ISSN 1392-1215
ELECTRONICS AND ELECTRICAL ENGINEERING
ELEKTRONIKA IR ELEKTROTECHNIKA
2006. No. 8(72)

T170
ELECTRONICS
ELEKTRONIKA

Functional Test Generation Procedures

E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas
Software Engineering Department, Kaunas University of Technology,
Studentų str. 50-406, Kaunas, Lithuania,
e-mail: eduardas.bareisa@ktu.lt, vacius.jusas@ktu.lt, kestutis.motiejunas@ktu.lt, rimantas.seinauskas@ktu.lt

Introduction

The objective of test generation is to find a test sequence that, when applied to a circuit, can be used to distinguish between a good circuit response and a faulty circuit response. The goal is to detect defects, to achieve a given fault coverage and to assure product quality and reliability. The test effectiveness is measured by the achieved fault coverage as well by the cost of performing the test. Test generation is a complex problem with many interacting aspects e.g. the cost of test generation, test length and the quality of generated test. Test generation can be accomplished at different levels: micro-level, gate-level, and functional level [1-3].

In the paper [4] the functional test generation problem was formulated as an optimisation problem and the following objective Ψ was proposed:

$$\Psi = \alpha \sum_{s \in S} F^{s} (X^{\square}) - \beta \, | X^{\square}|, \qquad (1)$$

A set of input stimuli is denoted by $X^{\square}$, and its cardinality (the number of stimuli) - by $|X^{\square}|$. A set of conditions that have to be fulfilled by input stimuli of the set $X^{\square}$ is marked by S. An input stimulus $X \in X^{\square}$ may fulfill several conditions $s \in S$. A condition s may be fulfilled by many input stimuli $X \in X^{\square}$. In order to assess the fulfillment of the conditions $s \in S$ by the set of input stimuli $X^{\square}$, the estimate function F^s is defined. If at least one input stimulus $X \in X^{\square}$ fulfills the condition s, then the estimate function F^s has the value 1, i.e. $F^s (X^{\square}) = 1$, otherwise $F^s (X^{\square}) = 0$. The number of conditions fulfilled by an input stimuli set $X^{\square}$ is equal to the sum of values $F^s (X^{\square})$ taken over all conditions $s \in S$. α, β are positive coefficients. The test generation problem asks for a set of input stimuli at which the function Ψ is maximized.

Some test generation problems may be obtained and solved by changing conditions that have to be fulfilled. When the number of fulfilled conditions is more important factor than the number of input stimuli, we can take the coefficient $\beta=0$. An important aspect of functional test generation is that the fulfillment of the conditions cannot be evaluated analytically, and instead it has to be estimated using simulation techniques only. [4]

One of specific problems in the area of functional test generation is determining the relationship between input and output variables of a given module M.

Let us define the following terms used in the rest of the paper.

Definition 1. The relationship between input x_i and output z_j is called even if a transition 1→0 (0→1) on the input x_i causes the same transition 1→0 (0→1) on the output z_j.

Definition 2. The relationship between input x_i and output z_j is called uneven if a transition 1→0 (0→1) on the input x_i causes the opposite transition 0→1 (1→0) on the output z_j.

Functional test generation can be related to the problem where the input stimuli set has to be found that would determine the parity of the relationship among all input pairs and outputs of the module. The results can be presented using the three-dimensional matrix D, where $d_{i,h,j} = 1$, if there exists at least one input stimulus that determines uneven relationship between the input x_i and the output z_j, and the same input stimulus determines uneven relationship between the input x_h and the output z_j as well. Similarly, $d_{i,h,j} =1$ if there exists at least one input stimulus that determines even relationship between the input x_i and the output z_j, and the same input stimulus determines even relationship between the input x_h and the output z_j as well. Whereas, $d_{i,h,j} = 2$ if there exists at least one input stimulus that determines either even or uneven relationship between the input x_i and the output z_j, and the same input stimulus determines the opposite parity of the relationship between the input x_h and the output z_j. In all other cases $d_{i,h,j} = 0$. We are asked to find a set of input stimuli that would determine maximum values for every entry of the three-dimensional matrix D. The sum of entries in the matrix D is proportional to the number of the fulfilled conditions $s \in S$ divided by two, as the matrix D is symmetrical according to its definition. Thus, when $\beta=0$ and $\alpha=1$, the value of the objective function Ψ is equal to a half of the sum of entries in the matrix D. Let's call this problem of matrix identification the three-dimensional relationship determination problem. We are also interested in various modifications to this problem. We are unaware of analytical methods of solving this three-dimensional relationship problem even in the case when detailed module descriptions are at hands [5].

