Generating pairwise combinatorial test set using artificial parameters and values

Conference Paper · September 2008

DOI: 10.1109/ITSIM.2008.4632001 · Source: IEEE Xplore

3 authors:

Mohammed I. Younis
University of Baghdad
48 PUBLICATIONS **351** CITATIONS

Kamal Z Zamli
Universiti Malaysia Pahang
167 PUBLICATIONS **1,250** CITATIONS

Nor Ashidi Mat Isa
Universiti Sains Malaysia
225 PUBLICATIONS **2,398** CITATIONS

Generating Pairwise Combinatorial Test Set Using Artificial Parameters and Values

Mohammed I. Younis	Kamal Z. Zamli	Nor Ashidi Mat Isa
School of Electrical and Electronic Engineering Universiti Sains Malaysia Engineering Campus 14300 Nibong Tebal, Penang, Malaysia	*School of Electrical and Electronic Engineering Universiti Sains Malaysia Engineering Campus 14300 Nibong Tebal, Penang, Malaysia*	*School of Electrical and Electronic Engineering Universiti Sains Malaysia Engineering Campus 14300 Nibong Tebal, Penang, Malaysia*
younismi@gmail.com	*eekamal@eng.usm.my*	*ashidi@eng.usm.my*

Abstract

In order to meet market demands for quality software products, software engineers are increasingly under pressure to test more lines of codes. To maintain acceptable test coverage, software engineers need to consider a significantly large number of test set. Many combinations of possible input parameters, hardware/software environments, and system conditions need to be tested and verified against for conformance. Often, this results into combinatorial explosion problem (i.e. too many test data set too consider). Earlier work suggests that pairwise sampling strategy based on parameter interactions of variables can be effective. This paper discusses an efficient pairwise strategy, termed RA and ORA, that can systematically minimize the pairwise test set generated from higher order test parameters to lower order ones. In doing so, this paper demonstrates and compares the results against existing strategies including IRPS, IPO, GA, ACA, Jenny and All Pairs.

1. Introduction

Nowadays, we are increasingly dependent on software to assist as well as facilitate our daily chores. In fact, whenever possible, most hardware implementation is now being replaced by the software counterpart. From the washing machine controllers, mobile phone applications to the sophisticated airplane control systems, the growing dependent on software can be attributed to a number of factors. Unlike hardware, software does not wear out. Thus, the use of software can also help to control maintenance costs. Additionally, software is also malleable and can be easily changed as the need arises.

Covering as much as 40 to 50 percent of the total software development costs, testing can be considered one of the most important activities for software validation and verification. Lack of testing can lead to disastrous consequences including loss of data, fortunes and even lives. Many combinations of possible input parameters, hardware/software environments, and system conditions need to be tested and verified against for conformance based on the system's specification. Often, this results into combinatorial explosion problem.

Combinatorial explosion problem [4][5][6][7][8] poses one of the biggest challenges in modern computer science due to the fact that it often kills traditional approaches to analysis, verification, monitoring and control. A number of techniques have been explored in the past to address this NP complete problem. Undoubtedly, parallel testing (e.g. [12],[13],[14]) can be employed to reduce the time required for performing the tests. Nevertheless, as software and hardware are getting more complex than ever, parallel testing approach becomes immensely expensive due to the need for faster and higher capability processors along state-of-the-art computer hardware. Apart from parallel testing, random testing could also be another option. However, random testing [e.g. [2]) tends to dwell on unfair distribution of the test set.

Earlier wok suggests that pairwise sampling strategy

978-1-4244-2328-6/08/$25.00 © 2008 IEEE

(i.e. based on two-way parameter interaction) can be effective to systematically reduce the test data set to some manageable combinations (yet with the capability to detect from 60 to 80 percent of the faults

ding and complementing earlier work, the paper proposes two efficient pairwise strategies, called RA and ORA, that can systematically minimize pairwise test set generated from higher order test parameters to lower order ones. Thus, RA and ORA, can be used in conjunction with existing pairwise strategies (e.g. IRPS [6], AETG and its variations [5], IPO [4][8], GA [9], ACA [9] and All Pairs [10]).

This paper is organized as follows. Section 2 discusses some related work. Section 3 introduces the concept of adding the artificial parameters along with a step-by-step example using the proposed strategy. Section 4 highlights comparison of both RA, and ORA with existing strategies. Finally, section 6 gives the conclusion and suggestion for future work.

2. Related Work

As discuss earlier, many different sampling strategies exist in the literature. In general, existing strategies can be categorized into two categories based on the dominant approaches, that is, algebraic approaches or computational approaches.

Algebraic approaches construct test sets using pre-defined rules. Most algebraic approaches compute test sets directly by a mathematical function [4]. Thus, the computations involved in algebraic approaches are typically lightweight, and in some cases, algebraic approaches can produce the most optimal test sets.

In a nut shell, algebraic approaches are often based on the extensions of the mathematical methods for constructing orthogonal arrays (OA) [1], and covering arrays (CA) [15]. Some variations of the algebraic approach also exploit recursion in order to permit the construction of larger test sets from smaller ones.

Unlike algebraic approaches, computational approaches often rely on the generation of the all pair combinations. Based on all pair combinations, the computational approaches iteratively search the combinations space to generate the required test case until all pairs have been covered. Nevertheless, in the case where the number of pairs to be considered is significantly large, adopting computational approaches can be expensive due to the need to consider explicit enumeration from all the combination space. Some of these strategies are AETG (Automatic Efficient Test Generator) [5], GA (Genetic Algorithm) and ACA (Ant Colony Algorithm) [9], Jenny [11], In Parameter Order (IPO) strategy [4] [8], hill climbing and

[4][5]). Here, any two combinations of parameter values are to be covered by at least one test to ensure acceptable test coverage. Buil

simulated annealing (SA) [7], and the Intersection of Residues Pair Set Strategy (IRPS) [6].

