BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 56, No. 1, 2008

Heuristic methods to test frequencies optimization for analogue circuit diagnosis

P. JANTOS*, D. GRZECHCA, T. GOLONEK, and J. RUTKOWSKI

Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, 16 Akademicka St., 44-100 Gliwice, Poland

Abstract. This paper presents methods for optimal test frequencies search with the use of heuristic approaches. It includes a short summary of the analogue circuits fault diagnosis and brief introductions to the soft computing techniques like evolutionary computation and the fuzzy set theory. The reduction of both, test time and signal complexity are the main goals of developed methods. At the before test stage, a heuristic engine is applied for the principal frequency search. The methods produce a frequency set which can be used in the SBT diagnosis procedure. At the after test stage, only a few frequencies can be assembled instead of full amplitude response characteristic. There are ambiguity sets provided to avoid a fault tolerance masking effect.

Key words: analog electronic circuits, analog fault diagnosis, fuzzy logic, gene expression programming, genetic algorithm, simulated annealing.

1. Introduction

An ongoing miniaturization and an increasing complexity of analog electronic circuits (AEC) or integrated circuit (IC) cause a necessity of developing reliable and efficient diagnostic and testing methods. There are well known testing procedures of digital circuits and devices. However, methods of testing analog and mixed circuits have not been developed, yet. There are several problems to solve in the aim of developing standardized AEC diagnosis methods. The most important of them are a variety of analog signals and elements, a larger set of possible faults in comparison to digital circuits, a limited accessibility to the measurement points, and a fault tolerance masking effect [1].

A material of fault diagnosis is very complex. Generally, there are three aims of diagnosis: a fault detection, a fault localization, and a fault identification. A fault detection is the most basic diagnosis test. The purpose is to determine whether the circuit under test (CUT) is damaged or healthy, the "GO/NO GO" test. The other question is to find a source of a fault. Answer to this problem is obtained with a fault isolation procedure. Once the fault is localized it may be important to identify the fault (determining a type and a value of the fault). Taking under consideration used diagnosing tools, three groups of diagnosis procedures can be distinguished: simulation before test (SBT) methods, simulation after test (SAT) methods, and a built-in self-test (BIST). The SBT methods are based on simulations of a certain and previously chosen set of a CUT faults before the explicit test phase. It allows for shortening the diagnosis total time. The goal of the SAT methods is, in most cases, an identification of faults. These methods are efficient in soft faults identification. A disadvantage of these routines is high computing cost and long analyses' time. The BIST requires designing a whole circuit in the way that allows for independent diagnosis of chosen test blocks (often test blocks are chosen in the way they are equal to functional blocks).

An integrated circuit (IC) testing may be performed in a time domain, a frequency domain (AC testing) and a direct currents domain (DC testing). The DC testing makes it possible to obtain results with high reliability level. The AC or time domain testing, though, allows for gathering more information about the CUT state than DC testing without a need of measurements in more than one test point. The presented procedures of finding optimal frequency set for purpose of AEC testing are a dictionary (signature) method based on ambiguity sets (AS) concept. The proposed methods may be implemented as a preparation for signature dictionary constructing in SBT routines. A very important asset of methods is a self-adapting size of AS [2–4].

In the Section 2 diagnostic procedure and idea of AS is explained. The Sections 3, 4 and 5 contain descriptions of the three proposed frequency optimization approaches based on heuristic computations. Next, in Section 6 exemplary results of test frequencies searching are presented and in Section 7 some final conclusions are placed.

2. Diagnostic procedure

For the practical IC design tolerances, limited accuracy of test measurements and simulation models inaccuracies have to be considered. That is why the IC measurements (e.g. RMS voltage, RMS current) can not be predicted precisely, and only the boundary values for the measurements can be designated. Predicted ranges of the test measurements define AS [3–4]. Figure 1 illustrates an example of a RMS voltage decom-

*e-mail: pjantos@polsl.pl

position for a CUT states from S_0 up to S_5, test node N_1 (e.g. output node of a CUT) and test frequencies f_1 and f_2. The sensitivity of the measure for tolerance dispersions of a circuit parameters (e.g. resistances, capacitances) depends on frequency, so the minimal widths $\Delta N_1(f_k)$ of AS should be calculated individually ($k = 1, .., K$) for each sinusoidal stimulus. To make the method robust to the other practical inaccuracies, the AS sizes should be maximally extended.

