INFORMATICA, 2007, Vol. 18, No. 1, 3–26

Functional Test Generation Based on Combined Random and Deterministic Search Methods

Eduardas BAREIŠA, Vacius JUSAS, Kęstutis MOTIEJŪNAS, Rimantas ŠEINAUSKAS
Software Engineering Department, Kaunas University of Technology
Studentų 50-406, LT-51368 Kaunas, Lithuania
e-mail: kestas@soften.ktu.lt

Received: December 2005

Abstract. The aim of this paper is to explore some features of the functional test generation 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 method for generating functional tests. This method incorporates the analyzed termination conditions of generation, exploits the advantages of random and deterministic search, as well as the feature that the sets of the selected input stimuli can be merged easily in order to obtain a better set of test patterns.

Key words: functional test generation, random search, adjacent stimuli.

1. 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 and 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 (Breuer and Friedman, 1976; Cheng and Agrawal, 1989; Bareisa *et al.*, 2003).

The aim of this paper 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. When dealing with the development of test generation methods one usually faces various random and deterministic search problems. Specific methods, based on algebraic formulae manipulations, were developed and being used (Srinivasan *et al.*,1993; Stanion *et al.*, 1995). Functional test generation is usually based on simulation during which output values are computed for given input stimuli. In the general case, the functional test generation problem can be formulated in the following way.

The input stimulus to the functional module M having n input and m output variables is described by the vector $X =< x_1, x_2, x_3, \ldots, x_i, \ldots, x_n>$, and the output response is described by the vector $Z =< z_1, z_2, z_3, \ldots, z_j, \ldots, z_m >$, where Z values directly depend on the X values, $x_i \in \{0, 1\}$ and $z_j \in \{0, 1\}$. In general, 2^n input stimuli may occur. The collection of all possible sets of input stimuli is denoted by X^D. A set of input stimuli is denoted by $X^\Box$, where $X^\Box \in X^D$, and its cardinality (the number of stimuli) – by $|X^\Box|$. Suppose there is given a set S of conditions that have to be fulfilled by input stimuli of the set $X^\Box$. An input stimulus $X \in X^\Box$ may fulfil several conditions $s \in S$. A condition s may be fulfilled by many input stimuli $X \in X^\Box$. In order to assess the fulfilment of the conditions $s \in S$ by the set of input stimuli $X^\Box$, the estimate function F^s is defined. If at least one input stimulus $X \in X^\Box$ fulfils the condition s, then the estimate function F^s has the value 1, i.e., $F^s(X^\Box) = 1$, otherwise $F^s(X^\Box) = 0$. The number of conditions fulfilled by an input stimuli set $X^\Box$ is equal to the sum of values $F^s(X^\Box)$ taken over all conditions $s \in S$. On the base of the estimate function F^s, the objective function Ψ is defined as follows:

$$\Psi = \alpha \sum_{s \in S} F^s(X^\Box) - \beta|X^\Box|, \quad \text{where } \alpha, \beta \text{ are positive coefficients.}$$

The test generation problem asks for a set of input stimuli at which the function Ψ is maximized:

$$\operatorname*{Max}_{X^\Box \in X^D} \Big(\alpha \sum_{s \in S} F^s(X^\Box) - \beta|X^\Box|\Big).$$

Specific 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 fulfilment of the conditions cannot be evaluated analytically, and instead it has to be estimated using simulation techniques only.

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

The problem can be stated as follows. Given a description of the behaviour of a module M with n input variables and m output variables, determine the relationships of inputs and outputs. The only input stimulus can reveal the existing relationship between the input x_i and the output z_j. That is, when the transition occurs on the input x_i, the transition is observed on the output z_j also. The results can be presented using the relationship matrix $M = m_{i,j}$, where $m_{i,j} = 1$ if the input i is connected to the output j, and $m_{i,j} = 0$ otherwise. Each nonzero entry of this matrix corresponds to the fulfilled condition $s \in S$. The condition is fulfilled if the matrix entry is nonzero, and the condition is not fulfilled if the matrix entry is equal to zero. Let's call this problem of matrix identification the relationship determination problem No. 1. The relationship matrix can be constructed analytically from the description of the behaviour of a module. Various software tools for solving this problem can be used. Finding a solution gets more complicated if the description of the behaviour is not applicable for finding the solution analytically, i.e., when the description is changed or there are no software tools for analysis of the description of the behaviour, and only simulation tools can be used, which calculate output signal values for every input stimulus. This is typical for descriptions of behaviour at a high level of abstraction. In this case, the solution has to be found using methods based on simulation.

The determination of the relationship matrix using methods based on simulation requires analyzing all possible input stimuli. This is practically impossible to accomplish for real modules. Consequently, one may search for a solution using random search for maximizing the objective function Ψ. When $\beta = 0$ and $\alpha = 1$, the value of the objective function Ψ is equal to the number of nonzero entries in the matrix M.

Let us define the following basic 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 $1 \to 0$ $(0 \to 1)$ on the input x_i causes the same transition $1 \to 0$ $(0 \to 1)$ on the output z_j.

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

The test generation problem for functional delay faults (Michael and Tragoudas, 2002) may be formulated as a problem of determining even and uneven relationships among inputs and outputs of the module. The results can be presented using the relationship parity matrix L, where $l_{i,j} = 0$, if the relationship between input x_i and output z_j doesn't exist, $l_{i,j} = 1$, if there exists either even or uneven relationship, and $l_{i,j} = 2$ when both relationships exist. The set of input stimuli has to be found; this set determines all relationships among inputs and outputs of the module. Nonzero entries of the matrix correspond to relationship conditions $s \in S$. Let's call this problem of matrix L identification the relationship determination problem No. 2. In the case of this relationship problem the fulfilled conditions $s \in S$ are determined differently than in the case of the relationship problem No. 1. When $\beta = 0$ and $\alpha = 1$, the value of the objective function Ψ is equal to the sum of entries of the matrix L.