3. Illustrative Examples

Based on empirical results, we noted that higher number of parameters and values give better size for the generated set. For example, in IRPS strategy the size for 11 11-valued parameters is 121 while the size for 10 10-valued parameters is 149, and the execution time for the first is less than the later. Our contribution in this paper is to generate 10 10-valued parameters pairwise test set from 11 11-valued parameters pairwise set. Here, we expect that the size of the generated test set be at most 121 or something less than it. As will be shown later, by adding artificial parameters and values, we can obtain smaller test set size. To illustrate this issue, we have considered the IRPS strategy for 11 11-valued parameters. The obtained test set is given in Table 1. Apart from IRPS, it should be noted here that our strategy discussed here could be equally applicable to other strategies as well.

Our first step is to remove the artificial parameter (i.e. the last columns), and replace the artificial variables (i.e. all values that equal to 10) by don't care (-). Using this straightforward optimization , note that T11 contains all don't care, so the resulted test size equal to 120 test cases only (i.e. as given in Table 2). It should be noted that all parameters and variables have balance coverage. We called this RA strategy.

Finally, our strategy (we call it optimized RA or ORA) is to optimize the resulted test sets by removing the test cases that contains don't care values. This is done iteratively one-pair-at-a-time by replacing don't care values with suitable values and removing repetitions if they exists. For example, referring to Table 2, T120 (i.e. -b9c8d7e6f5g4h3i2j1) can be removed. If T120 is removed, then 9*8/2=36 pairs are also removed, namely; (b9, c8), (b9, d7), (b9, e6), (b9, f5), (b9, g4), (b9, h3), (b9, i2), (b9, j1), (c8, d7), (c8, e6), (c8, f5), (c8, g4), (c8, h3), (c8, i2), (c8, j1), (d7, e6), (d7, f5), (d7, g4), (d7, h3), (d7, i2), (d7, j1), (e6, f5), (e6, g4), (e6, h3), (e6, i2), (e6, j1), (f5, g4), (f5, h3), (f5, i2), (f5, j1), (g4, h3), (g4, i2), (g4, j1), (h3, i2), (h3, j1), and (i2, j1). These pairs are distributed to test cases: T19, T69, T49, T39, T99, T29, T89, T79, T26, T17, T91, T73, T82, T44, T64, T77, T34, T15, T101, T58, T66, T45, T31, T63, T13, T38, T62, T97, T103,

978-1-4244-2328-6/08/$25.00 © 2008 IEEE

T51, T67, T41, T25, T86, T105, and T92 respectively.

By the same token, T12 can also be removed (i.e. a1b2c3d4e5f6g7h8i9-) resulting into the removal of the following pairs: (a1, b2), (a1, c3), (a1, d4), (a1, e5), (a1, f6), (a1, g7), (a1, h8), (a1, i9), (b2, c3), (b2, d4), (b2, e5), (b2, f6), (b2, g7), (b2, h8), (b2, i9), (c3, d4), (c3, e5), (c3, f6), (c3,g7), (c3,h8), (c3, i9), (d4, e5), (d4, f6), (d4, g7), (d4, h8), (d4, i9), (e5, f6), (e5, g7), (e5, h8), (e5, i9), (f6, g7), (f6, h8), (f6, i9), (g7, h8), (g7, i9), and (h8, i9). These pairs are distributed to test cases: T100, T111, T109, T56 , T98 , T78, T76 , T54 , T93 , T53 , T113 , T22 , T70 , T80 , T40 , T55 , T106 , T68 , T28 , T46 , T102 , T74 , T104 , T112 , T61 , T117 , T81 , T94 , T42 , T88 , T52 , T114 , T85 , T33 , T57, and T27 respectively. The resulting test set after optimization is given in Table 3.

4. Result and Discussion

As discussed earlier, our RA and ORA strategy starts with existing pairwise test set. This test set could be generated from any of the existing strategies. The advantage using RA and ORA strategy is the fact that the computation will be minimal compared to generating the pairwise set completely from scratch. In this case, lower order pairwise test set can be generated by the addition of the artificial parameters and values from known higher order pairwise test set. Both RA and ORA guaranty the coverage of pairwise interaction with minimal test set. Which save efforts and execution time for the generation of the test case as well as running the test.

Concerning the performance, Table 4 compares our result obtained in both RA, ORA techniques with some available published results and tools. It is clear that both RA, ORA give minimal or near minimal solutions and outperformed IRPS [6], IPO [4][8], Jenny [11], GA and ACA [9], as well as All Pairs [10] in both the size of generated test set and the execution time. Also, ORA outperformed RA in the size of generated test case but required more time of execution due to the optimization process.

5. Conclusion

In this paper we propose two strategies, namely RA, ORA, for building test set from higher order test set that contains more parameters and values. Both ORA and RA are performs well, and give minimal number of the test set than other published result and available tools. As a continuation of this work, we plan to investigate other optimization technique that will give optimal results and run under the grid environment.

6. References

[1] Bush, K.A., Orthogonal Array of Index Unity, Annals of Mathematical Statistics, Vol. 23 (1952), pp. 426-434.

[2] Ammann, P.E. and Offutt, A.J., Using Formal Methods to Derive Test Frames in Category-Partition Testing. *In Proc. of the 9th Annual Conference on Computer Assurance (COMPASS'94)*, IEEE CS Press, June 1994, pp. 69-80.

[3] Kuhn, D.R., Wallace, D.R., Gallo, A.M., Software Fault Interactions and Implications for Software Testing., *IEEE Transaction on Software Engineering*, Vol. 30, No. 6, IEEE Computer Society (June 2004), pp. 418-421.

