University of Baghdad
College of Engineering
JOURNAL OF ENGINEERING

Journal of Engineering
journal homepage: www.joe.uobaghdad.edu.iq
Number 11 Volume 25 November 2019

Electrical, Electronics and communications, and Computer Engineering

MVSCA: Multi-Valued Sequence Covering Array

Mohammed Issam Younis *
Assist. Prof. (Ph.D.)
Computer Engineering Department, College of Engineering, University of Baghdad
Baghdad, Iraq
E-mail: younismi@coeng.uobaghdad.edu.iq

ABSTRACT

This paper discusses the limitation of both Sequence Covering Array (SCA) and Covering Array (CA) for testing reactive system when the order of parameter-values is sensitive. In doing so, this paper proposes a new model to take the sequence values into consideration. Accordingly, by superimposing the CA onto SCA yields another type of combinatorial test suite termed Multi-Valued Sequence Covering Array (MVSCA) in a more generalized form. This superimposing is a challenging process due to NP-Hardness for both SCA and CA. Motivated by such a challenge, this paper presents the MVSCA with a working illustrative example to show the similarities and differences among combinatorial testing methods. Consequently, the MVSCA is a new trend that can be a research vehicle for researchers to develop new and/or modify existing combinatorial strategies to deal with the combinatorial explosion problem raised by the MVSCA.

Keywords: combinatorial testing, covering array, event testing, sequence covering array.

مصفوفة التغطية المتسلسلة متعددة القيم

محمد عصام يونس *
استاذ مساعد (دكتوراه)
قسم هندسة الحاسبات , كلية الهندسة , جامعة بغداد, بغداد, العراق

الخلاصة

يناقش هذا البحث قيود كل من مصفوفة التغطية المتسلسلة و مصفوفة التغطية لاختبار النظام التفاعلي عندما يكون ترتيب قيم المعامل حساسًا. عند القيام بذلك ، تقترح هذه الورقة نموذجًا جديدًا لأخذ قيم التسلسل في الحسبان. وفقًا لذلك، من خلال تركيب مصفوفة التغطية على مصفوفة التغطية المتسلسلة، ينتج عنه نوع آخر من مجموعة الاختبارات الاندماجية التي تسمى مصفوفة تغطية المتسلسلة متعددة القيم (MVSCA) بشكل أكثر تعميماً. هذا التراكب هو عملية صعبة بسبب اندماج المصفوفتين المركبتين

*Corresponding author
Peer review under the responsibility of University of Baghdad.
https://doi.org/10.31026/j.eng.2019.11.07

Article received: 2/6/2019
Article accepted: 10/7/2019
Article published: 1/11/2019

مع بعضهما. بدافع من هذا التحدي ، تقدم هذه الورقة التمثيل الرياضي ل MVSCA مع مثال توضيحي لإظهار أوجه التشابه والاختلاف بين أساليب الاختبار الاندماجي. وبالتالي، فإن التعامل مع ال MVSCA المقترحة هو اتجاه جديد يمكن أن يكون وسيلة بحثية للباحثين لتطوير استراتيجيات جديدة و/ أو تعديل الاستراتيجيات الاندماجية الحالية للتعامل مع مشكلة الانفجار الاندماجي التي أثارتها MVSCA المقترحة.

الكلمات الرئيسية: الفحص الاندماجي، مصفوفة التغطية، فحص الحدث، مصفوفة التغطية المتسلسلة.

1. INTRODUCTION

Exhaustive system testing is a beneficial activity for quality and reliability enhancements; however, due to resources and timing constraints, exhaustive system testing is prohibitively impossible **, Zamli and Younis, 2010**. Earlier works adopted combinatorial testing as an efficient black-box technique for system testing for any system under test (SUT) that reduces the size of an exhaustive test suite significantly in a systematic manner. The generation of test suite involves covering the interaction elements (for parameter values) of a certain degree of parameter interaction strength (t). Currently, there are two types of combinatorial interaction testing, namely: covering array (CA) and sequence covering array (SCA) **,Nasser, et al., 2018**.

