ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 11(4), DOI: 10.17485/ijst/2018/v11i4/121086, January 2018*

Evaluating Bees Algorithm for Sequence-based T-way Testing Test Data Generation

M. H. Mohamed Zabil[1]*, K. Z. Zamli[2] and K. C. Lim[1]

[1]College of Computer Science and IT, Universiti Tenaga Nasional, Kajang, Malaysia; Hazli@uniten.edu.my, kokcheng@uniten.edu.my
[2]Faculty of Computer System and Software Engineering, Universiti Malaysia Pahang, Pekan, Malaysia; kamalz@ump.edu.my

Abstract

T-way testing is a testing technique that is used to detect faults due to parameter interactions or software configurations. In order to perform t-way testing, software testers need to prepare the test data. For a system with many configuration parameters, the test data could lead to combinatorial explosion problem, since all possible parameter combinations need to be considered. Therefore, test data generation for t-way testing need to be optimized. Many t-way test data generation strategies have been proposed in the literature to generate optimized t-way test data. However, very few strategies have been proposed for sequence-based t-way. This paper presents statistical analysis on the performance of Bees Algorithm against the other sequence t-way strategies, in order to generate test cases.

Keywords: Bees Algorithm, Optimization, Sequence-based T-way Testing, Test Case Generation

1. Introduction

One of the important objectives of software testing is to find as many faults as possible in a System Under Test (SUT). As there are many classes of systems, in order to achieve the objective, different types and approaches of testing might need to be perform (e.g. regression testing, performance testing, compatibility testing and interaction/combinatorial testing). For an event-driven system, faults may be triggered from a combination of events triggered during a process. To detect such faults, interaction testing or combinatorial testing can be performed against the SUT. This is to ensure that combination of any events will not flag any errors. In the literature, many combinatorial or interaction test strategy have been developed for the past 20 years (e.g. Jenny[1], IPOG[2] MC-IPOG[3] AETG[4] and TConfig[5], ACS[6], PICT[7], PSO[8], HSS[9], ParaOrder[10], Density[11], TVG[12] and ITTDG[13]). Although these strategies are able to detect faults due to combination or interaction between events, the sequence of events occur are not being considered. This would risk the SUT to faults due to sequence of event.

In an event-driven system, event can occur in many sequences. In order to detect the fault due to sequences of events, we need to test all possible sequence of event for the SUT. However, testing all possible sequence,

*Author for correspondence

even for a small system is inefficient and yet affordable due to resource constraints (i.e. time, budget and human resource).

In[14,15] have proposed a new approaches to generate test data for testing event-driven system. Their work, focus on systems with distinct number of event and each event occurs only once, since event repetition is not always the case in an event-driven system. Kuhn has been using computational greedy approach while Esra is proposing a rule-based approach using Answer Set Programming (ASP). Both approaches have its strength and limitations; hence we are looking into improving the limitations of the two approaches. The advantages and limitations of the two mentioned approaches are discussed in the next following section.

Researchers have been adopting Artificial Intelligence (AI) to solve combinatorial optimization problems. In the emerging research area of software testing, researchers have been proposing the use of AI approach in generating optimized test data for t-way testing (e.g GA[16], ACS[6], PSO[8] and HSS[9]). From the published results, in general, the AI-based strategies produce smaller test suite size compared to the computational strategies especially when it comes to higher number of parameters (events).

In[17], a new t-way test data generation strategy using Bees Algorithm has been proposed for generating test data for a sequence-based system. In this paper, we benchmark the performance of the Bees Algorithm strategy from[17] against other t-way strategies in generating sequence-based t-way test data. The benchmarking is done by comparing the different in final test suit size as well as using statistical approach to determine whether the differences in test suit size is significant or not.

For the purpose of presentation, this paper is organized as follows. Section 2 discusses the background; Section 3 elaborates the results and benchmarking against existing strategies; and Section 4 provides the conclusion.

2. Background

Although ideally and desirably, all possible combination of configurations need to be tested, in real world, exhaustive testing is not always practical and feasible[18]. This is due to resource (i.e. cost, human, time) and timing (such as time to market) constraints. As a result, many sampling strategies have been developed and commonly used such as equivalent partitioning, boundary value analysis, cause and effects graphing and decision table mapping. All of the mentioned sampling strategies are useful for input fault detection and prevention; these strategies are not sufficiently effective to detect faults due to interaction between input parameters.