[4] Lei, Y., Kacker, R., Kuhn, D.R., Okun, V., Lawrence, J., IPOG: A General Strategy for T-Way Software Testing. In *Proc. of the 14th Annual IEEE Intl. Conf. and Workshops on the Engineering of Computer-Based Systems*, Tucson, AZ, IEEE Computer Society Press (March 2007) pp. 549-556.

[5] Cohen, D.M., Dalal, S.R., Fredman, M.L., Patton, G.C., The AETG system: An Approach to Testing Based on Combinatorial Design, *IEEE Transaction on Software Engineering*, 23 (7), (1997), pp. 437-443.

[6] Younis, M.I., Zamli, K.Z., and Isa, N.A.M., "IRPS - An Efficient Test Data Generation Strategy for Pairwise Testing", accepted for publication in KES 2008 (forthcoming).

[7] Cohen, M.B., Designing Test Suites for Software Interaction Testing, PhD Thesis, The University of Auckland (2004).

[8] Lei, Y., Tai, K.C., In-Parameter-Order: A Test Generating Strategy for Pairwise Testing, *IEEE Transaction on Software Engineering*, 28 (1), (2002), pp. 1-3.

[9] Shiba, T., Tsuchiya, T., Kikuno, T.: Using Artificial Life Techniques to Generate Test Cases for Combinatorial Testing. In *Proc. of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04)*, Hong Kong, China, (September 2004), pp.72-77.

[10] URL: http://www.satisfice.com.

[11] URL: http://burtleburtle.net/bob/math/jenny.html.

[12] Zamli, K.Z., Mat Isa, N.A., Klaib, M.F.J., and Azizan, S.N. Designing a Combinatorial Java Unit Testing Tool. In *Proc. of the IASTED Intl. Conference on Advances in Computer Science and Technology (ACST 2007)*, Phuket, Thailand, April 2007.

[13] Zamli, K.Z., Mat Isa, N.A., Klaib, M.F.J, Azizan, S.N., A Tool for Automated Test Data Generation (and Execution) Based on Combinatorial Approach, *International Journal of Software Engineering and Its Applications*, July 2007, pp. 19-34.

[14] Zamli. K.Z. and Mat Isa, N.A., JTst: An Automated Unit Testing Tool for Java Program, *American Journal of Applied Science*, Vol 5, No 2, Sept 2007, pp. 77-82.

[15] Zekaaoui, L., Mixed Covering Arrays on Graphs and Tabu Search Algorithm, MSc Thesis, Ottawa-Carleton

Institute for Computer Science, Univ of Ottawa, September 2006.