The aim of this paper is to explore the features of one of test generation subproblems, namely the functional test generation subproblem, and on the basis of the gained experience, to propose a practical method for functional test generation.

Deterministic Search Procedure for Adjacent Input Stimuli

The methods based on the generation of input stimuli adjacent to the selected ones [5] improve convergence of the random generation process. Two input stimuli are adjacent if they differ in the value of a single input. There could be defined a procedure for generation of adjacent input stimuli based on already selected ones. The procedure could iterate the process of generation of adjacent stimuli. The procedure would terminate a generation when no new adjacent input stimuli were formed from the selected ones.

Let the set of stimuli adjacent to an input stimulus X be denotet by $\Theta(X)$. The set of stimuli adjacent to an subset $X^{\square}$ of the input stimuli is $\Theta(X^{\square})= \bigcup \Theta(X)| X\in X^{\square}$.

The procedure PG for generating and selecting adjacent stimuli can formally be defined in the following way:

REPEAT
FOR $X\in \Theta(X^{\square})$
$X^{\square} \leftarrow X^{\square} \bigcup \{X\}| \Psi(X^{\square}) < \Psi(X^{\square} \bigcup \{X\})$
ENDFOR
UNTIL $\Psi(X^{\square})\neq \Psi(X^{\square} \bigcup \Theta(X^{\square}))$
RETURN $X^{\square}$

Let's analyze the capabilities of the procedure PG. For the experiments, we have used the benchmark circuits ISCAS'85 [6]. The strategy of generating adjacent input stimuli allowed to improve solutions for the three-dimentional relationship determination problem. Table 1 presents the results of adjacent stimuli generation and random stimuli generation. Adjacent stimuli generation was started with a single random stimuli. We express the solution quality in percent as the ratio between the obtained value of the objective function and the value of the best-known solution. The best known value of objective function Ψ which we have derived during various experiments is presented in the last column under heading "Best". Table 2 presents the results of adjacent stimuli generation following random stimuli generation, the search size of which was equal to the number presented in column (Table 1) under heading "Number of random stimuli" for each circuit. Observe that the generation of adjacent stimuli improved the solution considerably for three

Table 1. Adjacent and random stimuli generation

Circuit	Number of adjacent stimuli	Value of the objective function Ψ	Solution quality (%)	Number of random stimuli	Value of the objective function Ψ	Best	Solution quality (%)
C432	21064	14518	95,17	21064	15222	15254	99,79
C499	406365	360618	87,41	406365	412736	412736	100
C880	225314	40078	72,49	225314	49474	55282	89,49
C1355	442934	363828	88,15	442934	412736	412736	100
C1908	231089	149242	96,73	231089	153868	154284	99,73
C2670	429085	121538	64,90	429085	128384	188082	68,56
C3540	362985	121428	98,45	362985	121298	123338	98,34
C5315	1905000	264508	98,06	1905000	269646	269726	99,97
C6288	325345	152614	99,87	325345	151529	152814	99,17
C7552	3631863	690928	85,41	3631863	419534	809850	51,82