Functional test generation can also 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 also. 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 also. 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 relationship determination problem No. 3. 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 (Bareisa *et al.*, 2005).

It is typical for all these problems that the maximum number of the fulfilled conditions s is not known in advance. The result of the unlimited random search nears to the maximum number of the fulfilled conditions. Therefore it is important to evaluate the process convergence during the random search and to make a decision about a termination of search on this basis.

The more complex is verification of the fulfilled conditions, the more difficult is the development of analytical methods for solving the problems. Therefore methods based on simulation and random and deterministic search have to be used.

2. A Brief Review of Random Search Methods

Pure random search consists of sampling a stream of independent and identically distributed random vectors and then selecting the best one as a solution. Pure random search is very easy to implement. Unfortunately, the convergence is extremely slow in most cases of interest. Much attention has been devoted to modifying pure random search to improve its convergence rate. There are approaches involving adaptive construction of distribution, which assign more mass to promising regions of the search space or shrink the domain by some factor. The rigorous mathematical comparisons of different approaches are reported in (Appel and Radulovic, 2000).

The adaptive random search methods differ in several respects. In particular, they differ in the choice of the neighbourhood structure, in the mode of selecting a candidate for solution, in the way the next point is determined and in the way the estimate of the best solution is defined (Andradottir, 1999; Holland, 1975). Random search methods that require only a small number of attempts per iteration can move more rapidly towards the best solution than random search methods for which each iteration involves a substantial amount of computer effort. Most (adaptive) random search methods choose the current estimate as the best solution after m iterations have been completed with the same solution.

To characterize problems arising in functional test generation we analyzed the nature of the random search process by applying this technique to the most complex problem among those formulated above, namely, the three-dimensional relationship determination

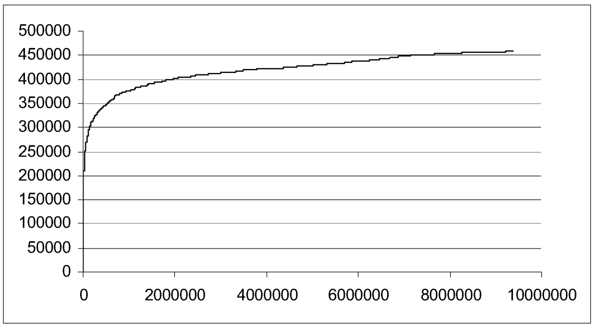

Fig. 1. The values of the objective function obtained during random search.

problem No. 3. For one instance of the problem the values of the objective function obtained during random search are plotted in Fig. 1. The number of generated stimuli is shown on the X axis, and the value of the objective function Ψ, when the coefficient β is zero, is shown on the Y axis.

A test generation task formulated as the objective function Ψ maximization problem can be solved using various methods (Spall, 2003). However the most convenient strategy and criteria have to be chosen for every problem being solved. The attempts to apply methods based on search area restrictions, which strive to perform the search only in the most promising areas (Kushner and Yin, 2003).), were not shown to be successful for the test generation problem. Also, we failed to develop effective methods based on the ideas of genetic algorithms. The created methods had no considerable advantages over using a pure random search to solve the problem. The results of these experiments will not be presented here due to their huge scope. Only the below described approach based on generation of input stimuli adjacent to the already selected ones had a perceptible effect. Therefore, the main attention will be paid to criteria for search termination, to the generation of adjacent stimuli and to the development of a practical search procedure that would use the solution merge effect.

3. Defining Random Search Termination Conditions

As it is well known, random search requires some termination condition to be defined. The simplest termination condition is the number of randomly generated input stimuli; the best solution is chosen from these stimuli. The number of randomly generated input stimuli for finding the best solution depends in large part on an instance of the problem being solved.

Having a solution obtained after performing a fixed number of random search iterations nothing can be said about its quality. In practice, frequently it is not possible to obtain the best solution; often one has no such purpose. First of all, one faces limited time and computer resources. However, in practice it is always worth to evaluate how much the solution could be improved, and how much time and computer resources it would take. Usually such an evaluation is considerably more expensive than finding the solution. When looking for random search termination conditions one has to evaluate both aspects – cost of solution finding and its quality estimation.

More information about the solution quality is gained when a sequence of solutions is constructed. Comparing the distribution of objective function values of these solutions one can decide whether the time allotted for random search is enough. Wide scattering of objective function values shows that termination of random search is premature. Theoretically, when the time allotted for random search is long enough, best or close to best solutions are to be found.

Suppose we have N solutions with objective function values $\Psi_1, \Psi_2, \Psi_3, \ldots, \Psi_i, \ldots, \Psi_N$. Let $\Psi_{\max}$ denote the maximum of these values. The distribution of objective function values is characterized by the following quantity expressed in percents:

$$D = \left(\left(\left(\sum_{i=1}^{N}(\Psi_{max} - \Psi_i)\right)/N\right)/\Psi_{\max}\right) * 100.$$

For the experiments, we have used the benchmark circuits ISCAS'85. The best values of the objective function Ψ solving the relationship determination problem No. 2 for these circuits were obtained analytically in (Michael and Tragoudas, 2002) for the case $\alpha = 1$ and $\beta = 0$. For each of these circuits million input stimuli were generated ten times and input stimuli that fulfil the conditions $s \in S$ were selected. During every run of the search procedure we recorded the number "$|X^{\square}|$" of the selected input stimuli and current number of the last selected stimulus "Last". The analytically obtained best values of the objective function Ψ (for $\alpha = 1$ and $\beta = 0$) (Michael and Tragoudas, 2002) are listed in column "Best" of Table 1. The sum of entries of relationship matrix was used as