In order to address this issue, researchers have been proposing many strategies in order to generate test suite for t-way testing. In the literature, many t-way strategies have been developed for the past 20 years. Chronologically, the t-way test data generation strategies started with Automatic Efficient Test Case Generator (AETG)[19,20]. AETG is a computational strategy based on greedy algorithm, a number of test cases are generated and the best test case (i.e. that covers the most intended combination) will be selected in each iteration. In-Parameter Order (IPOG)[21] generates test cases for the first t-parameters and then extend the test cases one parameter after another to cover all the intended combinations. Test Configuration[22] is a strategy based on mathematical algebraic approach to generate the final test suite. Later, Simulated Annealing[23] and Genetic Algorithm[24] are strategies that are using meta-heuristic algorithm. The former adopting the process of annealing while the latter is adopting chromosome mutation process. On top of that, Pairwise Independent Combinatorial Testing (PICT)[25] is another computational greedy strategy. In Parameter Order General[26] is an extended version of IPO that supports higher interaction strength (i.e. IPO only support pairwise or t = 2). Another strategy that is based on meta-heuristic algorithm is Ant Colony System[27] which adopt the ant food foraging behavior. Multi-Core-In Parameter Order General (MC-IPOG)[28] is a parallel implementation of IPOG. Integrated t-way Test-Suite Generator[29] is a strategy that integrates all three types of interaction strength in one strategy. Particle Swarm Optimization[30] Harmonic Search Strategy[31] and Cuckoo[32] are all meta-heuristic strategies adopting swarm intelligence, harmonic tuning of musical instrument and the behavior of Cuckoo birds laying their eggs in the nest of other host birds respectively.

T-way strategies generate test suites that provide coverage for as many interactions as possible for a set of input parameters. In t-way testing, input values for each parameter (e.g. all possible values for a configuration parameter) are selected and combined to generate a set of test cases or test suite. In t-way strategies, all possible combination of possible values of any t parameters is required to be tested at least once. Here, *t* indicates the interaction strength. If *t* equals to the number of parameters (i.e. full interaction strength), a state of exhaustive testing is reached, which is not desirable. Thus, in a typical setup, *t* is less that the total number of parameters. Although exhaustive interaction is not performed, empirical data shows that 100% faults can be exposed by considering a relatively low number of interaction (i.e. up to t = 6) among parameters[33,34].

Meta-heuristic approach is gaining its popularity in the field of optimization. In the literature, many meta-heuristic based algorithms yield better results compared to computational based algorithm especially when the size problem size become larger. While computational based algorithm is fast and deterministic, meta-heuristic algorithm is non-deterministic and relatively slower compared to computational based algorithm, but published works show that metaheuristic based produced better results especially when involving large search area. This is due to the heuristic search process in the algorithm that is used to improve the results in each iteration.

As mentioned above, there are a number of t-way strategies based on meta-heuristic algorithms. On the other hand, for sequence-based t-way testing, only a number of test data generation strategies have been proposed which includes computational approach and rule-based programming such as tseq[35] and ASP[15].

3. Results Analysis

Results in Table 1 is taken from[17] with additional row of 50 event sequence. For 3-way sequence and number of event that is less than 20, the rule based approach adopted in ASP outperforms both BA and t-seq. It is within our expectation that ASP able to produce best results. This is due to the advantage of the proposition logic processor that enables ASP to suggest test case that can cover the most tuples. However, as the size of event increases, (i.e. event sequence more than 20) the performance of BA is catching up with ASP. BA managed to produce similar test case size when number of event is 20 (for 3-way) and managed to outperforms ASP with slightly better results when it comes to 30, 40 and 50 events.