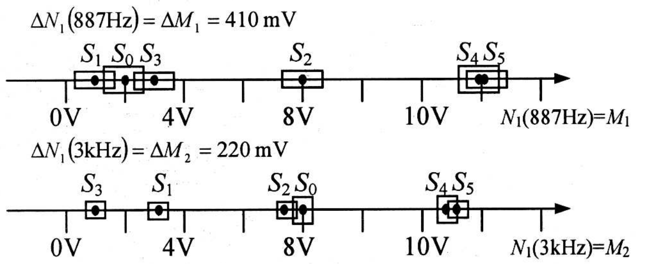

Fig. 1. Exemplary RMS voltage decomposition to for following CUT states (S_0, ..., S_5)

2.1. Before test stage. At this stage a new diagnostic system is created. The system should allow for obtaining maximum information about the CUT with the use of a minimal set of frequencies. It has been assumed that a CUT has N observed test nodes:

$$\mathbf{N} = \{N_1, N_2, .., N_N\}. \tag{1}$$

Amplitude responses have been calculated at all accessible nodes N for M test frequencies. So, the set of test points (test measurements) is represented by a vector **M**:

$$\mathbf{M} = \{M_1, M_2, \ldots, M_N, M_{N+1}, \ldots M_{2N}, \ldots, M_{N \cdot M}\}. \tag{2}$$

The CUT states set is given by a vector:

$$\mathbf{S} = \{S_0, S_1, .., S_L\}, \tag{3}$$

where S_0 represents a healthy circuit state. The following states ($S_1, \ldots, S_L$) represent hard faults. L is a number of hard faults taken into account. The amplitude characteristics have been measured in the chosen test nodes **N**. The most common strategy is measuring a characteristic on the CUT output ($N = 1$). Taking $f_{\min}$ and $f_{\max}$ as:

$$f_{\min} = 10^X \wedge f_{\max} = 10^Y \wedge X, Y \in C \wedge X < Y. \tag{4}$$

Total points number of amplitude characteristics:

$$M = 1000 \cdot \log\left(\frac{f_{\max}}{f_{\min}}\right). \tag{5}$$

Hence, the total frequencies set may be given with a vector:

$$\mathbf{F}_{all} = \left\{f_{\min}, .., f_{\max}, f_{i+1} = 10^{\frac{1}{1000}} f_i\right\} \wedge \|\mathbf{F}_{all}\| = M. \tag{6}$$

All presented methods produce a frequency vector:

$$\mathbf{F} = \{f_1, f_2, .., f_K\} \wedge \mathbf{F} \in \mathbf{F}_{all} \tag{7}$$

that a fault separability and detectability remains at the same levels as with the use of full set $\mathbf{F}_{all}$ (i.e. minimization set **F** power with simultaneous detection and localization levels maximization).

To determine an influence of elements tolerance on a CUT amplitude characteristic a number of Monte Carlo analyses (MCa) has been computed. Tolerance value of resistive elements is tol_R and reactive elements (capacitors and coils) tol_X. For the circuit considered in Section 6 $tol_R = 2\%$ and $tol_X = 5\%$. A result of the MCa simulations are upper and lower envelopes of amplitude responses of the CUT with respect to parameters' values changes. This leads to ambiguity region calculation. A difference between envelopes determines an ambiguity set for each of M frequencies:

$$\mathbf{\Delta M} = \{\Delta M_1, \Delta M_2, .., \Delta M_M\}. \tag{8}$$

Two states are separable if the distance (amplitude response) between them is greater than ΔM_k for frequency f_k. Frequencies selection routines are described in sections III, IV and V.

2.2. Test stage. During the test stage each measure from the CUT is classified to the adequate AS that allows to determine the fault signature. For the example voltage decomposition from Fig.1, all faults are separated from fault-free circuit (test go/no go), CUT is intact (state S_0) for:

$$M_1 = 2\text{V} \pm 410\text{mV} \wedge M_2 = 8\text{V} \pm 220\text{mV}.$$

However, AS for states S_4 and S_5 overlaps for frequencies f_1 and f_2, so additional measurements are necessary for their isolation (fault location and identification are limited).

3. Gene Expression Programming System Based Approach

The Gene Expression Programming (GEP) is an evolutionary algorithm (EA). GEP integrates features of genetic algorithm (GA) and genetic programming (GP). In following parts of this chapter a brief introduction to GEP has been presented, implementation issues and a fitness function described [5–6].

3.1. Individuals coding. Gene Expression Programming individuals have been built with chromosomes (genes strings) of fixed length. The chromosomes have been decoded to phenotypes of tree structure, i.e. expression trees (ET). There is no requirement of using all genes of chromosomes in the process of expressing individuals. Therefore, chromosomes of the same length code ET of different structure and complexity level. There is a multiploid implementation of individuals possible. In such a case, construction of a proper joining function is required.

Genes that chromosomes are coded with, may be divided into two groups functions set **H**, and a terminals set **T**.

The functions set may contain any functions possible to describe over the elements that belong to the terminals set. It may be arithmetic operations, set operations, logic operations, etc. Both, variables and constants may be elements of $\mathbf{T}$.