Table 1

1000000 random stimuli

Circuit	Last (Min)	Last (Max)	Last (Max)/ Last (Min)	$\|X^{\square}\|$		$D\%$	Ψ	Best
				Min	Max			
C432	2862	4899	1.7	55	67	0	540	540
C499	82877	102442	1.2	485	522	0	5184	5184
C880	80123	192373	2.4	171	218	0	1326	1326
C1355	70645	88129	1.2	470	511	0	5184	5184
C1908	96861	124203	1.3	282	329	0	3004	3004
C2670	965000	984508	1.0	184	194	3.73	3016	3320
20 million	16550550	16550550	1.0	257	257	?	3320	3320
C3540	90093	438846	4.9	227	263	0	2588	2588
C5315	83650	205820	2.5	552	605	0	10540	10540
C6288	81668	649648	7.9	115	131	0	3068	3068
C7552	958935	981376	1.0	633	686	2.65	9334	12188
90 million	82999170	82999170	1.0	851	851	?	10564	12188

the value of the objective function. In order to unify the values of the objective function with (Michael and Tragoudas, 2002), the sums were doubled. These sums, which are the values of the objective function with the coefficients $\alpha = 1$ and $\beta = 0$, are listed in the penultimate column. As we see, generation of a million random stimuli was sufficient to obtain best solutions for all the circuits except c2670 and c7552. The procedure was able to produce the best solution for the circuit c2670 when generating 20 million input stimuli; whereas no best solution was obtained for the circuit c7552 even generating 90 million random stimuli. The results of these generations are presented in Table 1 in an additional line under the circuit's results. For each circuit Table 1 shows the largest and the smallest amount of the selected stimuli (Columns 5 and 6), the largest and the smallest number of the last selected stimulus (Columns 2 and 3) taken over all 10 random generations. As we see, in order to find the best solution the procedure generated a very different number of input stimuli (from several thousands to nearly a million) for different circuits. The value that defines the scattering of the results is presented in the column under the heading "$D\%$". The number 0 indicates that the same objective function value was obtained for all 10 random searches. This result testifies that a million of input stimuli randomly generated for these circuits are sufficient for obtaining the best result. Ten random searches in turn achieving a solution with the same objective function value signify that there is a high possibility for this solution to be the best. If different solutions were obtained from ten random searches in turn, a possibility to find the best solution from them is low. The values 3.73% and 2.65% of D were obtained for the circuits c2670 and c7552 respectively. The best solution for the circuit c2670 was obtained when the random search size was enlarged 20 times; whereas there was no success in finding the best solution for the circuit c7552 even after enlarging the random search size 90 times. For the circuits with no best solution, the process of input stimuli selection was continued till the end of generation. The ratio between the maximal and the minimal number of last selected stimulus random stimuli is nearly eight for the circuit c6288 (the fourth Column in Table 1). Whereas, the number of the selected stimuli varies between the maximum and the minimum only little.

In total, 100 million input stimuli were generated for all the circuits in Table 1, ten generations up to one million for each one (except a separate generation of 20 and 90 million input stimuli for the circuits c2670 and c7552 respectively). There was no need in generating a million input stimuli for 8 circuits. The number of the last selected stimulus is presented in the third column of the table. It indicates how many random stimuli were generated for obtaining the best solution in the worst case out of ten trials. As it can be seen from this column, there was no need in generating a million input stimuli randomly for some circuits. Therefore, it may be worth to perform random search several times enlarging the number of generated random stimuli for those circuits for which the technique failed to get the solution of the same quality in ten runs. In order to explore this prediction, the random search procedure was applied for each circuit ten times, generating 100 000 random stimuli in each case. The results are presented in Table 2, which has the same structure as Table 1. In this experiment, the best solutions were obtained only for three circuits. In general, ten million random stimuli in total were generated for all the

Table 2

100000 random stimuli

Circuit	Last (Min)	Last (Max)	$\lvert X^{\square}\rvert$ Min	$\lvert X^{\square}\rvert$ Max	$D\%$	Ψ	Best
C432	2708	5251	56	76	0	540	540
C499	79255	89611	478	519	0	5184	5184
C880	84411	99638	173	200	>0	1326	1326
C1355	71877	82246	452	521	0	5184	5184
C1908	92973	99951	284	340	>0	3002	3004
C2670	63253	98654	130	144	>0	2788	3320
C3540	79747	99995	233	275	>0	2588	2588
C5315	84272	99651	548	593	>0	10540	10540
C6288	56950	96385	105	134	>0	3068	3068
C7552	98259	99468	494	560	>0	10564	12188

circuits. Therefore, by generating a million input stimuli for every circuit for which the best solution was not achieved, the same result as in the first experiment will be obtained after generating 80 million input stimuli in total. The results might be better if the number of generated stimuli in the first random search was enlarged slightly, say, to 300 000 or if more than two random search iterations with different number of generated stimuli were used. It is not easy to predict what generation steps are to be included in order to obtain the same result under the minimum amount of generated input stimuli. The question how to solve a problem using a sequence of several algorithms was analyzed in (Abraitis and Sheinauskas, 1969) and will not be discussed in this paper.

It is worth to relate the condition of the termination of random search dynamically to the number of the already selected input stimuli. The generation can be terminated when the total number of the generated input stimuli exceeds the number of the selected input stimuli multiplied by a coefficient K. The results of the experiment are presented in Table 3. The random search procedure was run ten times for every circuit. The left part of Table 3 contains the minimum and the maximum numbers of the generated input stimuli, the values of the parameter D, and the values of the objective function Ψ for the case of $K = 1000$. The random search procedure was rerun with $K = 5000$ for the circuits for which no success in obtaining the same solution ten times was observed (the right part of Table 3). Note that to achieve the same best solutions or even better ones (as in Tables 1 and 2), less than 78 million of input stimuli had to be generated in total. Termination of random search generation based on the number of the selected input stimuli allows adjusting of the whole solution process to meet to resource requirements necessary for the generation more flexibly and effectively.