Table 1. Number of test suit size for 3-way and 4-way sequence

Event	3-way sequence			4-way sequence		
	t-seq	ASP	BA	t-seq	ASP	BA
5	8	7	8	29	-	26
6	10	**8**	**8**	36	-	35
7	12	**8**	10	46	-	41
8	12	**8**	10	50	-	50
9	14	**9**	12	58	-	57
10	14	**10**	12	66	**55**	64
11	14	**10**	13	70	-	69
12	16	**10**	15	78	-	77
13	16	**10**	14	86	-	81
14	16	**10**	16	90	-	86
15	18	**10**	16	96	-	91
16	18	**11**	17	100	-	97
17	20	**11**	18	108	-	99
18	20	-	18	112	-	104
19	22	-	18	114	-	107
20	22	**19**	19	120	104	104
21	22	-	20	126	-	116
22	22	-	21	128	-	120
30	26	**23**	23	156	149	145
40	32	**27**	26	182	181	175
50	34	**31**	30	214	-	190

As for 4-way sequence, the performance of BACA is consistent with results 3-way sequence. BA outperforms t-seq in almost all configurations. Furthermore, the difference in size of test cases between BACA and t-seq are relatively higher if compared to the results obtained from 3-way sequence.

As for results from ASP and BA for 4 event sequences, the results from ASP are only available for of 20, 30 and 40 events sequence. BA managed to produce same size of test suit for 20 event sequences, relatively 2.7% better for 30 event sequence and 3.3% better for 40 event sequence. Based on these trending, it is foreseen, the results for higher number of sequence, that is, more than 50, form BACA will still able to outperforms ASP.

4. Statistical Analysis

The performance of BA strategy in generating test cases can also be compared to ASP and t-seq by adopting statistical analysis approach. From the experiment results, each strategy has its own strength and weaknesses. In order

Table 2. Wilcoxon signed rank test for ASP and BACA

(a) Test Config.	(b) ASP x_i	(c) BACA y_i	(d) Observe difference $d_i=x_i - y_i$	(e) Absolute values of differences	(f) Sign	(g) Ordered absolute value of differences	(h) Ranks	(i) signed ranks d_{is}
1	7	8	-1	1		0	2.5	2.5
2	8	8	0	0		0	2.5	2.5
3	8	10	-2	2		0	2.5	2.5
4	8	10	-2	2		0	2.5	2.5
5	9	12	-3	3	-ve	1	6	-6
6	10	12	-2	2		1	6	6
7	10	13	-3	3		1	6	6
8	10	15	-5	5	-ve	2	9	-9
9	10	14	-4	4	-ve	2	9	-9
10	10	16	-6	6	-ve	2	9	-9
11	10	16	-6	6	-ve	3	11.5	-11.5

Table 2 Continued

12	11	17	-6	6	-ve	3	11.5	-11.5
13	11	18	-7	7	-ve	4	13.5	-13.5
14	19	19	0	0		4	13.5	13.5
15	23	23	0	0	-ve	5	15	-15
16	27	26	1	1	-ve	6	17.5	-17.5
17	31	30	1	1	-ve	6	17.5	-17.5
18	104	104	0	0	-ve	6	17.5	-17.5
19	149	145	4	4		6	17.5	17.5
20	181	175	6	6	-ve	7	20	-20

to assess the statistical evidence whether there is significance difference between results from BA compared to one other strategy in a particular configuration, the Wilcoxon signed-rank test (Wilcoxon Test) is performed. The Wilcoxon Test is a non-parametric test that allows us to compare two paired samples.

Table 3. Result of Wilcoxon test

Wilcoxon Test	**Statistical Result**	**Reject or Accept H_0**
BA vs. t-seq	Effective sample size is 21 $W^+ = 231$, $W^- = 0$ W statistic is 0 Wα at 0.05: 58	W < Wα **Reject** $H_{0:}$ result is significant
BA vs. ASP	Effective sample size is 20 $W^+ = 53$, $W^- = 157$ W statistic = 53 Wα at 0.05= 52	W > Wα **Accept** $H_{0:}$ result is not significant

Based on experiment results in Table 1, two sets of Wilcoxon Test are performed. The first one is between BA and t-seq and the other one is between BA and ASP. To demonstrate how Wilcoxon Signed Rank test works, part of results from BACA and ASP (taken from Table 1) are compared as in Table 2 below. In Table 2, column a represents the test configuration number, column b and c lists the results from ASP and BACA respectively. Column d shows the different between b and c and column e list the absolute of column d. In column f and) the absolute values are sorted and assigned with its sign (negative or positive). The absolute values are ranked in Column h and then re-assigned back with its rank in column i.