There has been a set of frequencies coded with each individual of a population in the presented method. It has been decided that set $\mathbf{H}$ will contain only one element (parameter), that is a sum of sets, which allows for adding a new frequency to the set $\mathbf{F}$:

$$\mathbf{H} = \{'\cup'\}. \tag{9}$$

where $\cup$ is a sum of two sets. Terminals set has been described as a sum of set $\mathbf{F}_{all}$ and an "empty" element which did not add any frequency to $\mathbf{F}$.

$$\mathbf{T} = \mathbf{F}_{all} \cup \{'\varnothing'\}. \tag{10}$$

where $\varnothing$ is an empty terminal that does not code any frequency.

There is a head and tail of a chromosome distinguishable in the body of chromosomes. There are no limits that would determine the length of the head. It is important, though, that the head is started with an element of the set $\mathbf{H}$. The rest of the head elements may belong to either $\mathbf{H}$ or $\mathbf{T}$ sets. The tail of chromosome may be built only with elements of the set of terminals. Their length is given by equation:

$$||tail|| = ||head|| \cdot (J-1) + 1 \tag{11}$$

where: $||tail||$ is a length of the tail; $||head||$ is a length of the head; J – is the maximum number of arguments of implemented functions from $\mathbf{H}$ set.

Such dependency is required to ensure the consistency of an ET (i.e. all branches of ET ending with a $\mathbf{T}$ set element).

In the presented works the $J = 2$ (a sum of two sets). In the Fig. 2 there is an exemplary GEP chromosome presented.

0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*	*1*	*2*
$\cup$	11	$\cup$	$\cup$	$\varnothing$	56	34	8	7	23	97	87	10
Head						**Tail**						

Fig. 2. An exemplary GEP chromosome

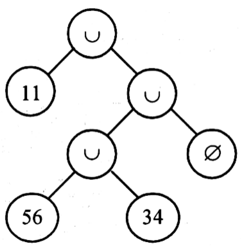

Fig. 3. A phenotype – Expression Tree – of the exemplary chromosome from the Fig. 2

Encoded expression tree in Fig. 3 presents the chromosomes from the Fig. 2 The set $\mathbf{F}$ described with the chromosome has 3 elements (frequencies): $\{$*11, 34, and 56*$\}$ [7].

In our research all individuals have been built with 4 chromosomes. The length of the chromosomes head has been fixed to 7. Therefore, one individual allows for coding up to 32 different frequencies. The population size equals 40 individuals.

3.2. Genetic operations. There is no significant difference between different selection methods. It is strongly advised to use a simple elitism in any GEP implementation [5, 6]. The elitism means copying the best (or few best) individual to the offspring population without modifying them. There are a few genetic operators used in GEP a mutation, crossovers, transpositions.

There is a uniform mutation implemented usually. It is important, though, to keep functions set element on the first chromosome position. Types of mutation implemented in our approach:

– function to frequency (and vice versa);
– function to empty terminal (as above);
– frequency to empty terminal (as above).

A crossover in GEP is not different from a crossover operator in a regular genetic algorithm. The crossover does not add new genetic material to the gene pool of the population. There are implemented 3 different types of the crossover a single point crossover, a two point crossover, and a chromosomes crossover.

A transposition operator causes copying a part of a chromosome of the certain length to the other position of the same chromosome. The transposition operators with mutation add a new genetic quality to the population's genetic pool. Transpositions that are used in GEP are an insertion sequence (IS) transposition, a root insertion sequence (RIS) transposition, and a chromosomes transposition.

It is important to stress a great ease of implementation of above operators [5–6].

3.3. Fitness function. A fitness function (FF) is the most important part of any EA application. As it was previously mentioned a FF choice determines a proper AE work. In the presented approach the task for FF was to find such a $\mathbf{F}$ set that the healthy circuit is separated (obligatory condition), find such a $\mathbf{F}$ set that the most hard faults are localized and a distance between each two of them is maximized, and find an $\mathbf{F}$ of the least possible power (additional condition) [7].

To cover above points the FF has been designed:

$$Q = 20(U+1) + 10P. \tag{12}$$

where: U – a number of correctly localized CUT states; P – a penalty modifier.

$$P = A\left(1 - e^{-\Delta_{\min}}\right) \tag{13}$$

$$\Delta_{\min} = \min_i \left(\Delta M_i \wedge f_i \in \mathbf{F}\right) \tag{14}$$

$$\Delta_{avg} = \frac{\sum_{i=1}^{M} (\Delta M_i \wedge f_i \in \mathbf{F})}{\|\mathbf{F}\|} \tag{15}$$

$$A = \begin{cases} Z \text{if} S_0 \text{localized} \\ -0.5 \text{if} S_0 \text{notlocalized} \end{cases} \tag{16}$$

$$Z = \begin{cases} 1.00 & \text{if} & K \leq 2 \\ 0.70 & \text{if} & K = 3 \\ 0.65 & \text{if} & K = 4 \\ 0.50 & \text{if} & 5 \leq K \leq 7 \\ 0.30 & \text{if} & 8 \leq K \leq 12 \\ 0.10 & \text{if} & 13 \leq K \end{cases} \tag{17}$$

where K is a number of used frequencies.