When the same value of the objective function was reported during several runs of the random search procedure, it could be expected that the best solution was obtained. The more times the same value of the objective function is achieved during several independent runs of the random search procedure, the more is a probability that the obtained

Table 3

Generation according to the number of selected stimuli

Circuit	$K = 1000$				$K = 5000$				Best
	Min gen.	Max gen.	$D\%$	Ψ	Min gen.	Max gen.	$D\%$	Ψ	
C432	58000	65000	0	540					540
C499	458000	503000	0	5184					5184
C880	146000	206000	0.15	1326	181084	1060000	0	1326	1326
C1355	492000	514000	0	5184					5184
C1908	278000	338000	0	3004					3004
C2670	108000	147000	6.13	2676	988263	1015000	5.77	3072	3320
C3540	240000	263000	0.15	2588	486098	1375000	0	2588	2588
C5315	559000	597000	0	10540					10540
C6288	109000	130000	0.65	3068	181084	1060000	0	3068	3068
C7552	623000	651000	3.43	8970	2523409	2530000	2.69	9872	12188
Total		3414000				4359938			

solution is the best one. However, the random search size is increased also. If the best solution was not obtained in all the cases, then it remains unclear what the next size of the random search has to be used. The dependence of D on the random search size and the possibilities to predict this size when the value of D will become 0 were analyzed also. It was noticed that the convergence is very slow and, therefore, any prediction would be very imprecise.

Let's analyze how the duration time of random search can be estimated during random search itself. This would allow making decision reasonably regarding the moment when the random search has to be terminated. In the process of random search, a big amount of input stimuli is selected in the beginning, and then more and more random

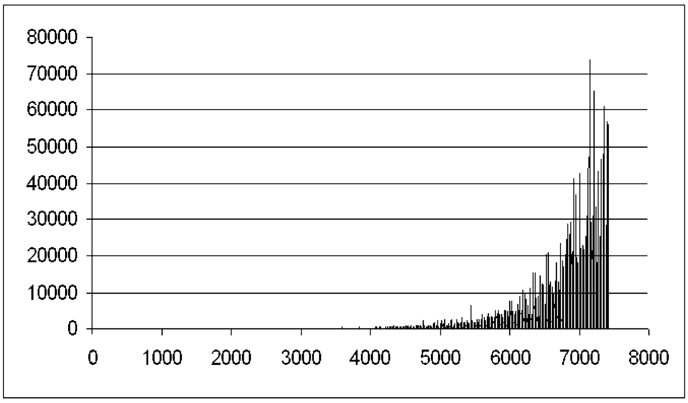

Fig. 2. The dependence of the number of stimuli required for selecting a new testing stimulus.

stimuli have to be generated for selecting a new stimulus. The trade-off between the selected current number of input stimuli and the number of generated random stimuli for relationship determination problem No. 3 is shown in Fig. 2. The current number of the selected stimulus is shown on the X axis, and the number of random stimuli that had to be generated till a new stimulus was selected is shown on the Y axis.

Note that the initial 4000 input stimuli were selected very fast. Later, the number of generated stimuli, required for selecting a new stimulus, increased considerably and this number is quite different.

The completeness of search can be defined by the ratio of how many the last input stimuli selections are rarer than in the beginning of search. Let R_i be the number of selected stimuli at the moment when i random stimuli have been generated. The percent $P = ((R_i - R_{i/c})/R_i) * 100$, where $C > 1$, has a tendency to decrease during random search. $R_{i/c}$ denotes the number of selected stimuli when i/C random stimuli have been generated. The difference $R_i - R_{i/c}$ 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. The dependence of P on the increase of the number of generated random stimuli can be determined and on the basis of this dependence one can evaluate how many random stimuli are required till P would get zero value. If the required number of random stimuli cannot be generated due to the limited calculation resources, there is a possibility using the value of P to evaluate how far the obtained solution is from the best one. The values of P obtained during one run of random search are plotted in Fig. 3. The number of selected stimuli is shown on the X axis, and the coefficient P is shown on the Y axis.

However, the analysis of values of P (Fig. 3) shows that the forecasting when the value of P becomes zero is quite problematic.

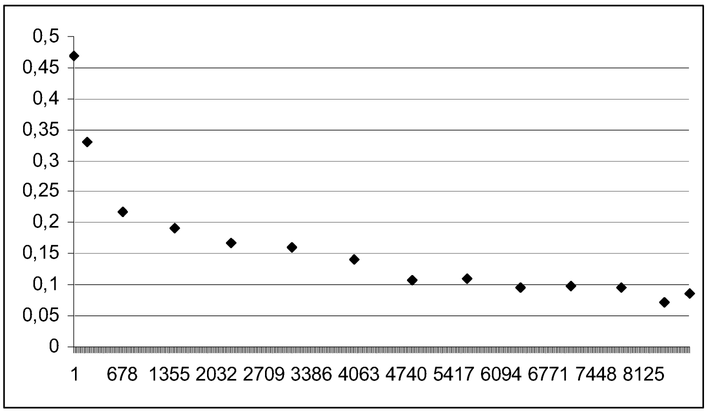

Fig. 3. Values of P versus the number of selected stimuli.

Table 4

Generation according to the number of selected stimuli

Circuit	$C = 2$				$C = 3$			Best
	Min	Max	K	Worst Ψ	Min	Max	K	
C432	5744	8930	10	540	8622	11757	10	540
C499	163880	181774	10	5184	252240	291060	10	5184
C880	193948	355604	9	1324	302022	551604	10	1326
C1355	143778	164968	10	5184	222822	256842	10	5184
C1908	194186	242276	10	3004	282540	371805	10	3004
C2670	22760	29965284	4	2488	28043244	42046896	10	3320
C3540	177504	526998	9	2586	275688	1040529	10	2588
C5315	171400	466498	10	10540	294960	573948	10	10540
C6288	183186	710416	7	3060	375942	1679799	10	3068
C7552	*	*	*	*	*	*	*	12188
Total	1256386	32622748		34754	30058080	46824240		34754