With the advent of the Internet of Things (IoT), enormous sensors and actuators produce many dynamic events. These events may have multiple values. For instance, a temperature sensor may read a temperature less than or equal to a threshold value, or greater than a threshold value. Similar notes could be observed for other sensors and actuators **,Younis and Hussein, 2018**. In addition, the events with their corresponding values can be entered asynchronously out of order (i.e., the parameter-values are non-adjacent) which fits the demands of the derivation test beds and quality assurance test suites for IoT systems. This raises the need for a new combinatorial testing array that takes into consideration the sequences and their parameter-values, we call it a Multi-Valued Sequence Covering Array (MVSCA).

This paper illustrates the similarities and differences between CA, SCA, and the proposed MVSCA. The remaining sections of this paper are organized as follows. Section 2 gives a mathematical presentation for combinatorial testing. Section 3, gives a step-by-step-example and further discussion regarding non-uniform parameters' values. Section 4 summarizes the trends in related works. Finally, Section 5 states the conclusion and gives future trends in combinatorial test data generation.

2. MATHEMATICAL NOTATIONS

Mathematically, the CA takes parameters of N, t, p, and v witch corresponding to the size, strength of coverage, number of parameters, and uniform-value respectively **,Cohen, 2004; Lei, et al., 2008; Younis and Zamli, 2011**. The number of parameter interaction is determined by Eq. (1). The number of t-way interaction elements (tuples) per parameter interaction is determined by Eq. (2). Thus, the total number of tuples is determined by Eq. (3). The CA is an N*p matrix in which each t-way combination tuples occurs at least one row. It should be mentioned that the parameters are in order (i.e., adjacent).

$$\binom{p}{t} \stackrel{\text{def}}{=} \frac{p!}{t!\,(p-t)!} \tag{1}$$

$$v^t \tag{2}$$

$$\frac{p!}{t!\,(p-t)!}\,v^t \tag{3}$$

The SCA takes the parameters of N, t, and p **,Kuhn, et al., 2012; Chee, et al., 2013.** Similar to CA the number of parameter interaction is determined by Eq. (1). However, the SCA takes the permutation of parameters (sequences), thus the number of tuples per parameter interaction is given in Eq. (4). Thus, the total number of tuples is determined by Eq. (5). The SCA is an N*p matrix in which each t-way permutation tuples occurs at least one row; the t symbols in the permutation may be out of order (i.e., are not required to be adjacent).

$$t! \tag{4}$$

$$\frac{p!}{t!\,(p-t)!}\,t! \tag{5}$$

The MVCA like CA, MVSCA takes parameters of N, t, p, and v. Similar to both CA and SCA the number of parameter interaction is determined by Eq. (1). Unlike both CA and SCA, MVSCA takes every interaction elements and their permutation to generate t-way tuples. Hence, the number of tuples per parameter interaction is given in Eq. (6). Thus, the total number of tuples is determined by Eq. (7). The MVSCA is an N*p matrix in which each t-way permutation tuples occurs at least one row; the t-way permutation of the interaction elements may be out of order (i.e., are not required to be adjacent).

$$t!\ v^{\,t} \tag{6}$$

$$\frac{p!}{t!\,(p-t)!}\ t!\ v^{t} \tag{7}$$

Equations 2,4, and 6 represent the lower bound for CA, SCA, and MVCA respectively. The lower bound of the test suite is the minimum number of test cases in the test suite. Whilst, equations 3,5, and 7 represent the trivial upper bound for CA, SCA, and MVCA respectively.
It should be mentioned that earlier works on both CAs and SCAs demonstrated that the upper bound can be minimized logarithmically as the number of parameters increased **,Cohen, 2004; Lei, et al., 2008; Younis and Zamli, 2011; Kuhn, et al., 2012; Chee, et al., 2013.**

3. ILLUSTRATIVE EXAMPLE

Assume the SUT consists of three parameters (ABC). We start our interpretation by considering the exhaustive testing space (i.e., 3-way tuples). Assume further that each parameter has two values (i.e., a0, a1 for parameter A, b0, b1 for parameter B, and finally, c0, c1 for parameter C).
The exhaustive test suite in which the MVSCA generates the permutations of 3-way parameter-values as tabulated in **Table 1.** The exhaustive test suite of CA generates all the possible combinations of 3-way parameter-values in order as tabulated in **Table 2.** The SCA generates the permutations of 3-way parameters as tabulated in **Table 3.**

Table 1. MVSCA (48, 3, 3,2).