Results from both ASP and t-seq are only considered if the data is complete. For the unavailable values (i.e. NA results from ASP), the pair is considered incomplete and excluded in this test. The results of the Wilcoxon Test are presented in Table 3 below.

5. Conclusion

In this paper, the performance of BA to generate test data for t-way is evaluated. From the results, in some test configuration, BA is performing well compared to both ASP and t-seq vice versa. Besides comparing the size of the final test suit, a statistical approach using Wilcoxon Signed Rank test is used to see whether the differences in test suit size is significant or not. For BA vs. t-seq, statistical evidence reveals that the results between the two are significance. Since BA produced all positive results compared to t-seq, it can be concluded that in overall BA is significantly better than t-seq in term of performance. On the other hand, with regard to BA vs. ASP, although ASP produces many better results, statistical approach reveal that it is not significant that ASP performs better than BA.

6. Acknowledgements

The work reported in this paper is funded by the generous Fundamental Research Grant Scheme (FRGS 2014-1) from the Ministry of Higher Education, Malaysia and Universiti Tenaga Nasional.

7. References

1. Jenkin B. jenny. 2012. http://burtleburtle.net/bob/math/jenny.html
2. Lei Y, Kacker R, Kuhn DR, Okun V, Lawrence J. IPOG: A general strategy for t-way software testing. 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems ECBS '07; 2007. p. 549–56. Crossref.
3. Younis MI, Zamli KZ. MC-MIPOG: A parallel t-way test generation strategy for multicore systems. Electronics and Telecommunications Research Institute. 2010 Feb; 32(1):73–83.
4. Cohen DM, Dalal SR, Fredman ML, Patton GC. The AETG system: An approach to testing based on combinatorial design. IEEE Transactions on Software Engineering. 1997; 23(7):437–44. Crossref.
5. Williams AW. Determination of test configurations for pair-wise interaction coverage. Proceedings of the IFIP TC6/WG6.1 13th International Conference on Testing Communicating Systems: Tools and Techniques; Deventer, The Netherlands. 2000. p. 59–74. Crossref.
6. Chen X, Gu Q, Li A, Chen D. Variable strength interaction testing with an ant colony system approach. Software Engineering Conference, 2009. APSEC '09. Asia-Pacific; 2009. p. 160–7. Crossref.
7. Czerwonka J. Pairwise Testing in real world. Proceedings of 24th Pacific Northwest Software Quality Conference; 2006.
8. Ahmed BS, Zamli KZ, Lim CP. Constructing a T-way interaction test suite using the particle swarm optimization approach. IJICIC. 2011; 8(1):1–10.
9. Alsewari ARA, Zamli KZ. Design and implementation of a harmony-search-based variable-strength t-way testing strategy with constraints support. Inf Softw Technol. 2012 Jun; 54(6):553–68. Crossref.
10. Ziyuan W, Changhai N, Baowen X. Generating combinatorial test suite for interaction relationship. Fourth International Workshop on Software Quality Assurance:

In conjunction with the 6th ESEC/FSE Joint Meeting; New York, NY, USA. 2007. p. 55–61. Crossref.