Table 4 reports the results of ten random generations when termination condition of generation was based on the number of selected stimuli. In the case of coefficients $C = 2$ and $C = 3$, random generation reached zero value of P for all the circuits, except circuit c7552 (the computer used for experiments was to slow). Nevertheless, the generation did not achieve the best solution for the circuits c880, c2670, c3540 and c6288 when coefficient C was 2, though the difference from the best value was marginal (except the circuit c2670). The worst obtained solution is shown in the column under heading "Worst Ψ". The number of experiments (out of ten) we succeeded to obtain the best solution is shown in the columns under heading "K". The smallest and the largest number of input stimuli till the termination condition was fulfilled ($P = 0$) are presented in columns "Min" and "Max", respectively. The increase of the coefficient C from 2 to 3 allowed obtaining the best solutions for all the circuits in all ten runs. On this basis, we recommend to use the value 3 of the coefficient C while solving such problems in practice, and to invoke the random search procedure only once thereby reducing the need for computer resources. The total number of the analyzed random stimuli for the worst cases was approximately 47 millions only. The generation which fails in fulfilling the termination condition has to be stopped when it runs out of resources. In such case the closeness of the obtained solution to the best one could be evaluated using P.

The relationship determination problem No. 3 was solved also, when the termination condition of the generation was $P = 0$ and the coefficient $C = 3$. In the rest of this paper we will take $C = 3$ in the formula for P. The results are presented in Table 5. The second column "Gen." shows the number of randomly generated stimuli, and the third column "Last. Sel." shows the number of the last selected stimulus. 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 5

Stimuli selection for the relationship determination problem No. 3

Circuit	Gen.	Last Sel.	Ψ	$\lvert X^{\square}\rvert$	P	Best
C432	62568	20856	15254	1125	0	15254
C499	137397	45799	412736	3352	0	412736
C880	100000000	47290644	55280	4904	0.17	55282
C1355	151473	50491	412736	3365	0	412736
C1908	11485245	3828415	154284	2491	0	154284
C2670	100000000	99337376	182366	2849	4.38	188082
C3540	100000000	81215067	123322	7018	0.05	123338
C5315	22650798	7550266	269726	4637	0	269726
C6288	100000000	99531738	152678	2140	0.04	152814
C7552	100000000	99807091	548563	8869	9.61	810040

The random search termination condition $P = 0$ was fulfilled for 5 circuits and the best-known solution was obtained in all the cases. 100 million input stimuli were generated for each of the remained five circuits, but there was a failure in fulfilling the termination condition. One can judge about the quality of obtained solution according to the value P. The value P is very small for three circuits, so we can expect that the solution is not far from the best one and the termination condition would be fulfilled after increasing the number of input stimuli. The results on other two circuits indicate that the search for the best-known solution would be long enough.

Next we will discuss the possibilities of predicting the best solution as well as the search duration. The analysis is based on the generation of input stimuli for five circuits, where the termination condition of the generation was fulfilled. The break points were determined for the following values of P: 10, 5, 3, and 1. The results are presented in Table 6.

The columns under heading "Gen.%" show us how much percents of input stimuli were generated until the specified value of P was reached, comparing with the number of

Table 6

The break points based on P value

Circuit	$P = 10$		$P = 5$		$P = 3$		$P = 1$	
	Gen.%	Ψ%.	Gen.%	Ψ%	Gen.%	Ψ%	Gen. %	Ψ%
C432	23	98.36	33	99.58	48	99.86	81	99.99
C499	44	99.77	59	99.95	72	99.98	99	99.99
C1355	40	99.75	53	99.95	59	99.89	82	99.99
C1908	1	98.73	2	99.13	3	99.35	5	99.64
C5315	0.5	96.4	1	97.91	2	99.26	4	99.70
Min	0.5	96.4	1	97.91	2	99.26	4	99.70
Max	44	97.77	59	99.95	72	99.98	99	99.99

stimuli, which were generated until zero value of P was reached. For the considered circuits these values are very different. The columns under heading "Ψ" express in percents how the found solution is close to the best one. The last two rows present the minimum and maximum values of the corresponding column. The minimum and the maximum values differ a lot in Columns "Gen.%", therefore, the expected limits of the number of random stimuli to be generated till value P reaches zero are large. This result leads to a conclusion that it is meaningful to estimate the number of random stimuli to be generated till the termination condition is fulfilled for every problem individually. Nevertheless, the results in Table 6 indicate that the input stimuli are selected mainly in the beginning of the generation and afterwards the solution improves very slowly. Even for the value of P equal to ten the found solution differs from the best one less than by four percents. Also, we can see that when P decreases the difference between the maximal and the minimal search sizes expressed in percents increases, but the difference between maximal and minimal values of Ψ that characterize the solutions quality decreases. Based on these findings we can conclude that the estimation of solution precision should be more reliable than the estimation of the search size of random generation till P reaches zero value.

4. Deterministic Search Procedure for Adjacent Input Stimuli

The methods based on the generation of input stimuli adjacent to the selected ones (Holland, 1975) 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 denoted by $\Theta(X)$. The set of stimuli adjacent to a 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 be formally 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. Firstly, the solutions provided by the procedure PG, which starts with a single random input stimulus (Table 7), will be analyzed and then will be compared with the results of generating random input stimuli (Table 8), the number of which is the same as the number of generated adjacent 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.