978-1-4244-2328-6/08/$25.00 © 2008 IEEE

TABLE 1
THE PAIRWISE GENERATED TEST SET FOR 11 11-VALUED PARAMETERS USING IRPS

Test Name	A	B	C	D	E	F	G	H	I	J	K
T1	a0	b0	c0	d0	e0	f0	g0	h0	i0	j0	k0
T2	a1	b1	c1	d1	e1	f1	g1	h1	i1	j1	k1
T3	a2	b2	c2	d2	e2	f2	g2	h2	i2	j2	k2
T4	a3	b3	c3	d3	e3	f3	g3	h3	i3	j3	k3
T5	a4	b4	c4	d4	e4	f4	g4	h4	i4	j4	k4
T6	a5	b5	c5	d5	e5	f5	g5	h5	i5	j5	k5
T7	a6	b6	c6	d6	e6	f6	g6	h6	i6	j6	k6
T8	a7	b7	c7	d7	e7	f7	g7	h7	i7	j7	k7
T9	a8	b8	c8	d8	e8	f8	g8	h8	i8	j8	k8
T10	a9	b9	c9	d9	e9	f9	g9	h9	i9	j9	k9
T11	a10	b10	c10	d10	e10	f10	g10	h10	i10	j10	k10
T12	a0	b1	c2	d3	e4	f5	g6	h7	i8	j9	k10
T13	a1	b2	c3	d4	e5	f6	g7	h8	i9	j10	k0
T14	a2	b3	c4	d5	e6	f7	g8	h9	i10	j0	k1
T15	a3	b4	c5	d6	e7	f8	g9	h10	i0	j1	k2
T16	a4	b5	c6	d7	e8	f9	g10	h0	i1	j2	k3
T17	a5	b6	c7	d8	e9	f10	g0	h1	i2	j3	k4
T18	a6	b7	c8	d9	e10	f0	g1	h2	i3	j4	k5
T19	a7	b8	c9	d10	e0	f1	g2	h3	i4	j5	k6
T20	a8	b9	c10	d0	e1	f2	g3	h4	i5	j6	k7
T21	a9	b10	c0	d1	e2	f3	g4	h5	i6	j7	k8
T22	a10	b0	c1	d2	e3	f4	g5	h6	i7	j8	k9
T23	a0	b2	c4	d6	e8	f10	g1	h3	i5	j7	k9
T24	a1	b3	c5	d7	e9	f0	g2	h4	i6	j8	k10
T25	a2	b4	c6	d8	e10	f1	g3	h5	i7	j9	k0
T26	a3	b5	c7	d9	e0	f2	g4	h6	i8	j10	k1
T27	a4	b6	c8	d10	e1	f3	g5	h7	i9	j0	k2
T28	a5	b7	c9	d0	e2	f4	g6	h8	i10	j1	k3
T29	a6	b8	c10	d1	e3	f5	g7	h9	i0	j2	k4
T30	a7	b9	c0	d2	e4	f6	g8	h10	i1	j3	k5
T31	a8	b10	c1	d3	e5	f7	g9	h0	i2	j4	k6
T32	a9	b0	c2	d4	e6	f8	g10	h1	i3	j5	k7
T33	a10	b1	c3	d5	e7	f9	g0	h2	i4	j6	k8
T34	a0	b3	c6	d9	e1	f4	g7	h10	i2	j5	k8
T35	a1	b4	c7	d10	e2	f5	g8	h0	i3	j6	k9
T36	a2	b5	c8	d0	e3	f6	g9	h1	i4	j7	k10
T37	a3	b6	c9	d1	e4	f7	g10	h2	i5	j8	k0
T38	a4	b7	c10	d2	e5	f8	g0	h3	i6	j9	k1
T39	a5	b8	c0	d3	e6	f9	g1	h4	i7	j10	k2
T40	a6	b9	c1	d4	e7	f10	g2	h5	i8	j0	k3
T41	a7	b10	c2	d5	e8	f0	g3	h6	i9	j1	k4
T42	a8	b0	c3	d6	e9	f1	g4	h7	i10	j2	k5
T43	a9	b1	c4	d7	e10	f2	g5	h8	i0	j3	k6
T44	a10	b2	c5	d8	e0	f3	g6	h9	i1	j4	k7
T45	a0	b4	c8	d1	e5	f9	g2	h6	i10	j3	k7
T46	a1	b5	c9	d2	e6	f10	g3	h7	i0	j4	k8
T47	a2	b6	c10	d3	e7	f0	g4	h8	i1	j5	k9
T48	a3	b7	c0	d4	e8	f1	g5	h9	i2	j6	k10
T49	a4	b8	c1	d5	e9	f2	g6	h10	i3	j7	k0
T50	a5	b9	c2	d6	e10	f3	g7	h0	i4	j8	k1
T51	a6	b10	c3	d7	e0	f4	g8	h1	i5	j9	k2
T52	a7	b0	c4	d8	e1	f5	g9	h2	i6	j10	k3
T53	a8	b1	c5	d9	e2	f6	g10	h3	i7	j0	k4
T54	a9	b2	c6	d10	e3	f7	g0	h4	i8	j1	k5
T55	a10	b3	c7	d0	e4	f8	g1	h5	i9	j2	k6
T56	a0	b5	c10	d4	e9	f3	g8	h2	i7	j1	k6
T57	a1	b6	c0	d5	e10	f4	g9	h3	i8	j2	k7
T58	a2	b7	c1	d6	e0	f5	g10	h4	i9	j3	k8
T59	a3	b8	c2	d7	e1	f6	g0	h5	i10	j4	k9
T60	a4	b9	c3	d8	e2	f7	g1	h6	i0	j5	k10
T61	a5	b10	c4	d9	e3	f8	g2	h7	i1	j6	k0
T62	a6	b0	c5	d10	e4	f9	g3	h8	i2	j7	k1
T63	a7	b1	c6	d0	e5	f10	g4	h9	i3	j8	k2
T64	a8	b2	c7	d1	e6	f0	g5	h10	i4	j9	k3
T65	a9	b3	c8	d2	e7	f1	g6	h0	i5	j10	k4
T66	a10	b4	c9	d3	e8	f2	g7	h1	i6	j0	k5
T67	a0	b6	c1	d7	e2	f8	g3	h9	i4	j10	k5
T68	a1	b7	c2	d8	e3	f9	g4	h10	i5	j0	k6
T69	a2	b8	c3	d9	e4	f10	g5	h0	i6	j1	k7
T70	a3	b9	c4	d10	e5	f0	g6	h1	i7	j2	k8
T71	a4	b10	c5	d0	e6	f1	g7	h2	i8	j3	k9
T72	a5	b0	c6	d1	e7	f2	g8	h3	i9	j4	k10
T73	a6	b1	c7	d2	e8	f3	g9	h4	i10	j5	k0
T74	a7	b2	c8	d3	e9	f4	g10	h5	i0	j6	k1
T75	a8	b3	c9	d4	e10	f5	g0	h6	i1	j7	k2
T76	a9	b4	c10	d5	e0	f6	g1	h7	i2	j8	k3
T77	a10	b5	c0	d6	e1	f7	g2	h8	i3	j9	k4
T78	a0	b7	c3	d10	e6	f2	g9	h5	i1	j8	k4
T79	a1	b8	c4	d0	e7	f3	g10	h6	i2	j9	k5
T80	a2	b9	c5	d1	e8	f4	g0	h7	i3	j10	k6
T81	a3	b10	c6	d2	e9	f5	g1	h8	i4	j0	k7
T82	a4	b0	c7	d3	e10	f6	g2	h9	i5	j1	k8
T83	a5	b1	c8	d4	e0	f7	g3	h10	i6	j2	k9
T84	a6	b2	c9	d5	e1	f8	g4	h0	i7	j3	k10
T85	a7	b3	c10	d6	e2	f9	g5	h1	i8	j4	k0
T86	a8	b4	c0	d7	e3	f10	g6	h2	i9	j5	k1
T87	a9	b5	c1	d8	e4	f0	g7	h3	i10	j6	k2
T88	a10	b6	c2	d9	e5	f1	g8	h4	i0	j7	k3
T89	a0	b8	c5	d2	e10	f7	g4	h1	i9	j6	k3
T90	a1	b9	c6	d3	e0	f8	g5	h2	i10	j7	k4
T91	a2	b10	c7	d4	e1	f9	g6	h3	i0	j8	k5
T92	a3	b0	c8	d5	e2	f10	g7	h4	i1	j9	k6
T93	a4	b1	c9	d6	e3	f0	g8	h5	i2	j10	k7
T94	a5	b2	c10	d7	e4	f1	g9	h6	i3	j0	k8
T95	a6	b3	c0	d8	e5	f2	g10	h7	i4	j1	k9
T96	a7	b4	c1	d9	e6	f3	g0	h8	i5	j2	k10
T97	a8	b5	c2	d10	e7	f4	g1	h9	i6	j3	k0
T98	a9	b6	c3	d0	e8	f5	g2	h10	i7	j4	k1
T99	a10	b7	c4	d1	e9	f6	g3	h0	i8	j5	k2
T100	a0	b9	c7	d5	e3	f1	g10	h8	i6	j4	k2
T101	a1	b10	c8	d6	e4	f2	g0	h9	i7	j5	k3
T102	a2	b0	c9	d7	e5	f3	g1	h10	i8	j6	k4
T103	a3	b1	c10	d8	e6	f4	g2	h0	i9	j7	k5