11. Wang Z, Xu B, Nie C. Greedy heuristic algorithms to generate variable strength combinatorial test suite. The Eighth International Conference on Quality Software, QSIC '08; 2008. p. 155–60. Crossref.
12. Arshem J. Test vector generator. http://tvg.sourceforge.net/
13. Othman RR, Zamli KZ. ITTDG: Integrated T-way test data generation strategy for interaction testing. Scientific Research and Essays. 2011 Aug; 6(17):3638–48. Crossref.
14. Kuhn DR, Higdon JM, Lawrence JF, Kacker RN, Lei Y. Combinatorial methods for event sequence testing. 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation (ICST); 2012. p. 601–9. Crossref.
15. Esra E, Inoue K, Oetsch J, Puhrer J, Tompits H, Yilmaz C. Answer-set programming as a new approach to event-sequence testing. The Third International Conference on Advances in System Testing and Validation Lifecycle (VALID 2011); Barcelona, Spain. 2011. p. 26–34.
16. Shiba T, Tsuchiya T, Kikuno T. Using artificial life techniques to generate test cases for combinatorial testing. Proceedings of the 28th Annual International Computer Software and Applications Conference - Volume 01; Washington, DC, USA. 2004. p. 72–7.
17. Zabil MHM, Zamli KZ, Othman RR. Sequence-based interaction testing strategy implementation using bees algorithm. The 2012 IEEE Symposium on Computers and Informatics; 2012. p. 81–5. Crossref.
18. Cohen MB, Gibbons PB, Mugridge WB, Colbourn CJ. Constructing test suites for interaction testing. Proceedings of 2003 25th International Conference on Software Engineering; 2003. p. 38–48. Crossref.
19. Cohen DM, Dalal SR, Kajla A, Patton GC. The Automatic Efficient Test Generator (AETG) system. 1994. Proceedings of 5th International Symposium on Software Reliability Engineering; 1994. p. 303–9. Crossref.
20. Cohen DM, Dalal SR, Fredman ML, Patton GC. The AETG system: An approach to testing based on combinatorial design. IEEE Transactions on Software Engineering.1997 Jul; 23(7):437–44. Crossref.
21. Lei Y, Tai KC. In-Parameter-Order: A test generation strategy for pairwise testing. The 3rd IEEE International Symposium on High-Assurance Systems Engineering; Washington, DC, USA. 1998. p. 254–61.
22. Williams AW. Determination of test configurations for pair-wise interaction coverage. Proceedings of the IFIP TC6/WG6.1 13th International Conference on Testing Communicating Systems: Tools and Techniques; Deventer, The Netherlands; 2000. p. 59–74. Crossref.
23. Cohen MB, Colbourn CJ, Ling ACH. Augmenting simulated annealing to build interaction test suites. Proceedings of the 14th International Symposium on Software Reliability Engineering; Denver, CO, USA. 2003. p. 394. Crossref.
24. Shiba T,Tsuchiya T, Kikuno T. Using artificial life techniques to generate test cases for combinatorial testing. Proceedings of the 28th Annual International Computer Software and Applications Conference - Volume 01; Washington, DC, USA. 2004. p. 72–7.
25. Czerwonka J. Pairwise testing in real world. Proceedings of 24th Pacific Northwest Software Quality Conference; 2006.
26. Lei Y, Kacker R, Kuhn DR, Okun V, Lawrence J. IPOG: A general strategy for t-way software testing. 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-based Systems, ECBS '07; 2007. p. 549–56. Crossref.
27. Chen X, Gu Q, Li A, Chen D. Variable strength interaction testing with an ant colony system approach. Software Engineering Conference, APSEC '09. Asia-Pacific; 2009. p. 160–7. Crossref.
28. Younis MI, Zamli KZ. MC-MIPOG: A parallel t-way test generation strategy for multicore systems. Electronics and Telecommunications Research Institute. 2010 Feb; 32(1):73–83.
29. Othman RR, Zamli KZ. ITTDG: Integrated T-way test data generation strategy for interaction testing. Scientific Research and Essays. 2011 Aug; 6(17):3638–48. Crossref.
30. Ahmed BS, Zamli KZ, Lim CP. Constructing a t-way interaction test suite using the particle swarm optimization approach. IJICIC. 2011 Nov; 8(1):1–10.
31. Alsewari ARA, Zamli KZ. Design and implementation of a harmony-search-based variable-strength t-way testing strategy with constraints support. Inf Softw Technol. 2012 Jun; 54(6):553–68. Crossref. https://doi.org/10.1016/j.infsof.2012.01.002

32. Ahmed BS, Abdulsamad TS, Potrus MY. Achievement of minimized combinatorial test suite for configuration-aware software functional testing using the cuckoo search algorithm. Information and Software Technology. 2015 Oct; 66:13–9. Crossref.

33. Kuhn DR, Wallace DR, Gallo JAM. Software fault interactions and implications for software testing. IEEE Transactions on Software Engineering. 2004 Jun; 30(6):418–21. Crossref.

34. Kuhn R, Lei Y, Kacker R. Practical combinatorial testing: beyond pairwise. IT Professional. 2008 Jun; 10(3):19–23. Crossref.

35. Kuhn DR, Higdon JM, Lawrence JF, Kacker RN, Lei Y. Combinatorial methods for event sequence testing. 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation (ICST); 2012. p. 601–9. Crossref.