Table 7

Adjacent stimuli generation starting with a single random stimulus

Circuit	Number of generated stimuli	Value of the objective function Ψ	Solution quality (%)	Number of selected stimuli ($\|X^{\square}\|$)
C432	2116	528	97.77	209
C499	77175	5184	100	2100
C880	24107	1262	95.17	603
C1355	77591	5184	100	2121
C1908	29652	2998	98.80	1108
C2670	35360	1564	47.10	559
C3540	26403	2578	99.61	901
C5315	214764	10252	97.26	2619
C6288	18753	3068	100	586
C7552	231977	10978	90.07	2854

Table 8

Random stimuli generation

Circuit	Number of generated stimuli	Value of the objective function Ψ	Solution quality (%)	Number of selected stimuli ($\|X^{\square}\|$)
C432	2116	540	100	124
C499	77175	5184	100	1011
C880	24107	1318	99.40	365
Adjacent	38393	1326	100	374
C1355	77591	5184	100	1106
C1908	29652	3004	100	619
C2670	35360	2512	75.66	247
Adjacent	63165	3196	96.26	444
C3540	26403	2584	99.85	509
Adjacent	41281	2588	100	514
C5315	214764	10540	100	1176
C6288	18753	3042	99.15	225
C7552	231977	8494	69.69	1420
Adjacent	395733	11738	96.30	1843

The second column in Table 7 lists the number of adjacent stimuli generated for all circuits starting with a single random stimulus. Table 8 has the same structure as Table 7, but the same number of stimuli was generated randomly. One can observe that the random generation produced better results except for the largest circuits c6288 and C7552. The number of the selected input stimuli is nearly two times smaller also. The adjacent stimuli for four circuits with no best solution were generated after the random generation has been completed. The four rows “Adjacent” (Table 8) provide the total number of generated input stimuli, the value of the objective function and the number of selected input stimuli for the circuits C880, C2670, C3540 and C7552, respectively. The best solutions were not obtained for the circuits C2670 and C7552 even after extending the generation. However, the solution quality is over 96%.

Generally, the final result depends on the size of the random search. The objective function Ψ increases by enlarging the random search size and then using the adjacent stimuli generation. It is not clear what size of the random search has to be chosen in order to obtain the best solution after adjacent stimuli generation. To this end the experiments with the circuits c2670 and c7552 were performed changing the size of the random search. During the experiments we have noticed that the better solution was obtained after the random search, the less relatively the solution was improved by adjacent stimuli generation. We increased the random search size by a fixed value and examined the solution obtained after adjacent stimuli generation. The random search size for the circuit c2670 was increased up to 310274 input stimuli in such a way, the generation of adjacent stimuli afterwards increased the volume of the search up to 339373 input stimuli and then the best solution with 459 selected input stimuli was obtained. Experimenting by analogy with the circuit c7552 the random search size was increased up to 8364882 (after adjacent stimuli generation up to 8542976) and it enabled to obtain the value of the objective function 11738; the latter value is less than the value of the best-known solution 12188. However, it has to be mentioned that after generating 90 million of input stimuli randomly a rather less value of the objective function $\Psi = 10564$ was reached (Table 1). This demonstrates the benefit of adjacent stimuli generation once again.

In summary, it is quite problematic to estimate the size of the random generation for obtaining the best solution. It has to be mentioned that the random generation for the circuit c2670 was terminated after reaching the value $P = 8.87\%$ (when $C = 2$) and the generation of adjacent stimuli afterwards enabled to obtain the maximum value of the objective function. Whereas the random generation for the circuit c7552 was terminated at the less P value $P = 3.9\%$ (when $C = 2$), however the generation of adjacent stimuli afterwards did not ensure the obtaining of the maximum value of the objective function.

The strategy of generating adjacent input stimuli allowed improving solutions for the relationship determination problem No. 3 also. Table 9 presents the results of adjacent stimuli generation and random stimuli generation. Adjacent stimuli generation was started with a single random stimulus. Note that the random generation achieved better results in seven cases out of ten.

Table 10 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 9) under heading “Number of random stimuli” for each circuit. Observe that the generation of adjacent stimuli improved the solution considerably for three 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

Table 9

Stimuli generation for the relationship determination problem No. 3

Circuit	Number of adjacent stimuli	Value of the objective function Ψ	Solution quality (%)	Number of random stimuli	Value of the objective function Ψ	Solution quality (%)
C432	21064	14518	95.17	21064	15222	99.79
C499	406365	360618	87.41	406365	412736	100
C880	225314	40078	72.49	225314	49474	89.49
C1355	442934	363828	88.15	442934	412736	100
C1908	231089	149242	96.73	231089	153868	99.73
C2670	429085	121538	64.90	429085	128384	68.56
C3540	362985	121428	98.45	362985	121298	98.34
C5315	1905000	264508	98.06	1905000	269646	99.97
C6288	325345	152614	99.87	325345	151529	99.17
C7552	3631863	690928	85.30	3631863	419534	51.79

Table 10

Adjacent stimuli generation following random generation

Circuit	The last column of Table 9	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	51.79	1103895	744878	91.96	40.17

one. Information that the generation of adjacent stimuli does not improve the solution is certain information about solution quality as well.

Recall that after generating 100 million input stimuli the termination condition $P = 0$ was not reached for five circuits (Table 5). We made attempt to improve this outcome using adjacent input stimuli generation. The obtained results are presented in Table 11. First, we generated 100 million random stimuli for each circuit. The achieved value of objective function and the number of selected stimuli after generation of 100 million random stimuli are given in the columns 2 and 3, respectively. Next, we used the selected stimuli for the adjacent input stimuli generation. The results of this generation are presented in the columns 4, 5 and 6. The adjacent input stimuli generation improved the solution for all the circuits except C880.

Table 11

100 000 000 input stimuli

Circuit	Value of objective function Ψ	Number of selected stimuli	Number of adjacent stimuli	Value of objective function Ψ	Number of selected stimuli	Increase of the objective function value	Best
C880	55280	4904	194030	55280	4904	0	55282
C2670	182366	2849	205511	187270	3259	4904	188082
C3540	123322	7018	197128	123332	7022	10	123338
C6288	152678	2140	69632	152802	2176	124	152814
C7552	548563	8869	1103223	805932	12880	257369	810040

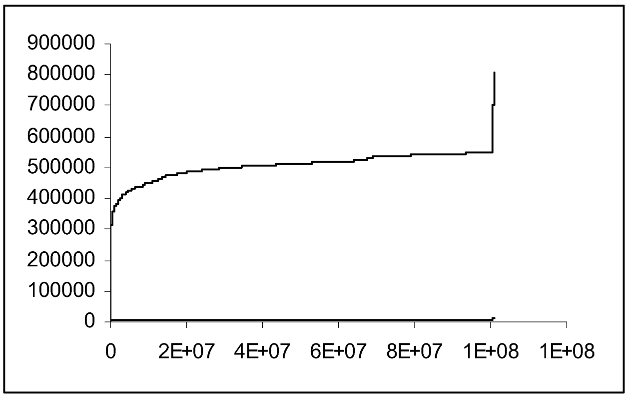

Fig. 4. The alternation of objective function using the generation of adjacent stimuli after random generation.