Test Case Number	Multi-Valued Sequence Test Case		
1	a0	b0	c0
2	a0	b0	c1
3	a0	b1	c0
4	a0	b1	c1
5	a1	b0	c0
6	a1	b0	c1
7	a1	b1	c0
8	a1	b1	c1
9	a0	c0	b0
10	a0	c0	b1
11	a0	c1	b0
12	a0	c1	b1
13	a1	c0	b0
14	a1	c0	b1
15	a1	c1	b0
16	a1	c1	b1
17	b0	a0	c0
18	b0	a0	c1
19	b0	a1	c0
20	b0	a1	c1
21	b1	a0	c0
22	b1	a0	c1
23	b1	a1	c0
24	b1	a1	c1
25	b0	c0	a0
26	b0	c0	a1
27	b0	c1	a0
28	b0	c1	a1
29	b1	c0	a0
30	b1	c0	a1
31	b1	c1	a0
32	b1	c1	a1
33	c0	a0	b0
34	c0	a0	b1
35	c0	a1	b0
36	c0	a1	b1
37	c1	a0	b0
38	c1	a0	b1
39	c1	a1	b0
40	c1	a1	b1
41	c0	b0	a0
42	c0	b0	a1
43	c0	b1	a0
44	c0	b1	a1
45	c1	b0	a0
46	c1	b0	a1
47	c1	b1	a0
48	c1	b1	a1

Table 2. CA (8, 3, 3,2).

Test Case Number	Test Case		
1	a0	b0	c0
2	a0	b0	c1
3	a0	b1	c0
4	a0	b1	c1
5	a1	b0	c0
6	a1	b0	c1
7	a1	b1	c0
8	a1	b1	c1

Table 3. SCA (6,3,3).

Test Case Number	Sequence Test Case		
1	a	b	c
2	a	c	b
3	b	a	c
4	b	c	a
5	c	a	b
6	c	b	a

Looking at **Tables 1 and 2**, it is clear that the generated tuples of CA are a subset of the MVSCA's tuples (i.e., the first eight tuples in **Table 1** is identical to **Table 2**). In fact, if only adjacent tuples of the MVSCA is taken into consideration, the MVSCA degrades to CA. Looking at **Tables 1 and 3**, if only one parameter-value is taken into consideration (e.g., the values a0, b0, and c0), the MVSCA degrades to CA. Taking the exhaustive space into account, it is clear that the number of tuples in MVSCA is very large as compared to both CA and SCA. In fact, the number of tuples in MVSCA is the product of CA's tuples and SCA's tuples.

In order to reduce the exhaustive test suites when relaxing the interaction coverage to 2-way (pairwise), we adopt an exhaustive greedy strategy (EGS). In EGS, the exhaustive test suite is generated as a candidate test cases pool. In addition, the generation of all t-way interaction elements (tuples) is followed by searching for test cases that cover the maximum number of uncovered tuples. The generated test case is added to the test suite in one-test-at-a-time fashion. This process is repeated for each test case until all tuples are covered. Moreover, EGS takes an intelligent parameter (combinatorial types) to control the generation of tuples as well as the exhaustive pool, and hence the generated test suite to be either CA, SCA, or MVSCA.

Return to our working example, the interaction parameters are unique as specified by Eq. (1) (i.e., AB, AC, and BC). However, the tuples are different according to the type of combinatorial testing method as tabulated in **Table 4**. Moreover, the exhaustive test suites are different also as tabulated in **Tables 1, 2, and 3**. The number of tuples per parameter interaction is determined by equations 2,4, and 6 for CA, SCA, and MVSCA respectively (i.e., 4, 2, and 8 respectively). The total number of tuples are determined by equations 3,5, and 7 for CA, SCA, and MVSCA respectively (i.e., 12, 6, and 24 respectively). Following the steps of the EGS, the generated test suites and the corresponding covered tuples are tabulated in **Tables 5,6, and 7** for MVSCA, CA, and SCA respectively.

Table 4. The 2-way combinatorial interaction elements for different combinatorial types.

