization problems using black hole algorithm *JACST*. **4** 68
- [10] Gholizadeh S, Razavi N and Shojaei E 2019 Improved black hole and multiverse algorithms for discrete sizing optimization of planar structures *Eng. Optimiz.* **51** 1645-67
- [11] Ahmed BS, Zamli KZ and Lim CP 2012 Constructing a t-way interaction test suite using the particle swarm optimization approach *IJICIC* **8** 431-51