T104	a4	b2	c0	d9	e7	f5	g3	h1	i10	j8	k6
T105	a5	b3	c1	d10	e8	f6	g4	h2	i0	j9	k7
T106	a6	b4	c2	d0	e9	f7	g5	h3	i1	j10	k8
T107	a7	b5	c3	d1	e10	f8	g6	h4	i2	j0	k9
T108	a8	b6	c4	d2	e0	f9	g7	h5	i3	j1	k10
T109	a9	b7	c5	d3	e1	f10	g8	h6	i4	j2	k0
T110	a10	b8	c6	d4	e2	f0	g9	h7	i5	j3	k1
T111	a0	b10	c9	d8	e7	f6	g5	h4	i3	j2	k1
T112	a1	b0	c10	d9	e8	f7	g6	h5	i4	j3	k2
T113	a2	b1	c0	d10	e9	f8	g7	h6	i5	j4	k3
T114	a3	b2	c1	d0	e10	f9	g8	h7	i6	j5	k4
T115	a4	b3	c2	d1	e0	f10	g9	h8	i7	j6	k5
T116	a5	b4	c3	d2	e1	f0	g10	h9	i8	j7	k6
T117	a6	b5	c4	d3	e2	f1	g0	h10	i9	j8	k7
T118	a7	b6	c5	d4	e3	f2	g1	h0	i10	j9	k8
T119	a8	b7	c6	d5	e4	f3	g2	h1	i0	j10	k9
T120	a9	b8	c7	d6	e5	f4	g3	h2	i1	j0	k10
T121	a10	b9	c8	d7	e6	f5	g4	h3	i2	j1	k0