Combinatorial Type	AB	AC	BC
MVSCA	a0b0 a0b1 a1b0 a1b1 b0a0 b0a1 b1a0 b1a1	a0c0 a0c1 a1c0 a1c1 c0a0 c0a1 c1a0 c1a1	b0c0 b0c1 b1c0 b1c1 c0b0 c0b1 c1b0 c1b1
CA	a0b0 a0b1 a1b0 a1b1	a0c0 a0c1 a1c0 a1c1	b0c0 b0c1 b1c0 b1c1
SCA	AB BA	AC CA	BC CB

Table 5. The MVSCA (8, 2, 3, 2).

Test Case Number	Test Case	Tuples Covered
1	a0b0c0	[a0b0], [a0c0], [b0c0]
2	a0b1c1	[a0b1], [a0c1], [b1c1]
3	a1b0c1	[a1b0], [a1c1], [b0c1]
4	a1b1c0	[a1b1], [a1c0], [b1c0]
5	c0b0a0	[c0b0], [c0a0], [c0b0]
6	c0b1a1	[c0b1], [c0a1], [b1a1]
7	c1b0a1	[c1b0], [c1a1], [b0a1]
8	c1b1a0	[c1b1], [c1a0], [b1a0]

Table 6. The CA (4, 2, 3, 2).

Test Case Number	Test Case	Tuples Covered
1	a0b0c0	[a0b0], [a0c0], [b0c0]
2	a0b1c1	[a0b1], [a0c1], [b1c1]
3	a1b0c1	[a1b0], [a1c1], [b0c1]
4	a1b1c0	[a1b1], [a1c0], [b1c0]

Table 7. The SCA (2, 2, 3).

Test Case Number	Test Case	Tuples Covered
1	abc	[ab], [ac], [bc]
2	cba	[cb], [ca], [ba]

To illustrate the construction of test cases for MVSCA, the candidate test cases pool is given in **Table 1.** The maximum number of tuples per test case can be determined by Eq. (1) (i.e., three in our example). The first candidate test case in **Table 1** (a0b0c0) covers three tuples; therefore; it is added to the test set and the tuples covered are deleted from the interaction elements set. Next, it is deleted from the pool. The next candidate test case is a0b0c1 it only covers two tuples (i.e., a0c1 and b0c1 because the tuple a0b0 already covered). The same observation is true for the test case

number 3 in **Table 1**. Each of the test cases numbered 4, 6, 7, 41, 44, 46, and 47 (Table 1) cover three tuples and hence are added to **Table 5.** It should be mentioned that all tuples are covered and this is an indication that the generation is finished. A similar procedure is done to generate CA and SCA. Considering **Table 2 and 3** as candidate pools to generate CA and SCA respectively and their corresponding tuples from **Table 4**. The CA and SCA for our working example are tabulated in **Tables 6 and 7** respectively.
Thus, by relaxing the coverage strength from t=3 to t=2, the exhaustive testing suites for MVCA, CA, and SCA are reduced from 48, 8, 6 to merely 8, 4, 2 respectively.

It should be mentioned that the parameters'-values could be non-uniformed for the system under the test. In this situation, the total number of tuples is determined as the sum of multiplications for each t-way interaction values. For instance, let's consider a system with seven parameters with mixed configuration values denoted by CA (N, 3, ($5^{2}\ 4^{2}\ 2^{3}$)). In this example, there are two parameters with five values, two parameters with four values, and three parameters with two values. An alternative representation of this system could be CA (N, 3, (5 5 4 4 2 2 2)). According to Eq. (1) there are 35 3-way parameter interactions.
Thus, the total number of tuples in CA (N, 3, ($5^{2}\ 4^{2}\ 2^{3}$)) =
[5*5*4+5*5*4+5*5*2+5*5*2+5*5*2+5*4*4+5*4*2+5*4*2+5*4*2+5*4*2+5*4*2+5*4*2+5*2
*2+5*2*2+5*2*2+5*4*4+5*4*2+5*4*2+5*4*2+5*4*2+5*4*2+5*4*2+5*2*2+5*2*2+5*2*2+4
*4*2+4*4*2+4*4*2+4*2*2+4*2*2+4*2*2+4*2*2+4*2*2+4*2*2+2*2*2] =1310.
When take the t-way sequence into account, for the MVSCA (N, 3, ($5^{2}\ 4^{2}\ 2^{3}$)); there are 3! =6 possible permutations for each 3-way interaction elements, hence, the total number of tuples in this case= 1310 *6=7860 tuples.