The largest improvement of the solution is observed for the circuits c2670 and c7552. However, there was no case that the adjacent stimuli generation would yield the best solution. This effect confirms once more the fact that the generation of adjacent stimuli after random generation allows improving the solution though does not assure obtaining of the best solution. The values of the objective function when the generation of adjacent stimuli is applied after random generation for the relationship determination problem No. 3 are plotted in Fig. 4. The number of the analyzed input stimuli is shown on the X axis, and the values of the objective function are shown on the Y axis. Note that the selection from the adjacent input stimuli at the termination leads to a significant increase of the value of the objective function.

However, the termination condition P, which was 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.

5. Practical Functional Test Generation Method

The presented analysis of methods of random search and adjacent stimuli generation for solving functional input and output relationship problem allows formulating a practical method for generating functional tests. This method incorporates the analyzed termination conditions of generation, exploits the advantages of random and deterministic search, as well as 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 $(G(K) \to X^{\square})$. Next, the generation of random and adjacent stimuli is repeated from scratch and generation procedure $G(K)$ finds a new set of stimuli $X_1^{\square}(G(K) \to X_1^{\square})$. During the next step, the stimuli from the set $X_1^{\square}$ that increase the value of the objective function are included into the set $X^{\square}$. The iterations of generation and inclusion of stimuli into the set $X^{\square}$ are repeated till we arrive at the state when the stimuli from a set $X_1^{\square}$ 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:

$G(K) \to X^{\square}$
REPEAT
REPEAT
$G(K) \to X_1^{\square}$
FOR $X \in X_1^{\square}$
$X^{\square} \leftarrow X^{\square} \bigcup \{X\} | \Psi(X^{\square}) < \Psi(X^{\square} \bigcup \{X\})$
ENDFOR
UNTIL $\Psi(X^{\square}) = \Psi(X^{\square} \bigcup X_1^{\square})$
$K \leftarrow K + \Delta K$
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 presented in Table 7. 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

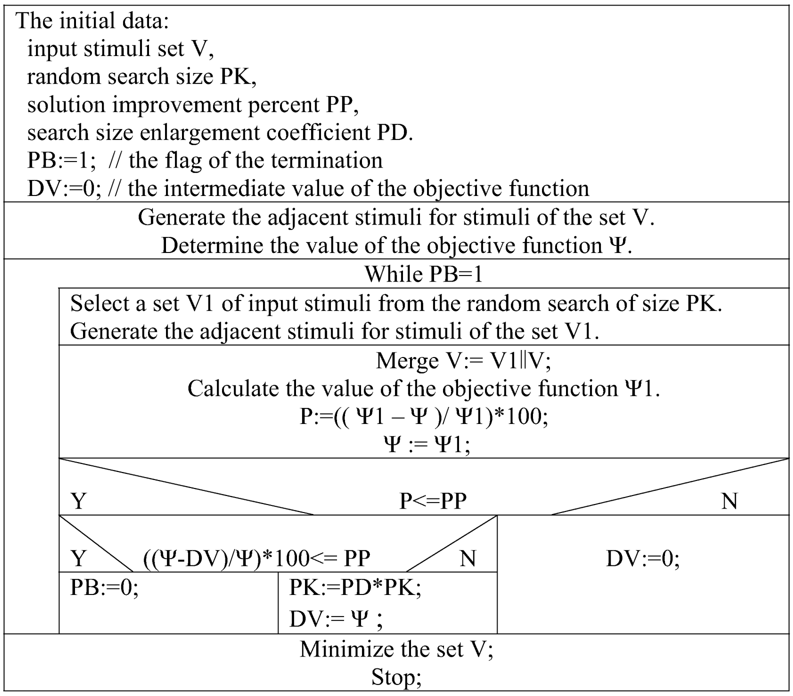

Fig. 5. Functional test generation procedure (FTGP).

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. 5.

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 $\Psi 1$ of the objective function is calculated. In order to evaluate the increase of the value of the objective function, the values Ψ and $\Psi 1$ 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.

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 relationship determination problem No. 3 for two largest circuits c2670 and c7552 are presented in Tables 12 and 13, 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 stim-

Table 12

Intermediate results obtained for circuit C2670

Iteration	Random generation			Adjacent generation			Merge			Solution quality (%)
	Number of stimuli	Ψ	Select. stimuli	Total number of stimuli	Ψ	Select. stimuli	Stimuli	Ψ	Select. 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

uli 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 9th iteration 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.

The same experiment was carried out for the circuit c7552 also (Table 13). Having set $PP = 1$, the solution quality 98.68% was got. Continuing the iterations with $PP = 0.1$, the value 809850 of the objective function was obtained (solution quality 99.98%; 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 value 809850 of the objective function was not reached.

Afterwards three additional iterations were carried out generating randomly 19080784 input stimuli and merging results with input stimuli derived after sixth iteration. The results presented in the last three rows of Table 13 show that then the best known value of objective function was got.

The proposed procedure FTGP uses a reasonable search termination condition. The termination condition of the procedure is based on the rate of solution improvement.