Table 2. Adjacent stimuli generation following random generation

Circuit	The last column of Table 1	Number of adjacent stimuli	Value of the objective function Ψ	Solution quality (%)	Improvement (%)
C432	99,79	10395	15222	99,79	0
C499	100	120320	412736	100	0
C880	89,49	192716	55156	99,77	10,28
C1355	100	122186	412736	100	0
C1908	99,73	64430	154080	99,86	0,13
C2670	68,56	188514	174722	93,29	24.73
C3540	98,34	194186	122856	99,60	1,26
C5315	99,97	423183	269706	99,99	0,02
C6288	99,17	64655	152712	99,94	0,77
C7552	51,82	1103895	744878	92,08	40,26

circuits where the initial solution was quite far from the best one (C880, C2670, C7552). When the initial solution is close to the best one, the generation of adjacent stimuli improves the solution only slightly, and there is no guarantee of obtaining the best solution even in the case it is quite near. Thus, we can conclude that if the generation of adjacent stimuli did not improve the initial solution, we could expect that the solution can be close to the best one. Information that the generation of adjacent stimuli does not improve the solution is certain information about solution quality as well.

In the paper [4] the random search termination condition was defined as well:

$$P= ((R_i - R_{i/c})/R_i)*100, \text{ where } C>1. \qquad (2)$$

Ri is the number of selected stimuli at the moment when i random stimuli have been generated. Ri/c denotes the number of selected stimuli when i/C random stimuli have been generated. The difference Ri – Ri/c shows how many input stimuli were selected after generating C times more random stimuli. As the random search size increases, P decreases to zero. The rate of decrease depends on the coefficient C. The bigger coefficient means the slower convergence to zero. The value P can be calculated for every random input stimulus which has an index larger than C. If we assume that the termination condition of generation is P=0, this termination condition will be more demanding when the value of the coefficient C is larger.

However, the termination condition P, which is applied during the random generation, cannot be applied in the case of adjacent stimuli. We can calculate the value P during the random stimuli generation till the moment the generation of adjacent stimuli starts up. A possible way to terminate the generation in this case will be discussed in the next section.

Practical Functional Test Generation Method

The ideas and results published in [4] together with the above presented analysis of random search and adjacent stimuli generation allow formulating a practical method for generating functional tests. The principles of optimization approaches applied to solve the quadratic assignment problem [7] will be used. However, we will not use the tabu search approach and the development of the initial solution will not be carried out after the local search, but instead the new random search will start. Our decision is based on the fact that functional test has a lot of input stimuli, which fulfill a condition $s \in S$, and it is not clear which one is the most acceptable. This method incorporates the mentioned termination condition of generation P [4], exploits the advantages of random and deterministic search, as well the feature that the sets of the selected input stimuli can be merged easily in order to obtain a better set of test patterns.

Firstly, a predefined number K of random stimuli are generated and the stimuli that increase the value of the objective function are selected. Then the procedure PG of generating adjacent stimuli is applied to the selected stimuli. These two steps are combined and form the procedure G(K), which finds the initial set of the test stimuli (G(K) →X□). Next, the generation of random and adjacent stimuli is repeated from scratch and generation procedure G(K) finds a new set of stimuli X□1 (G(K) → X□1). During the next step, the stimuli from the set X□1 that increase the value of the objective function are included into the set X□. The iterations of generation and inclusion of stimuli into the set X□ are repeated till we arrive at the state when the stimuli from a set X□1 do not increase the value of the objective function. Then the K, which value denotes the number of randomly generated stimuli, is increased by some parameter ΔK, and the iterations proceed again. The test generation procedure stops when the predefined random search size limit L is reached. The test generation (TG) procedure can be defined formally as follows:

G(K) → $X^{\square}$
REPEAT
REPEAT
G(K) → $X^{\square}_1$
FOR X∈ $X^{\square}_1$
$X^{\square} \leftarrow X^{\square} \bigcup \{X\} \mid \Psi(X^{\square}) < \Psi(X^{\square} \bigcup \{X\})$
ENDFOR
UNTIL $\Psi(X^{\square}) = \Psi(X^{\square} \bigcup X^{\square}_1)$
K←K+ΔK
UNTIL K>L
RETURN $X^{\square}$

The use of the procedure in practice is strongly influenced by the runtime and memory limits of the computer. To overcome this difficulty, the procedure can be modified using heuristic simplifications and improvements. Various experiments were performed before we have implemented the procedure. The results will not be presented here due to a large size of tables; only the conclusions based on them will be given here.