TABLE 2

THE RESULTING TEST SET FOR 10 10-VALUED PARAMETERS BY RA ALGORITHM

Test Name	A	B	C	D	E	F	G	H	I	J
T1	a0	b0	c0	d0	e0	f0	g0	h0	i0	j0
T2	a1	b1	c1	d1	e1	f1	g1	h1	i1	j1
T3	a2	b2	c2	d2	e2	f2	g2	h2	i2	j2
T4	a3	b3	c3	d3	e3	f3	g3	h3	i3	j3
T5	a4	b4	c4	d4	e4	f4	g4	h4	i4	j4
T6	a5	b5	c5	d5	e5	f5	g5	h5	i5	j5
T7	a6	b6	c6	d6	e6	f6	g6	h6	i6	j6
T8	a7	b7	c7	d7	e7	f7	g7	h7	i7	j7
T9	a8	b8	c8	d8	e8	f8	g8	h8	i8	j8
T10	a9	b9	c9	d9	e9	f9	g9	h9	i9	j9
Removed	-	-	-	-	-	-	-	-	-	-
T11	a0	b1	c2	d3	e4	f5	g6	h7	i8	j9
T12	a1	b2	c3	d4	e5	f6	g7	h8	i9	-
T13	a2	b3	c4	d5	e6	f7	g8	h9	-	j0
T14	a3	b4	c5	d6	e7	f8	g9	-	i0	j1
T15	a4	b5	c6	d7	e8	f9	-	h0	i1	j2
T16	a5	b6	c7	d8	e9	-	g0	h1	i2	j3
T17	a6	b7	c8	d9	-	f0	g1	h2	i3	j4
T18	a7	b8	c9	-	e0	f1	g2	h3	i4	j5
T19	a8	b9	-	d0	e1	f2	g3	h4	i5	j6
T20	a9	-	c0	d1	e2	f3	g4	h5	i6	j7
T21	-	b0	c1	d2	e3	f4	g5	h6	i7	j8
T22	a0	b2	c4	d6	e8	-	g1	h3	i5	j7
T23	a1	b3	c5	d7	e9	f0	g2	h4	i6	j8
T24	a2	b4	c6	d8	-	f1	g3	h5	i7	j9
T25	a3	b5	c7	d9	e0	f2	g4	h6	i8	-
T26	a4	b6	c8	-	e1	f3	g5	h7	i9	j0
T27	a5	b7	c9	d0	e2	f4	g6	h8	-	j1
T28	a6	b8	-	d1	e3	f5	g7	h9	i0	j2
T29	a7	b9	c0	d2	e4	f6	g8	-	i1	j3
T30	a8	-	c1	d3	e5	f7	g9	h0	i2	j4
T31	a9	b0	c2	d4	e6	f8	-	h1	i3	j5
T32	-	b1	c3	d5	e7	f9	g0	h2	i4	j6
T33	a0	b3	c6	d9	e1	f4	g7	-	i2	j5
T34	a1	b4	c7	-	e2	f5	g8	h0	i3	j6
T35	a2	b5	c8	d0	e3	f6	g9	h1	i4	j7
T36	a3	b6	c9	d1	e4	f7	-	h2	i5	j8
T37	a4	b7	-	d2	e5	f8	g0	h3	i6	j9
T38	a5	b8	c0	d3	e6	f9	g1	h4	i7	-
T39	a6	b9	c1	d4	e7	-	g2	h5	i8	j0
T40	a7	-	c2	d5	e8	f0	g3	h6	i9	j1
T41	a8	b0	c3	d6	e9	f1	g4	h7	-	j2
T42	a9	b1	c4	d7	-	f2	g5	h8	i0	j3
T43	-	b2	c5	d8	e0	f3	g6	h9	i1	j4
T44	a0	b4	c8	d1	e5	f9	g2	h6	-	j3
T45	a1	b5	c9	d2	e6	-	g3	h7	i0	j4
T46	a2	b6	-	d3	e7	f0	g4	h8	i1	j5
T47	a3	b7	c0	d4	e8	f1	g5	h9	i2	j6
T48	a4	b8	c1	d5	e9	f2	g6	-	i3	j7
T49	a5	b9	c2	d6	-	f3	g7	h0	i4	j8
T50	a6	-	c3	d7	e0	f4	g8	h1	i5	j9
T51	a7	b0	c4	d8	e1	f5	g9	h2	i6	-
T52	a8	b1	c5	d9	e2	f6	-	h3	i7	j0
T53	a9	b2	c6	-	e3	f7	g0	h4	i8	j1
T54	-	b3	c7	d0	e4	f8	g1	h5	i9	j2
T55	a0	b5	-	d4	e9	f3	g8	h2	i7	j1
T56	a1	b6	c0	d5	-	f4	g9	h3	i8	j2
T57	a2	b7	c1	d6	e0	f5	-	h4	i9	j3
T58	a3	b8	c2	d7	e1	f6	g0	h5	-	j4
T59	a4	b9	c3	d8	e2	f7	g1	h6	i0	j5
T60	a5	-	c4	d9	e3	f8	g2	h7	i1	j6
T61	a6	b0	c5	-	e4	f9	g3	h8	i2	j7
T62	a7	b1	c6	d0	e5	-	g4	h9	i3	j8
T63	a8	b2	c7	d1	e6	f0	g5	-	i4	j9
T64	a9	b3	c8	d2	e7	f1	g6	h0	i5	-
T65	-	b4	c9	d3	e8	f2	g7	h1	i6	j0
T66	a0	b6	c1	d7	e2	f8	g3	h9	i4	-
T67	a1	b7	c2	d8	e3	f9	g4	-	i5	j0
T68	a2	b8	c3	d9	e4	-	g5	h0	i6	j1
T69	a3	b9	c4	-	e5	f0	g6	h1	i7	j2
T70	a4	-	c5	d0	e6	f1	g7	h2	i8	j3
T71	a5	b0	c6	d1	e7	f2	g8	h3	i9	j4
T72	a6	b1	c7	d2	e8	f3	g9	h4	-	j5
T73	a7	b2	c8	d3	e9	f4	-	h5	i0	j6
T74	a8	b3	c9	d4	-	f5	g0	h6	i1	j7
T75	a9	b4	-	d5	e0	f6	g1	h7	i2	j8
T76	-	b5	c0	d6	e1	f7	g2	h8	i3	j9
T77	a0	b7	c3	-	e6	f2	g9	h5	i1	j8
T78	a1	b8	c4	d0	e7	f3	-	h6	i2	j9
T79	a2	b9	c5	d1	e8	f4	g0	h7	i3	-
T80	a3	-	c6	d2	e9	f5	g1	h8	i4	j0
T81	a4	b0	c7	d3	-	f6	g2	h9	i5	j1
T82	a5	b1	c8	d4	e0	f7	g3	-	i6	j2
T83	a6	b2	c9	d5	e1	f8	g4	h0	i7	j3

T84	a7	b3	-	d6	e2	f9	g5	h1	i8	j4
T85	a8	b4	c0	d7	e3	-	g6	h2	i9	j5
T86	a9	b5	c1	d8	e4	f0	g7	h3	-	j6
T87	-	b6	c2	d9	e5	f1	g8	h4	i0	j7
T88	a0	b8	c5	d2	-	f7	g4	h1	i9	j6
T89	a1	b9	c6	d3	e0	f8	g5	h2	-	j7
T90	a2	-	c7	d4	e1	f9	g6	h3	i0	j8
T91	a3	b0	c8	d5	e2	-	g7	h4	i1	j9
T92	a4	b1	c9	d6	e3	f0	g8	h5	i2	-
T93	a5	b2	-	d7	e4	f1	g9	h6	i3	j0
T94	a6	b3	c0	d8	e5	f2	-	h7	i4	j1
T95	a7	b4	c1	d9	e6	f3	g0	h8	i5	j2
T96	a8	b5	c2	-	e7	f4	g1	h9	i6	j3
T97	a9	b6	c3	d0	e8	f5	g2	-	i7	j4
T98	-	b7	c4	d1	e9	f6	g3	h0	i8	j5
T99	a0	b9	c7	d5	e3	f1	-	h8	i6	j4
T100	a1	-	c8	d6	e4	f2	g0	h9	i7	j5
T101	a2	b0	c9	d7	e5	f3	g1	-	i8	j6
T102	a3	b1	-	d8	e6	f4	g2	h0	i9	j7
T103	a4	b2	c0	d9	e7	f5	g3	h1	-	j8
T104	a5	b3	c1	-	e8	f6	g4	h2	i0	j9
T105	a6	b4	c2	d0	e9	f7	g5	h3	i1	-
T106	a7	b5	c3	d1	-	f8	g6	h4	i2	j0
T107	a8	b6	c4	d2	e0	f9	g7	h5	i3	j1
T108	a9	b7	c5	d3	e1	-	g8	h6	i4	j2
T109	-	b8	c6	d4	e2	f0	g9	h7	i5	j3
T110	a0	-	c9	d8	e7	f6	g5	h4	i3	j2
T111	a1	b0	-	d9	e8	f7	g6	h5	i4	j3
T112	a2	b1	c0	-	e9	f8	g7	h6	i5	j4
T113	a3	b2	c1	d0	-	f9	g8	h7	i6	j5
T114	a4	b3	c2	d1	e0	-	g9	h8	i7	j6
T115	a5	b4	c3	d2	e1	f0	-	h9	i8	j7
T116	a6	b5	c4	d3	e2	f1	g0	-	i9	j8
T117	a7	b6	c5	d4	e3	f2	g1	h0	-	j9
T118	a8	b7	c6	d5	e4	f3	g2	h1	i0	-
T119	a9	b8	c7	d6	e5	f4	g3	h2	i1	j0
T120	-	b9	c8	d7	e6	f5	g4	h3	i2	j1