The A and Z values have been chosen heuristically based on the number of frequencies.

Fitness function given by Eq. (12) allows for fulfilling all of the set conditions. If the fault-free CUT is localized properly the number of **F** is decreasing and the distances between localized faults are maximized. If the intact CUT is not localized the number of **F** is increasing and the distances between localized faults minimized. Therefore, the more frequencies is used, the more likely to be separated state S_0 correctly [18].

4. Genetic Algorithm with Fuzzy Fitness Function System Based Approach

The GA was proposed by Holland in 1975 and in classical version it codes phenotypes binary. The GA imitates the natural processes of selection, recombination and succession in the population of individuals. It techniques is very useful to solve difficult problems of optimization in many fields and plenty of genotype structures has been used in modified GA systems [7–10].

Contrary to the classical set theory, for fuzzy sets (FS) an element may belong to the set with a partial value [11–13]. The fuzzy logic is modeled on the human reasoning that is not precise in many cases. This property of FS is very suitable to solve many problems for which classical discrete sets can not be used.

The GA has great optimization ability. However, the goal function is the weakest point of the optimization process. A great advantage of the weighted goal function is its monotonic character. On the other hand, the genetic algorithm simulates a real population behaviour. Moreover, the evaluation of an individual in the real world is much more complex than linear. Therefore, the hybrid system based on fuzzy and weighted fitness function has been proposed.

4.1. Evolutionary process diagram. The process diagram for evolutionary system can be sorted into two blocks and is presented in Fig. 4 The primary population for fuzzy initialization stage is created randomly with uniform probability and consists of I genotypes:

$$\mathbf{G}^n = \{\mathbf{E}_0, .., \mathbf{E}_{I-1}\}. \tag{18}$$

Next, fuzzy fitness function evaluates the quality of all phenotypes and its fitness value is compared to the best fitness found recently. If better solution has been found, the stagnations counter variable t is cleared and the new best genotype is stored (cycle with the best fitness progress, cycle with success). The variable t is incremented in case of fitness progress absence. In the next step genetic operations: reproduction, crossover, mutation and reduction are executed for parents randomly paired from a mating pool [7–8]. During succession a new population replaces the last one and the generation number n is incremented. Next, the fuzzy fitness is used to evaluate the new population and the cycle is repeated until maximum allowed values T_{mx} or N_{mx} are reached.

After the fuzzy optimization process is finished, random individuals from current population are replaced by all progressive ones (found and stored in all previous and successful cycles) and initial population $\mathbf{G}^n$ for the weighted system is created. Next, the evolution is continued for the weighted fitness function until N_{mx} is reached. The best phenotype found during evolution represents the solution.

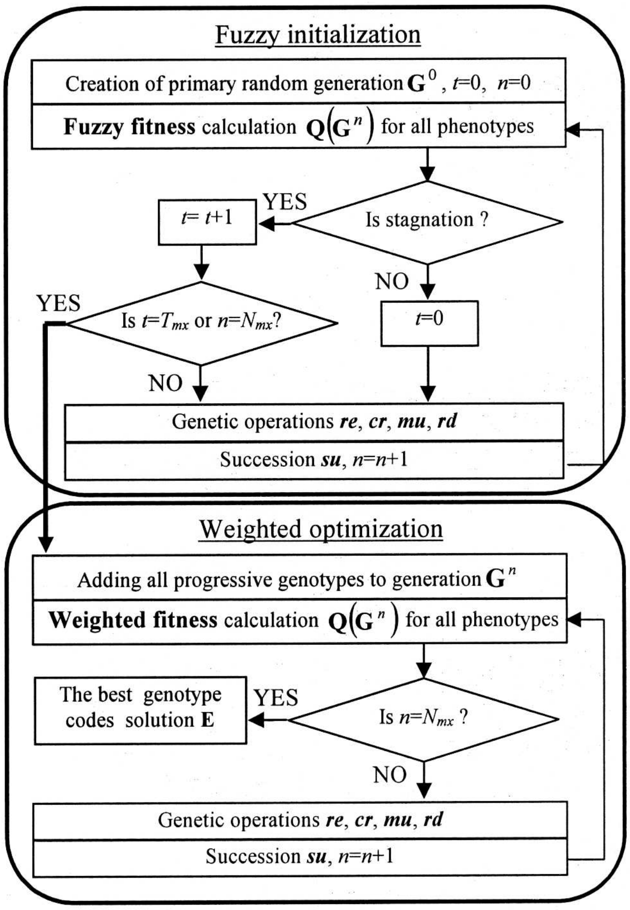

Fig. 4. The evolutionary process diagram

4.2. Phenotype coding. Each genotype **E** is a vector which contains K genes (integer numbers) and control value 0 at the end. Fig. 4 illustrates its structure. Control char is always equal 0 and designates the place of string termination. On the contrary to the classical version of GA, in the proposed algorithm the lengths of genotypes are adjusted during evolution. The number of genes is proportional to test frequencies set quantity and it can reach values from 2 (genes $\dddot{\Delta}_{ex}$ and $\dddot{f}_1$) up to K_{mx}. The allowed integer range for all genes is from 1 up to M, where M is the number of frequency steps of AC analyses assumed on the initial stage (see and compare 4, 5). The integer value of genes $\dddot{f}_1$ maps the real value of frequency f_k for adequate step of AC analysis (three dots above a variable denote a discrete value of the variable).