Adjacent stimuli generation starting with a single random stimulus was considered first. In general, 0s and 1s distribute nearly equally in randomly generated stimuli. Therefore, the possibility of starting the generation of adjacent stimuli with boundary stimuli, which have only 0s or 1s, was analyzed. After long-lasting experiments we have concluded that the best point to start generating adjacent stimuli is to take two stimuli where one stimulus has only 0s, and the other one has only 1s. In this case, the generation process converges after analyzing nearly two times less stimuli and the value of the objective function increases by several percents. Furthermore, the application of the adjacent stimuli generation procedure before the random search allows evaluating the search size of random generation more reasonably. Two initial stimuli are included into the stimuli set V. The functional test generation procedure is presented in Fig. 1.

Let's analyze the presented functional test generation procedure. The adjacent stimuli generation for stimuli of the initial set V begins the overall test generation. The value of the objective function Ψ is calculated upon termination of the adjacent stimuli generation. Then the iterative process starts. A new set V1 of selected stimuli is formed. The sets V and V1 are merged. The stimuli that increase the value of the objective function are included into the resulting set V. The new value Ψ1 of the objective function is calculated. In order to evaluate the increase of the value of the objective function, the values Ψ and Ψ1 are compared, and the outcome of the comparison is expressed in percents. If the increase of the value is more than PP percents, the generation is repeated using the same random search size PK. Otherwise, the random search size PK is increased by PD times, and the generation is repeated. The iterations are terminated when the value of the objective function has not increased more than PP percents after enlarging the size of the random search by PD times. When iterations are completed the set V of selected stimuli can be minimized. However, the procedure of minimization will not be discussed here. The iterations can be repeated according to the presented algorithm by taking the set V of selected stimuli as the initial set and enlarging the size PK of the random search.

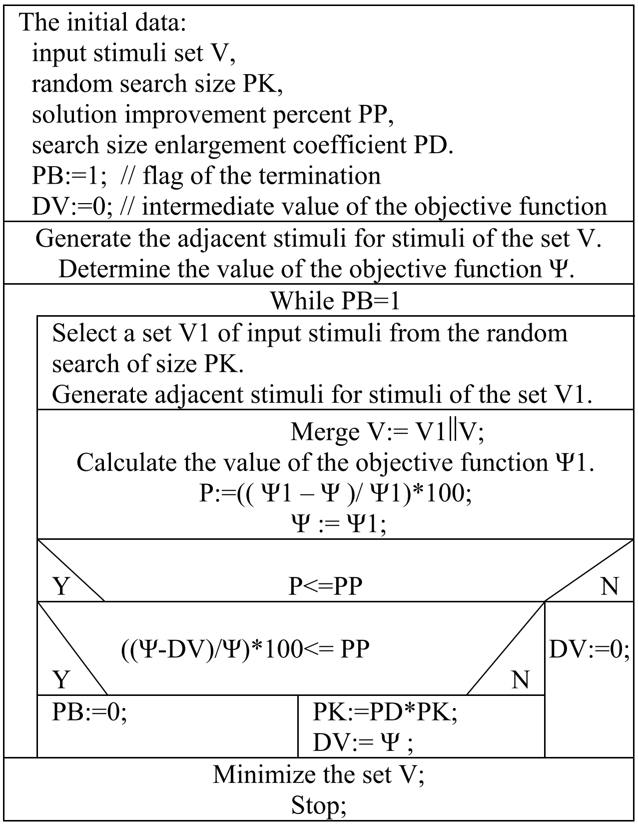

Fig. 1. Functional test generation procedure (FTGP)

We should mention that the adjacent stimuli generation allows improving the search efficiency. The better solution obtained after the random generation allows the achievement of the better final solution after the generation of adjacent input stimuli. However, this finding does not suggest what generation strategy is the most effective. The experiments allowed us conclude that it is worth to use the adjacent stimuli generation procedure as intensively as possible.