4. RESEARCH TRENDS IN COMBINATORIAL TESTING

To the best of author knowledge, the notation of MVSCA is not discussed in the literature. In short, earlier works in combinatorial testing mentioned that the generation of a test suite is considered as NP-Hard problem for both CA and SCA **, Nasser, et al., 2018; Younis and Zamli, 2011; Chee, et al., 2013**. That is, there is no single strategy dominant minimal test size. For this reason, many strategies appear in the literature. These strategies can be either algebraic or computational ,**Lei, et al., 2008**. In algebraic strategy, the generation of a test suite is done directly by a mathematical function **,Grindal, et al., 2004; Hartman and Raskin, 2004.** dynamic programming or mathematical transformation **,Colbourn, et al., 2006; Chee, et al., 2013**.
Computational approaches generate all of the t-way tuples and search the uncovered tuples to generate a test case until all tuples are covered by the test suite. These strategies can be either deterministic or non-deterministic **,Cohen, et al., 1997; Lei, et al., 2008**. Some of the combinatorial strategies adopted metaheuristics algorithms for searching the test space **,Ahmed, et al. 2012; Alsewari and Zamli, 2012; Hazli, et al., 2012; Rahman, et al., 2014; Yang, et al., 2014; Ahmed, et al., 2015; Prasad, et al., 2015; Zamli, et al., 2016; Rabbi, 2017; Zamli, et al., 2017; Alsewari, et al., 2018; Nasser, et al., 2018**.
Due to the NP-Hard problem, Colburn makes a dedicated website to record minimal upper bound for CA **,Colbourn, 2019**. Earlier works also reported that the number of tuples increased significantly when increased the number of parameters and/or the strength of coverage and/or the parameter-values. As such, adopting computational approaches can be expensive in terms of the space required to store the tuples and the time required for explicit enumeration **(Younis, et al., 2008)**. For this reason, there is a trend to adopts parallel and distributed computing to speed-up the execution time and distribute the tuples space **,Younis, et al., 2008; Younis and Zamli, 2010; Avila-George et al., 2012; Priyanka and Rana, 2012; Qi et al., 2016**.

5. CONCLUSION

Building from earlier works, this paper presented a new combinatorial covering array called MVSCA. The presented equations demonstrate both the tuple search space and tuple size for MVSCA is a linear product of its counterpart (i.e., superimposing CA onto SCA). That is, MVSCA is a more generalized form for combinatorial testing that inherits the combinatorial explosion problem and NP_Hardness. Thus, it is expected to re-design and re-implement the previous strategies to deal with MVSCA. Moreover, a valuable feature in the future strategies is to make a choice to the type of combinatorial problem in the same tool (i.e., like our illustration example). Furthermore, like Colburn's website, it is desired to make a benchmark website that collects and published the minimal test suite among developed strategies for both SCA and MVSCA. Finally, it is expected to have large execution time and complexity when dealing with MVSCA; therefore, it is important to develop such strategies with parallelism and distributed computing in mind.