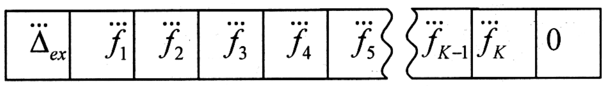

Fig. 5. Genotype structure

In the proposed system, gene $\dddot{\Delta}_{ex}$ is maximized and codes the value of relative extension coefficient Δ_{ex} for AS that can be calculated from:

$$\Delta_{ex} = \frac{\dddot{\Delta}_{ex}}{1000} \tag{19}$$

4.3. Genetic operations. For every cycle of evolution, system executes reproduction by means of rang method [7–8]. The rang is a integer number that designates the place of genotype in a quality ordered population. An individual with a rang r from population with the worst rang r_{last} is included to the mating pool with probability P_{re}:

$$P_{re} = 0.2 + 0.8 \cdot \left(1 - \frac{r}{r_{last}}\right) \tag{20}$$

During crossover, recombination for randomly selected genotypes is realized. Two kinds of crossover have been applied: discrete and averaging. The first one is fulfilled with probability P_{cr1} and is illustrated in the Fig. 6.

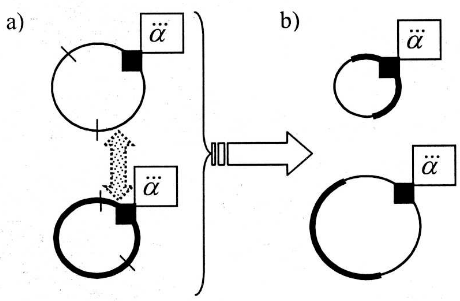

Fig. 6. The idea of discrete crossover: a) before, b) after recombination

At first, circles created by parent genotypes strings are randomly divided into four arcs. Then, substrings with extension coefficient $\dddot{\Delta}_{ex}$ genes are exchanged. Thanks to this kind of recombination, the probability of exchanging for each gene is the same and the size of genotype is regulated. During the second method of crossover that is executed with probability P_{cr2}, randomly selected parent genes are averaged:

$$\underset{i=1,\ldots,K_{\min}}{\vee} E^{(i)}_{offspring} = \frac{1}{2}\left(E^{(i)}_{1parent} + E^{(i)}_{2parent}\right) \quad \text{where } K_{\min} = \inf\left(\|E_{1parent}\|, \|E_{2parent}\|\right) \tag{21}$$

This recombination allows adjusting values of crossed genes. Next, genotypes can be modified during mutation process. The first kind of mutation is fulfilled for each gene with probability P_{mu1} and it replaces genes with the random ones. The second modification replaces full genotype string with the new one randomly generated with probability P_{mu2}. The last genetic modification of genotypes used in the system is called reduction and is executed with probability P_{rd}. During this process, randomly selected genes are deleted from the string. This operation impacts to the size of genotypes, so it reduces the power of test excitation set.

The new population is collected during succession from the offspring strings created after genetic modifications to intermediate mating pool. The elitary method [7–8] of succession has been applied, the best found genotype replaces the worst one.

4.4. Fuzzy fitness function. The fuzzy system is initially used to phenotypes evaluation. According to fuzzy set theory [11–12], the rule expert system has been designed. To evaluate fitness of the chromosome (an individual) IF – THEN rules have to be introduced. Typical Mamdani's IF-THEN rules [15] can be composed as follows:

$$\begin{gathered}\text{if } x_1 \text{ is } A_1^m \text{ and } x_2 \text{ is } A_2^m \ldots \\ \text{then } y^m \text{is} B^m, m = 1, 2, \ldots, M\end{gathered} \tag{22}$$

x_1, x_2 – input linguistic variables (LV); A_1^m, A_2^m – fuzzy sets of input LV; y^m – output LV; B^m – output fuzzy set.

The general structure of fuzzy rule base system is presented in the Fig. 7.

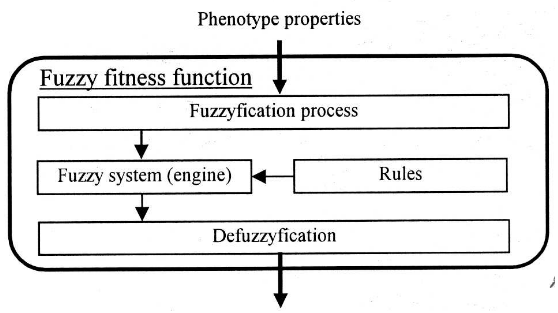

Fig. 7. The structure of fuzzy rule system

Each rule takes into account some premises and then produces a conclusion.