The obtained results of the procedure FTGP solving the three-dimensional relationship determination problem for two largest circuits c2670 and c7552 are presented in Tables 3 and 4, respectively.

During the initial iteration, the adjacent stimuli for two initial stimuli, one of which consists of 0s only and the other – of 1s only, were generated. Such a generation strategy allows revealing the search size for the next iterations. Note that 324932 adjacent input stimuli were generated for two initial stimuli; 5245 input stimuli were selected and the obtained value of the objective function 113068 reached 60.1% of the best-known solution. Then 324932 input stimuli were generated randomly during the first iteration, 2273 were selected and the obtained value of the objective function was 128378. Whereupon 510297-324932=185365 adjacent stimuli were generated for the selected stimuli and the total number of the analyzed stimuli reached 510297. 2943 input stimuli were selected and the obtained value of the objective function 172608 reached 91.7% of the best-known solution. The obtained 2943 input stimuli were merged with 5245 input stimuli that were selected during the initial iteration. Thus, 8188 input stimuli were obtained in total. After stimuli merge operation there were selected 2946 and the obtained value of the objective function was 172614. Then 324932 input stimuli were generated randomly again and 3001 input stimuli were selected at the end of the generation of adjacent stimuli (the obtained value of the objective function was 173826). The selected stimuli were merged with 2946 stimuli, which were selected before, and the value 181452 (96.4% of the best-known solution, i.e. solution improvement - 4.88%) of the objective function was obtained. Consequently, the iterations have to be continued because the coefficient PP of solution improvement was set to 1. However, after the repetition of the generation of 324932 input stimuli again, the solution improvement (0.52%) was less than one percent. This outcome indicated that the size of random search has to be increased, and it was doubled to 649862. During the next two iterations the obtained value of the objective function 186506 reached 99.1% of the best-known solution. However, after the second iteration the solution improvement was less than one percent, what indicated that the size of random search has to be increased. Having doubled the search size once more to 1299724, the solution improvement was less than one percent and the procedure FTPG terminated its work. 5049936 input stimuli were analyzed in total and the obtained value of the objective function 187130 reached 99.4% of the best-known solution. The results of generation 5049936 input stimuli randomly and supplementing the selected stimuli with the adajcent ones are presented in the row under heading "**R**". The solution quality comparing with derived using procedure FTPG is less in two percents. Further results were obtained after reducing the PP value to 0.1 and performing an additional number of iterations. In this case, the process converged when obtained value of the objective function 187990 reached 99.9% of the best-known solution. 8968840 input stimuli were analyzed in total. The random generation of such number of stimuli allows obtaining 1.5 % worse solution quality comparing with derived using procedure FTPG. The merging of selected stimuli after 9 iterations and after random 8968840 input stimuli generation produced the value of the objective function 188082 that reached the value of the best-known solution. The results, which are presented in the last row, are obtained after generating one hundred million input stimuli randomly. Note that even in this case after generating adjacent stimuli the best-known value of the objective function was not obtained.

Table 3. Intermediate results obtained for circuit C2670

Iteration	Random generation			Adjacent generation			Merge			Solution quality (%)
	Number of stimuli	Objective function Ψ	Selected stimuli	Total number of stimuli	Objective function Ψ	Selected stimuli	Stimuli	Objective function Ψ	Selected stimuli	
0				324932	113068	5245				60,1
1	324932	128378	2273	510297	172608	2943	8188	172614	2946	91,7
2	324932	128256	2273	514759	173826	3001	5947	181452	3224	96,4
3	324932	125230	2255	492841	165830	2679	5903	182424	3285	96,9
4	649862	128184	2308	836747	173164	2955	6240	184950	3393	98,3
5	649862	126140	2330	840837	178264	3011	6404	186506	3518	99,1
6	1299724	132990	2407	1490335	177718	3006	6524	187130	3547	99,4
R	5049936	150993	2503	5248351	183218	3135				97,4
7	1299724	141602	2406	1604131	178964	2954	6501	187506	3508	99,6
8	1299724	138340	2419	1499219	180406	3146	6654	187906	3616	99,9
9	1299724	137380	2393	1487683	178740	2961	6577	187990	3577	99,9
R	8968840	160488	2502	9169456	185100	3177				98.4
							6754	188082	3457	100
	100000000	182366	2614	100205511	187270	3152				99,6