REFERENCES

- Ahmed, B.S., Abdulsamad, T.S., and Potrus, M.Y., 2015. Achievement of minimized combinatorial test suite for configuration-aware software functional testing using the cuckoo search algorithm. *Information and Software Technology*, 66 (1), pp., 13-29, http://doi.org/10.1016/j.infsof.2015.05.005.
- Ahmed, B.S., Zamli, K.Z., and Lim, C.P., 2012. Application of particle swarm optimization to uniform and variable strength covering array construction. *Applied Soft Computing, 12 (4),* pp., 1330–1347, http://doi.org/10.1016/j.asoc.2011.11.029.
- Alsewari, A.A., and Zamli, K.Z., 2012. Design and implementation of a harmony-search-based variable-strength t-way testing strategy with constraints support. *Information and Software Technology, 54 (6),* pp., 553–568, http://doi.org/10.1016/j.infsof.2012.01.002.
- Alsewari, A.A., Muaza, A.A, Rassem, T.H, Tairan, N.M, Shah, H., and Zamli K.Z., 2018. One-parameter-at-a-time combinatorial testing strategy based on harmony search algorithm OPAT-HS. *Advanced Science Letters, 24 (10),* pp., 7273-7277, http://doi.org/10.1166/asl.2018.12927.
- Avila-George, H., Torres-Jimenez, J., Rangel-Valdez, N., Carrion, A., and Hernandez, V., 2012. Supercomputing and grid computing on the verification of covering arrays. *J Supercomput, 62 (2),* pp., 916-945, http://doi.org/10.1007/s11227-012-0763-0.
- Chee, Y.M., Colburn, C.J., Horsley, D., and Zhou, J., 2013. Sequence covering arrays. SIAM J. Discrete Math., 27 (4), pp., 1844–1861, http://dx.doi.org/10.1137/120894099.
- Cohen, D.M., Dalal, S.R., Fredman, M.L., and Patton, G.C., 1997. The AETG system: an approach to testing based on combinatorial design. *IEEE Transactions on Software Engineering,* 23 (7), pp., 437–443, http://doi.org/10.1109/32.605761.
- Cohen, M.B., 2004. Designing test suites for software interaction testing. *Doctoral dissertation, University of Auckland,* https://cse.unl.edu/~myra/papers/mbcdiss.pdf (last visit on 2 June 2019).
- Colbourn, C.J., 2019. Covering array tables. http://www.public.asu.edu/~ccolbou/src/tabby/catable.html, (last visit on 2 June 2019).
- Colbourn, C.J., Martirosyan, S.S., Trung, T.V., and Walker, R.A., 2006. Roux-type constructions for covering arrays of strengths three and four. *Designs, Codes and Cryptography, 41 (1),* pp., 33–57, http://doi.org/10.1007/s10623-006-0020-8.

- Grindal, M., Offutt, J., and Andler, S.F., 2004. Combination testing strategies: a survey. *J. Software Testing, Verification, and Reliability, 5 (3),* pp., 167-199, http://dx.doi.org/10.1002/stvr.319.
- Hartman, A., and Raskin, L. 2004. Problems and algorithms for covering arrays. *Discrete Mathematics, 284 (1-3),* pp., 149-156, https://doi.org/10.1016/j.disc.2003.11.029.
- Hazli, M.Z.M., Zamli, K.Z., and Othman, R.R., 2012. Sequence-based interaction testing implementation using bees algorithm. *Proceedings of the 2012 IEEE Symposium on Computers and Informatics (ISCI), Penang, Malaysia,* pp., 81-85, http://doi.org/10.1109/ISCI.2012.6222671.
- Kuhn, D. R., Higdon, J.M., Lawrence, J.F., Kacker, R.N., and Lei, Y., 2012. Combinatorial methods for event sequence testing. *Proceedings of the 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation, Montreal, QC,* pp., 601-609, http://dx.doi.org/10.1109/ICST.2012.1.
- Lei, Y., Kacker, R., Kuhn, D.R., Okun, V., and Lawrence, J., 2008. IPOG/IPOG-D: efficient test generation for multi-way combinatorial testing. *Software Testing, Verification, and Reliability, 18 (3),* pp., 125-148, http://dx.doi.org/10.1002/stvr.v18:3.
- Mahmood B.M.R., Younis M.I., and Ali H.M., 2013. Construction of a general-purpose infrastructure for RFID–based applications, *Journal of Engineering*, 19 (11), pp., 1425-1442.
- Nasser, A.B., Zamli, K.Z., Alsewari, A.A., and Ahmed, B.A., 2018. An elitist-flower pollination-based strategy for constructing sequence and sequence-less t-way test suite. *International Journal of Bio-Inspired Computation, 12 (2),* pp., 115–127, http://dx.doi.org/10.1504/IJBIC.2018.094223.
- Prasad, M.L., Susmitha, S., Sharanya, C.S., and Praneeth, B.V., 2015. Constructing T-way test cases using genetic algorithms. *International Journal of Research Studies in Computer Science and Engineering, 2 (5),* pp., 29-37.
- Priyanka, C.I., and Rana, A., 2012. An effective approach to build optimal t-way interaction test suites over cloud using particle swarm optimization. *In: Das V.V., Stephen J. (eds) Advances in Communication, Network, and Computing, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 108,* pp., 193–198.
- Qi, RZ., Wang, ZJ., and Li, SY., 2016. A parallel genetic algorithm based on spark for pairwise test suite generation. *Journal of Computer Science and Technology, 31 (2),* pp., 417–427, http://doi.org/10.1007/s11390-016-1635-5.
- Rabbi, K.F., 2017. Combinatorial testing strategies based on swarm intelligence. *Ph.D. thesis, School of Computing and Mathematics, Charles Sturt University,* http://researchoutput.csu.edu.au/files/20430339/Khandakar_Thesis_final.pdf (last visit on 2 June 2019).
- Rahman, M., Othman, R.R., Ahmad, R.B., and Rahman, M.M., 2014. Event driven input sequence t-way test strategy using simulated annealing. *Proceedings of the 2014 5th International Conference on Intelligent Systems, Modelling and Simulation, Langkawi, Malaysia,* pp., 663-667, http://doi.org/10.1109/ISMS.2014.119.
- Yang, XS., Deb, S., and Fong, S., 2014. Metaheuristic algorithms: optimal balance of intensification and diversification. *Applied Mathematics and Information Sciences, 8 (3),* pp., 977-983.