The fuzzyfication process considers the following properties of a phenotype: number U of states isolated in 100%

(maximized value), number of frequencies K (minimized value), and extension coefficient $\dddot{\Delta}_{ex}$ (maximized value).

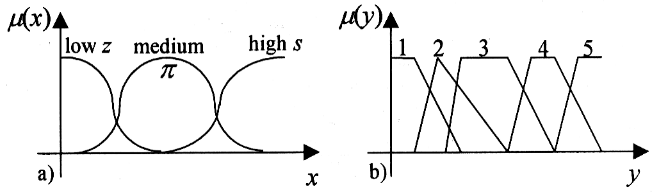

Fig. 8. The membership functions: a) of an input LV,b) of output LV

Based on the above data, LV have been chosen: detection (separation of S_0 state), location (other states separation), frequencies and extension. These LV are described by fuzzy sets: high, medium, low and determine z, π and s membership functions [12] illustrated in the Fig. 8a.

$$z\left(x; a_z, c_z\right) = \begin{cases} 1 & \text{for} \quad x \le a_z \\ 1 - 2\left(\dfrac{x - a_z}{c_z - a_z}\right)^2 & \text{for} \quad a_z \le x \le b_z \\ 2\left(\dfrac{x - c_z}{c_z - a_z}\right)^2 & \text{for} \quad b_z \le x \le c_z \\ 0 & \text{for} \quad x \ge c_z \end{cases}$$

$$s\left(x; a_s, c_s\right) = \begin{cases} 1 & \text{for} \quad x \le a_s \\ 2\left(\dfrac{x - a_s}{c_s - a_s}\right)^2 & \text{for} \quad a_s \le x \le b_s \\ 1 - 2\left(\dfrac{x - c_s}{c_s - a_s}\right)^2 & \text{for} \quad b_s \le x \le c_s \\ 0 & \text{for} \quad x \ge c_s \end{cases}$$

$$\pi(x; a_\pi, c_\pi, e_\pi) = \begin{cases} s(x; a_\pi, b_\pi, c_\pi) & \text{for} \quad x \le c_\pi \\ 1 - s(x; c_\pi, d_\pi, e_\pi) & \text{for} \quad x \ge c_\pi \end{cases} \tag{23}$$

where: $b_{z/s} = \dfrac{a_{z/s} + c_{z/s}}{2}$, $b_\pi = \dfrac{a_\pi + c_\pi}{2}$ and $d_\pi = \dfrac{c_\pi + e_\pi}{2}$.

Universe of Discourse (UD) depends on linguistic variable (characteristic points of fuzzy sets limits for the example problem from section VI in brackets):

1. Detection (UD: 0 – 33):
   a. High $s(x; 29, 33)$
   b. Medium $\pi(x; 12, 16, 32)$
   c. Low $z(x; 1, 15)$
2. Location (UD: 0 – 33):
   a. High $s(x; 12, 32)$
   b. Medium $\pi(x; 6, 12, 16)$
   c. Low $z(x; 18)$
3. Frequencies (UD: 1 – 25):
   a. High $s(x; 1, 8)$
   b. Medium $\pi(x; 7, 12, 14)$
   c. Low $z(x; 10, 25)$
4. Extension (UD: 1 – 4000):
   a. High $s(x; 900, 1200)$
   b. Medium $\pi(x; 400, 700, 1000$
   c. Low $z(x; 1, 900)$

There is only one output LV – fitness function and 5 fuzzy sets, illustrated in the Fig. 8b:

1. very low (reverse gamma) $\Gamma^{-1}\left(x; 0.1, 0.25\right)$
2. low (triangle) $t(x; 0.1, 0.25, 0.4)$;
3. medium (trapezoidal) $tr(x; 0.2, 0.45, 0.55, 0.7)$;
4. high (triangle) $t(x; 0.6, 0.75, 0.85)$;
5. very high (gamma) $\Gamma(x; 0.75, 1)$.

The proposed approach uses Mamdani's inference engine and COG (Center Of Gravity) method [13, 15] to obtain final output result Q. The centre of gravity is the average location of the weight of an object. An example of COG method is presented in the Fig. 9.

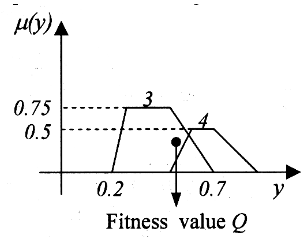

Fig. 9. Defuzzyfication method

The fuzzy inference system contains a number of rules and all of them are printed in the Tab. I. One can notice that not all possible combinations of rules have been created. It comes from the fact that other (not listed) rules do not influence on the best chromosome fitness value.