Table 4. Intermediate results obtained for circuit C7552

Iteration	Random generation			Adjacent generation			Merge			Solution quality (%)
	Number of stimuli	Objective function Ψ	Selected stimuli	Total number of stimuli	Objective function Ψ	Selected stimuli	Stimuli	Objective function Ψ	Selected stimuli	
0	1604131	714928	17116							88,2
1	1604131	391719	6288	2865462	792948	14583	31699	793506	14597	97,9
2	1604131	391032	6292	2731445	745250	13511	28108	793778	14628	98,0
3	3208262	418960	6722	4384683	779076	13717	28345	799358	14142	98,7
R	9370138	457753	7425	10424180	745708	12623				92,0
4	3208262	418472	6741	4468857	797230	14262	28404	806784	14476	99,6
5	3208262	424200	6833	4440450	795774	14103	28579	809504	14396	99,9
6	3208262	417547	6755	4457685	797030	14481	28877	809850	14705	100
R	100000000	548563	9320	101103223	805932	13625				99,51
R	200000000	596086	10468	201108397	809742	12414				99,98
Additional three iterations										
1	19080784	480323	7963	20261816	795894	13634	28339	809956	14130	100,013
2	19080784	484000	7911	20269258	793870	13790	27920	809998	14062	100,018
3	19080784	491469	7915	202644657	797202	13581	27643	810040	13953	100,023

The same experiment was carried out for the circuit c7552 as well (Table 4). Having set PP=1, the solution quality 98.7% was got. Continuing the iterations with PP=0.1, the best-known value of the objective function thus far was obtained (19080784 input stimuli were analyzed in total). The results of generation one and, respectively, two hundred million input stimuli are presented in the rows under heading “R”, however the best-known value of the objective function was not reached.

In order to demonstrate that maximal value of the objective function was not reached three additional iterations was carried out generating randomly 19080784 input stimuli and merging results with input stimuli of best obtained value of objective function. The results presented in the last three rows of Table 4 show that the best known value of objective function was slightly increased.

The random search method without termination condition can not be evaluated adequately. The decision on computation stopping may require plenty enough resources. A method without termination condition is unpractical.

The proposed procedure FTGP uses a reasonable search termination condition. The termination condition of the procedure is based on the rate of solution improvement. Such an approach can be applied for solving other optimization problems. Additionally, the procedure FTGP uses solutions’ merge operation successfully in order to improve its performance results. However, this is typical not for all optimization problems.

The adjacent stimuli generation is limited by the selected stimuli, as the adjacent stimuli generation uses the selected stimuli only. This restriction ensures the convergence of the procedure of adjacent stimuli generation, and only a small part of input stimuli are available for analyzing during the procedure. Therefore, when the initial set of the selected stimuli is changed, the set of stimuli, which are got during the generation of adjacent stimuli, changes as well. Thus, various sets of stimuli, which may increase the value of the objective function, are available during the adjacent stimuli generation. This fact allows explaining the usefulness of generating new adjacent stimuli in order to increase the solution quality.

Conclusions

In practice besides the deterministic algorithms of test generation the heuristic algorithms are used quite widely. The latter algorithms find the input stimuli that detect the fault but they cannot ensure that the fault is undetectable. And such are random search algorithms. In the paper the problem of test generation is formulated as an maximization problem. That enabled to use the random optimization methods for solving it. This is especially relevant for generating black-box functional tests. Such generation is based mostly on simulation and the use of deterministic algorithms is very limited practically.

A deterministic procedure of adjacent stimuli generation was suggested. It is based on the assumption that input stimuli that are similar to test patterns have good testing features. The search among such input stimuli improves the overall efficiency and the convergence speed of the search.