- Younis, M.I., 2014. Design and implementation of ICT-based recycle-rewarding system for green environment. *Journal of Engineering*, 20 (7), pp., 36-47.
- Younis, M.I., Abdulkareem H.F., and Ali, H.M., 2015. Construction of graduation certificate issuing system based on digital signature technique, *Journal of Engineering*, 21 (6), pp., 15-36.
- Younis, M.I., Al-Tameemi, M.I., and Hussein M.S., 2017. MPAES: a multiple-privileges access e-door system based on passive RFID technology. *Proceedings of The 7th Scientific Engineering and 1st International Conference on Recent Trends in Engineering Sciences and Sustainability, Baghdad, Iraq, 16-17 May 2017, pp. 375-380.*
- Younis, M.I., and Hussein, T.F., 2018. Design and implementation of a contactless smart house network system. *International Journal of Electrical and Computer Engineering*, 8 (6), pp., 4663-4672, http://dx.doi.org/10.11591/ijece.v8i6.
- Younis, M.I., and Zamli, K.Z., 2010. MC-MIPOG: a parallel t-way test generation strategy for multicore systems. *ETRI Journal*, 32 (1), pp., 73-83, http://doi.org/10.4218/etrij.10.0109.0266.
- Younis, M.I., and Zamli, K.Z., 2011. MIPOG - an efficient t-way minimization strategy for combinatorial testing. *International Journal of Computer Theory and Engineering*, 3 (3), pp., 388-397, http://ijcte.org/papers/337-G452.pdf.
- Younis, M.I., Zamli, K.Z., and. Isa, N.A.M., 2008. A strategy for grid based t-way test data generation. *Proceedings of the 2008 First International Conference on Distributed Framework and Applications, Penang, Malaysia,* pp., 73-78, http://doi.org/10.1109/ICDFMA.2008.4784416.
- Zamli, K.Z., Alkazemi, B.Y., and, Kendall, G., 2016. A tabu search hyper-heuristic strategy for t-way test suite generation. *Applied Soft Computing, 44 (C),* pp., 57–74, http://doi.org/10.1016/j.asoc.2016.03.021.
- Zamli, K.Z., and Younis, M.I., 2010. Interaction testing: from pairwise to variable strength interaction. *Proceedings of the 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, Bornea, Malaysia*, pp., 6-11. http://doi.org/10.1109/AMS.2010.15.
- Zamli, K.Z., Din, F., Baharom, S., and Ahmed, B.S., 2017. Fuzzy adaptive teaching learning-based optimization strategy for the problem of generating mixed strength t-way test suites. *Engineering Applications of Artificial Intelligence, 59 (C),* pp., 35–50, http://doi.org/10.1016/j.engappai.2016.12.014.