Table 1
The Fuzzy System Rules

IF (Premises)	1 c	1 b	1 a	1 a	1 a	1 a	1 a	1 a	1 a	1 a	1 a	1 a
	1 c	X	2 c	2 b	2 a	2 c	2 b	2 a	2 c	2 b	2 a	2 a
	X	X	3 c	3 c	3 c	3 b	3 b	3 b	3 a	3 a	3 a	3 a
	X	X	X	X	X	X	X	X	X	X	4 b	4 a
THEN (Conclusion)	1	1	1	1	2	1	1	4	2	3	4	5

4.5. Weighted fitness function. The weighted method of fitness calculating has been finally used. The used function Q (4.7) consists of three parameters for optimization: the number of fully separated states U, the number of excitations K and the extension coefficient $\dddot{\Delta}_{ex}$:

$$Q = w_1 \cdot (L - U) + w_2 \cdot K + w_3 \cdot \left(\dddot{\Delta}_{ex}\right) \tag{24}$$

L – the power of states set **S**; D *- the maximum possible integer value for gene $\ddot{\alpha}$ ($M = D$ has been assumed).

The weights w_1, w_2, w_3 allow to control the evolution process and can be calculated to achieve assumed hierarchy of optimization. The proposed method for weights calculating is based on discrete character of UD. For each optimized parameter (U, K, $\dddot{\Delta}_{ex}$) the minimal possible step (quant) and the maximum possible value can be designated. The weight w_1 controls full separation level and it should assure the highest: quant and the maximum possible value for section 1 (the highest priority of optimization). The weights w_2 and w_3 control the number of excitations and the size of extension coefficient adequately and they should assure that the maximum possible values of section 2 and 3 are smaller than the quant of the previous section (the medium and the lowest priority). To achieve described hierarchy for the proposed system, weights are calculated from equations given below:

$$w_1 = \frac{1}{L}; \;\; w_2 = \frac{1}{w_1 \cdot K_{mx}}; \;\; w_3 = \frac{1}{w_2 \cdot M} \tag{25}$$

Contrary to fuzzy fitness, the weighted fitness describes the phenotype unequivocally and precisely and it allows to create the best final solution from initial population of well fuzzy evaluated individuals [19].

5. Simulated Annealing with Fuzzy Fitness Function System Based Approach

Simulated Annealing (SA) optimization algorithm is applied to search minimum number of excitations in the third approach. The name comes from annealing in metallurgy, a technique of controlled cooling of a material to reduce their defects [15]. It is well known method for the global optimization problem in a large space search. SA algorithm belongs to the heuristic methods.

5.1. Simulated Annealing Algorithm. Simulated annealing is a random-search technique which exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system; it forms the basis of an optimization technique for combinatorial and other problems [16]. The method is developed by Kirkpatrick, Gelatt and Vecchi in 1983. Procedure of the SA algorithm is presented in Fig. 10. Each step of the SA considers a neighbour (*move*) of the current state $\bar{F}$. If new state energy is less than previous one, the previous state $\bar{F}$ is replaced by current one (with *move*) $\bar{F}'$.

Otherwise, a worse state can be accepted with probability p_c.

$$p_C\left(\overline{F}\right) = b\left[f\left(\overline{F}'\right) - f\left(\overline{F}\right), T\right] = \exp\frac{-\left[f\left(\overline{F}'\right) - f\left(\overline{F}\right)\right]}{kT} \tag{26}$$

Three elementary *moves* have been introduced [17]: adding, removing, and swapping a single frequency from the assumed range of frequency $f_{\min}$ and $f_{\max}$.

Obviously, the set **F** cannot contain chosen frequency more than once.