The nature of the task of functional test generation allows to select the test patterns from two independent test sets and to obtain a solution of no worse quality. That enabled to construct an iterative procedure for generating functional tests. The proposed procedure evaluates the rate of solution convergence, chooses the search size and uses solutions' merge operation. This procedure can be applied to solving of other optimization problems if there is a possibility to construct a new solution of no less quality from two solutions.

References

1. **Breuer M.A., Friedman A.D.** Diagnosis & Reliable Design of Digital Systems. – Computer Science Press, 1976.
2. **Cheng K.T., Agrawal V.D.** Unified Methods for VLSI Simulation and Test Generation. – Computer Science Press, 1989.
3. **Bareiša E., Jusas V., Motiejūnas K, Šeinauskas R.** Identifying Legal and Illegal States in Synchronous Sequential Circuits Using Test Generation // INFORMATICA. – 2003. – 14. – P. 135–154.
4. **Bareiša E, Jusas V., Motiejūnas K., Šeinauskas R.** Defining Random Search Termination Conditions. Electronics and electrical engineering. – Kaunas: Technologija, 2006. – No. 2(66). – P. 26–31.
5. **Bareiša E, Jusas V., Motiejūnas K., Šeinauskas R.** The Realization-Independent Testing Based on the Black Box Fault Models // INFORMATICA. – 2005. – 16. – P. 19–36.
6. **Brglez F. and Fujiwara H.** A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran // IEEE Int'l Symp. on Circuits and Systems. – June 1985. – Vol. 3. – P. 663–698.
7. **Misevicius A.** A Tabu Search Algorithm for the Quadratic Assignment Problem // Computational Optimization and Applications. – 2005. – Vol. 30. – P. 95–111.

Submitted for publication 2006 07 03

E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Functional Test Generation Procedures // Electronics and Electrical Engineering. – Kaunas: Technologija, 2006. – No. 8(72). – P. 43–48.

The aim is to explore the features of one of test generation sub problems, namely the functional test generation sub problem, and on the basis of the gained experience, to propose a practical method for functional test generation. In the paper presented analysis of random search methods and adjacent stimuli generation allowed formulating a practical procedure for generating functional tests. This method exploits the advantages of random and deterministic search, as well the feature that the sets of the selected input stimuli can be merged easily in order to obtain a better set of test patterns. Ill. 1, bibl. 7 (in English; summaries in English, Russian and Lithuanian).

Э. Барейша, В. Юсас, К. Мотеюнас, Р. Шейнаускас. Процедуры функциональной генерации тестов // Электроника и электротехника. – Каунас: Технология, 2006. –№ 8(72). – С. 43–48.

Основная цель – проанализировать свойства функциональной генерации тестов и на основе данного анализа предложить практический метод для генерации функциональных тестов. В статье представленный анализ методов случайного поиска и генерации смежных наборов позволил создать практическую процедуру генерации функциональных тестов. Предложенная процедура эффективно использует преимущества случайного и детерминистического поиска, а также свойство генерации тестов, которое с целью получения тестов лучшего качества позволяет объединить в одно множество тесты, полученные в отдельных решениях. Ил. 1, библ. 7 (на английском языке; рефераты на английском, русском и литовском яз.).

E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Funkcinio testų generavimo procedūros // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2006. – Nr. 8(72). – P. 43–48.

Pagrindinis tikslas – ištyrinėti funkcinio testų generavimo savybes ir, remiantis šia analize, pasiūlyti praktinį funkcinių testų generavimo metodą. Pateikta atsitiktinės paieškos metodų ir gretimų rinkinių generavimo analizė leido sudaryti praktinę funkcinių testų generavimo procedūrą. Ši procedūra efektyviai panaudoja atsitiktinės ir deterministinės paieškos teigiamybes bei testų generavimo ypatybę, kad, norint gauti kuo geresnės kokybės testus, atskirų sprendimų metu gautos testinės aibės gali būti lengvai sujungtos į vieną. Il. 1, bibl. 7 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).

DOI: 10.5755/j02.eie.10787