Table 13

Intermediate results obtained for circuit C7552

Iteration	Random generation			Adjacent generation			Merge			Solution quality (%)
	Number of stimuli	Ψ	Select. stimuli	Total number of stimuli	Ψ	Select. stimuli	Stimuli	Ψ	Select. stimuli	
0	1604131	714928	17116							88.26
1	1604131	391719	6288	2865462	792948	14583	31699	793506	14597	97.96
2	1604131	391032	6292	2731445	745250	13511	28108	793778	14628	97.99
3	3208262	418960	6722	4384683	779076	13717	28345	799358	14142	98.68
R	9370138	457753	7425	10424180	745708	12623				92.06
4	3208262	418472	6741	4468857	797230	14262	28404	806784	14476	99.60
5	3208262	424200	6833	4440450	795774	14103	28579	809504	14396	99.93
6	3208262	417547	6755	4457685	797030	14481	28877	809850	14705	99.98
R	100000000	548563	9320	101103223	805932	13625				99.49
R	200000000	596086	10468	201108397	809742	12414				99.96
				Additional three iterations						
1	19080784	480323	7963	20261816	795894	13634	28339	809956	14130	99.99
2	19080784	484000	7911	20269258	793870	13790	27920	809998	14062	99.995
3	19080784	491469	7915	202644657	797202	13581	27643	810040	13953	100

Additionally, the procedure FTGP uses solutions' merge operation successfully in order to improve its performance results.

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 also. 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.

6. 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 a maximization problem. That enabled to use the random and deterministic search methods for solving it. This is especially relevant for generating black-box functional tests. In this case the test generation is based mostly on simulation and the use of only deterministic algorithms is very limited practically.

The random search may last very long. The quality of the solution depends on the tackled task. The random search that spans too short may produce not qualitative solution, however, a long random search may be inefficient and waste computer resources. It is especially relevant when the task is solved for the first time. Therefore, the defining of random search termination conditions is an essential problem. In many cases the termination condition determines the quality of the solution. It is demonstrated that during functional test generation various random search termination conditions may be used and the quality of the obtained solution may be evaluated. The presented research results enable to choose the appropriate termination conditions for a proper search size and precision of the solution reasonably.

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. It is evaluated that the adjacent stimuli generation allowed improving the efficiency of random search up to 31.9%. Consequently, it is recommended the integrated use of random and adjacent stimuli generation during functional test design process.

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. The proposed technique enabled to reduce the search space up to 10.5 times in comparison to pure random search.

References

Abraitis, L., R. Sheinauskas (1969). Towards selection of optimal algorithm sequences. effectiveness and precision of computational algorithms. In *Proceedings of International Symposium*. Kiew. Vol. 5, pp. 3–18.

Andradottir, S. (1999). Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Transactions on Modeling and Computer Simulation*, **9**(4), 349–380.

Appel, M.J., D. Radulovic (2000). Accelerated random search. In *Proceedings of 16th IMACS World Congress 2000 on Scientific Computing*. Lausane, Switzerland. pp. 329–343.

Bareisa, E., V. Jusas, K. Motiejunas, R. Seinauskas (2005). The realization-independent testing based on the black box fault models. *Informatica*, **16**(1), 19–36.

Bareisa, E., K. Motiejunas, R. Seinauskas (2003). Identifying legal and illegal states in synchronous sequential circuits using test generation. *Informatica*, **14**(2), 135–154.

Breuer, M.A., A.D. Friedman (1976).*Diagnosis & Reliable Design of Digital Systems*. Computer Science Press.

Cheng, K-T., V.D. Agrawal (1989). *Unified Methods for VLSI Simulation and Test Generation*. Computer Science Press.

Holland, H.J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Kushner, H.J., G.G. Yin (2003). *Stochastic Approximation and Recursive Algorithms and Applications* (2nd ed.), Springer-Verlag, New York.

Michael, M., S. Tragoudas (2002). ATPG tools for delay faults at the functional level. *ACM Transactions on Design Automation of Electronics Systems*, **7**(1), 33–57.

Spall, J.C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control*. Wiley, Hoboken, NJ.

Srinivasan, S., G. Swaminthan, J.H. Aylor, M.R. Mercer (1993). Combinational circuit ATPG using binary decision diagrams. In *Proceedings of VLSI Test Symposium*. pp. 251–258.

Stanion, R.T., D. Bhattacharya, C. Sechen (1995). An efficient method for generating exhaustive test sets. *IEEE Transactions on Computer-Aided Design*, **14**(12), 1516–1525.

E. Bareiša graduated from Kaunas Polytechnic Institute in 1987. Currently he is in position of professor at Software Engineering Department, Kaunas University of Technology, Lithuania. His research interests include high-level synthesis and VLSI test generation.

V. Jusas graduated from Kaunas Polytechnic Institute in 1982. Currently he is in position of professor at Software Engineering Department, Kaunas University of Technology, Lithuania. His research interests include VLSI test generation at various levels of abstraction.

K. Motiejūnas graduated from Kaunas Polytechnic Institute in 1981. Currently he is in position of professor at Software Engineering Department, Kaunas University of Technology, Lithuania. His research interest is test generation for digital circuits.

R. Šeinauskas graduated from Kaunas Polytechnic Institute in 1972. He got his doctor habilitus from Leningrad Energetics Institute in 1982. Currently he is in position of professor at Software Engineering Department, Kaunas University of Technology, Lithuania. His research interest is VLSI test generation.

Kombinuotas atsitiktinės paieškos ir deterministinių metodų taikymas funkcinių testų generavime

Eduardas BAREIŠA, Vacius JUSAS, Kęstutis MOTIEJŪNAS, Rimantas ŠEINAUSKAS

Straipsnyje nagrinėjamos funkcinių testų generavimo ypatybės. Remiantis atsitiktinės paieškos metodų ir gretimų testinių rinkinių generavimo analize suformuluotas praktinis funkcinių testų sudarymo metodas. Pasiūlytas metodas apima straipsnyje tyrinėtas atsitiktinės paieškos pabaigos nustatymo sąlygas ir galimybę apjungti kelis nepriklausomus sprendinius siekiat pagerinti projektuojamo testo kokybę.