TABLE 3
THE RESULTING PAIRWISE TEST SET FOR 10 10-VALUED PARAMETERS BY ORA ALGORITHM

Test Name	A	B	C	D	E	F	G	H	I	J
T1	a0	b0	c0	d0	e0	f0	g0	h0	i0	j0
T2	a1	b1	c1	d1	e1	f1	g1	h1	i1	j1
T3	a2	b2	c2	d2	e2	f2	g2	h2	i2	j2
T4	a3	b3	c3	d3	e3	f3	g3	h3	i3	j3
T5	a4	b4	c4	d4	e4	f4	g4	h4	i4	j4
T6	a5	b5	c5	d5	e5	f5	g5	h5	i5	j5
T7	a6	b6	c6	d6	e6	f6	g6	h6	i6	j6
T8	a7	b7	c7	d7	e7	f7	g7	h7	i7	j7
T9	a8	b8	c8	d8	e8	f8	g8	h8	i8	j8
T10	a9	b9	c9	d9	e9	f9	g9	h9	i9	j9
T11	a0	b1	c2	d3	e4	f5	g6	h7	i8	j9
T12	a2	b3	c4	d5	e6	f7	g8	h9	i2	j0
T13	a3	b4	c5	d6	e7	f8	g9	-	i0	j1
T14	a4	b5	c6	d7	e8	f9	g4	h0	i1	j2
T15	a5	b6	c7	d8	e9	-	g0	h1	i2	j3
T16	a6	b7	c8	d9	e6	f0	g1	h2	i3	j4
T17	a7	b8	c9	-	e0	f1	g2	h3	i4	j5
T18	a8	b9	c8	d0	e1	f2	g3	h4	i5	j6
T19	a9	-	c0	d1	e2	f3	g4	h5	i6	j7
T20	-	b0	c1	d2	e3	f4	g5	h6	i7	j8
T21	a0	b2	c4	d6	e8	f6	g1	h3	i5	j7
T22	a1	b3	c5	d7	e9	f0	g2	h4	i6	j8
T23	a2	b4	c6	d8	-	f1	g3	h5	i7	j9
T24	a3	b5	c7	d9	e0	f2	g4	h6	i8	j1
T25	a4	b6	c8	d7	e1	f3	g5	h7	i9	j0
T26	a5	b7	c9	d0	e2	f4	g6	h8	i9	j1
T27	a6	b8	c3	d1	e3	f5	g7	h9	i0	j2
T28	a7	b9	c0	d2	e4	f6	g8	h3	i1	j3
T29	a8	-	c1	d3	e5	f7	g9	h0	i2	j4
T30	a9	b0	c2	d4	e6	f8	g4	h1	i3	j5
T31	-	b1	c3	d5	e7	f9	g0	h2	i4	j6
T32	a0	b3	c6	d9	e1	f4	g7	h8	i2	j5
T33	a1	b4	c7	d7	e2	f5	g8	h0	i3	j6
T34	a2	b5	c8	d0	e3	f6	g9	h1	i4	j7
T35	a3	b6	c9	d1	e4	f7	-	h2	i5	j8
T36	a4	b7	-	d2	e5	f8	g0	h3	i6	j9
T37	a5	b8	c0	d3	e6	f9	g1	h4	i7	j1
T38	a6	b9	c1	d4	e7	f5	g2	h5	i8	j0
T39	a7	b2	c2	d5	e8	f0	g3	h6	i9	j1
T40	a8	b0	c3	d6	e9	f1	g4	h7	i2	j2
T41	a9	b1	c4	d7	e5	f2	g5	h8	i0	j3
T42	-	b2	c5	d8	e0	f3	g6	h9	i1	j4
T43	a0	b4	c8	d1	e5	f9	g2	h6	i2	j3
T44	a1	b5	c9	d2	e6	f5	g3	h7	i0	j4
T45	a2	b6	c3	d3	e7	f0	g4	h8	i1	j5
T46	a3	b7	c0	d4	e8	f1	g5	h9	i2	j6
T47	a4	b8	c1	d5	e9	f2	g6	-	i3	j7
T48	a5	b9	c2	d6	e6	f3	g7	h0	i4	j8
T49	a6	-	c3	d7	e0	f4	g8	h1	i5	j9
T50	a7	b0	c4	d8	e1	f5	g9	h2	i6	j1
T51	a8	b1	c5	d9	e2	f6	g7	h3	i7	j0
T52	a9	b2	c6	d4	e3	f7	g0	h4	i8	j1
T53	a1	b3	c7	d0	e4	f8	g1	h5	i9	j2
T54	a0	b5	c3	d4	e9	f3	g8	h2	i7	j1
T55	a1	b6	c0	d5	e5	f4	g9	h3	i8	j2
T56	a2	b7	c1	d6	e0	f5	g7	h4	i9	j3
T57	a3	b8	c2	d7	e1	f6	g0	h5	i2	j4
T58	a4	b9	c3	d8	e2	f7	g1	h6	i0	j5
T59	a5	-	c4	d9	e3	f8	g2	h7	i1	j6
T60	a6	b0	c5	d4	e4	f9	g3	h8	i2	j7
T61	a7	b1	c6	d0	e5	f5	g4	h9	i3	j8
T62	a8	b2	c7	d1	e6	f0	g5	h3	i4	j9
T63	a9	b3	c8	d2	e7	f1	g6	h0	i5	j1
T64	-	b4	c9	d3	e8	f2	g7	h1	i6	j0
T65	a0	b6	c1	d7	e2	f8	g3	h9	i4	j1

978-1-4244-2328-6/08/$25.00 © 2008 IEEE

T66	a1	b7	c2	d8	e3	f9	g4	h3	i5	j0
T67	a2	b8	c3	d9	e4	f6	g5	h0	i6	j1
T68	a3	b9	c4	d7	e5	f0	g6	h1	i7	j2
T69	a4	b2	c5	d0	e6	f1	g7	h2	i8	j3
T70	a5	b0	c6	d1	e7	f2	g8	h3	i9	j4
T71	a6	b1	c7	d2	e8	f3	g9	h4	-	j5
T72	a7	b2	c8	d3	e9	f4	g4	h5	i0	j6
T73	a8	b3	c9	d4	e5	f5	g0	h6	i1	j7
T74	a9	b4	-	d5	e0	f6	g1	h7	i2	j8
T75	a1	b5	c0	d6	e1	f7	g2	h8	i3	j9
T76	a0	b7	c3	d7	e6	f2	g9	h5	i1	j8
T77	a1	b8	c4	d0	e7	f3	g7	h6	i2	j9
T78	a2	b9	c5	d1	e8	f4	g0	h7	i3	j1
T79	a3	b2	c6	d2	e9	f5	g1	h8	i4	j0
T80	a4	b0	c7	d3	e5	f6	g2	h9	i5	j1
T81	a5	b1	c8	d4	e0	f7	g3	h3	i6	j2
T82	a6	b2	c9	d5	e1	f8	g4	h0	i7	j3
T83	a7	b3	-	d6	e2	f9	g5	h1	i8	j4
T84	a8	b4	c0	d7	e3	f6	g6	h2	i9	j5
T85	a9	b5	c1	d8	e4	f0	g7	h3	i2	j6
T86	-	b6	c2	d9	e5	f1	g8	h4	i0	j7
T87	a0	b8	c5	d2	e5	f7	g4	h1	i9	j6
T88	a1	b9	c6	d3	e0	f8	g5	h2	i2	j7
T89	a2	-	c7	d4	e1	f9	g6	h3	i0	j8
T90	a3	b0	c8	d5	e2	f5	g7	h4	i1	j9
T91	a4	b1	c9	d6	e3	f0	g8	h5	i2	j1
T92	a5	b2	c3	d7	e4	f1	g9	h6	i3	j0
T93	a6	b3	c0	d8	e5	f2	g7	h7	i4	j1
T94	a7	b4	c1	d9	e6	f3	g0	h8	i5	j2
T95	a8	b5	c2	-	e7	f4	g1	h9	i6	j3
T96	a9	b6	c3	d0	e8	f5	g2	h3	i7	j4
T97	a1	b7	c4	d1	e9	f6	g3	h0	i8	j5
T98	a0	b9	c7	d5	e3	f1	g4	h8	i6	j4
T99	a1	b2	c8	d6	e4	f2	g0	h9	i7	j5
T100	a2	b0	c9	d7	e5	f3	g1	h3	i8	j6
T101	a3	b1	c3	d8	e6	f4	g2	h0	i9	j7
T102	a4	b2	c0	d9	e7	f5	g3	h1	i2	j8
T103	a5	b3	c1	d4	e8	f6	g4	h2	i0	j9
T104	a6	b4	c2	d0	e9	f7	g5	h3	i1	j1
T105	a7	b5	c3	d1	e5	f8	g6	h4	i2	j0
T106	a8	b6	c4	d2	e0	f9	g7	h5	i3	j1
T107	a9	b7	c5	d3	e1	-	g8	h6	i4	j2
T108	a1	b8	c6	d4	e2	f0	g9	h7	i5	j3
T109	a0	-	c9	d8	e7	f6	g5	h4	i3	j2
T110	a1	b0	c3	d9	e8	f7	g6	h5	i4	j3
T111	a2	b1	c0	d4	e9	f8	g7	h6	i5	j4
T112	a3	b2	c1	d0	e5	f9	g8	h7	i6	j5
T113	a4	b3	c2	d1	e0	f6	g9	h8	i7	j6
T114	a5	b4	c3	d2	e1	f0	-	h9	i8	j7
T115	a6	b5	c4	d3	e2	f1	g0	-	i9	j8
T116	a7	b6	c5	d4	e3	f2	g1	h0	i9	j9
T117	a8	b7	c6	d5	e4	f3	g2	h1	i0	-
T118	a9	b8	c7	d6	e5	f4	g3	h2	i1	j0

TABLE 4

COMPARISON OF TEST SIZE AND EXCUTION TIME BY DIFFERENT STRATEGIES

10 10-valued parameters	IRPS	IPO	Jenny	GA	ACA	ALL Pairs	RA	ORA
Test Set Size	149	169	157	157	159	177	120	118
Execution Time (second)	16.35	0.3	0.3	866	1180	5.03	<0.01	0.1

978-1-4244-2328-6/08/$25.00 © 2008 IEEE