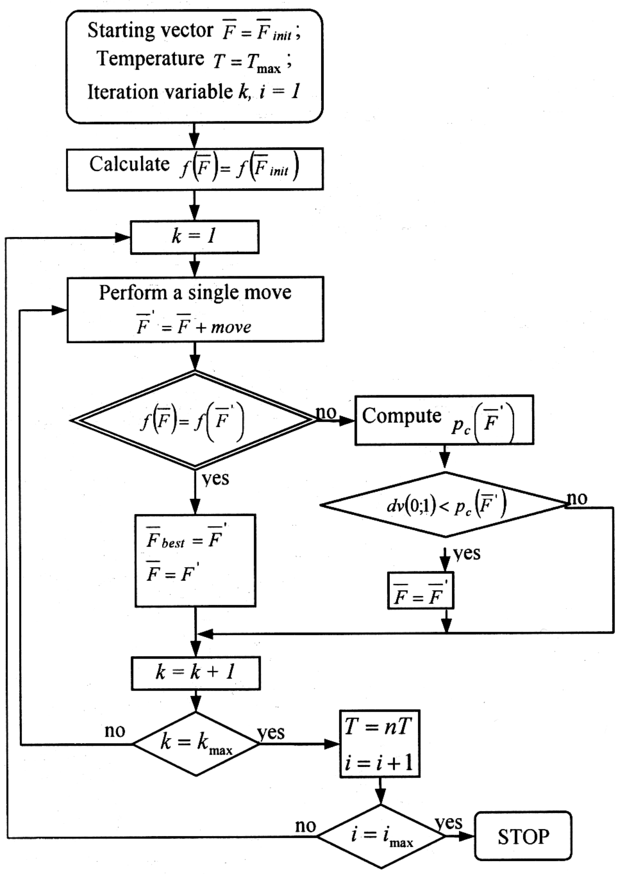

Fig. 10. Simulated Annealing workflow

Quality of heuristic optimization algorithms are linked with energy (fitness, goal) function which evaluates current solution. Originally, the weighted evaluating function is introduced, and weighted coefficients are related to significance of optimized parameters. As a new approach to the analog circuit diagnosis we have applied the fuzzy fitness system and compared to the weighted goal one.

5.2. Initialization of the process. The algorithm is initialized by a random frequencies vector which consists of 15 data. Obviously, a frequency on the list cannot be repeated ($\boldsymbol{F_{init}}$).

5.3. Fuzzy Fitness Function. The fuzzy fitness function used in simulated annealing system is identical as previously described in Section 4.4.

5.4. Weighted fitness function. Typical SA strategy operates with weighted energy function. Therefore, we decide to introduce one for the diagnosis problem. The optimization process considers the following input data:

1. Number of states separated from S_0 (detection rate – U_d) – parameter is maximized. If at least one frequency in the vector separates a state from S_0, then the state is isolated.
2. Number of other states isolated in 100% (location rate – U_l) – maximized parameter. Regardless S_0 state, if a single frequency separate two different states.

3. Number of frequencies K (minimized parameter) is being used to get the highest detection and location rate.

A fitness value Q is calculated from the formula:

$$Q = w_1 \cdot U_d + w_2 \cdot U_l + w_3 \cdot \frac{1}{K} \tag{27}$$

where w_1, w_2, and w_3 are coefficients chosen empirically. For the example circuit $w_1 = \frac{1}{32}$, $w_2 = \frac{1}{320}$, $w_3 = 0.01$; So, $\frac{1}{w_1}$ is number of detected faults – the most important subpart of the formula. The second component is 10 times less, and the number of frequencies is the least important part.

As can be seen the fitness function is maximized. The maximum value $Q = 1.11$, and it is produced if all states are isolated (32), and detected (32) by a single frequency.

5.5. Stop criterion. The algorithm is stopped if the optimal solution is achieved and one of the following criterions is satisfy:

1. Maximum number of iterations is reached.
2. There is no improvement in the next 10000 iteration.

During the test stage, all frequencies from the predefined range are applied, and the global solution is determined. It gives the greatest information about the CUT by answering on the following questions: how many states can be isolated (detection of a fault) from non faulty circuit and how many other states are separated from another faulty state (location of a fault) [20].

6. Exemplary circuit diagnosis

The presented methods have been tested with the use of an exemplary circuit (Fig. 11). The source reference of the circuit is Ref. [1]. There were 33 states of CUT assumed (intact circuit and 32 hard faults). The number of set **F** was 4000 frequencies ($f_{\min} = 100$ [Hz], $f_{\max} = 1$ [MHz]). There are amplitude characteristics of all CUT states presented in Fig. 12. In Fig. 13 there is a computed ambiguity region presented (see Eq. 8).

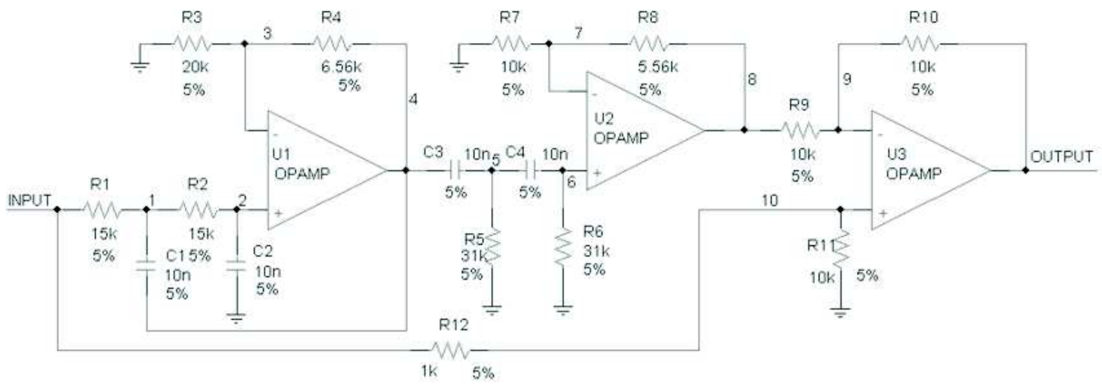

Fig. 11. The diagnosed exemplary circuit

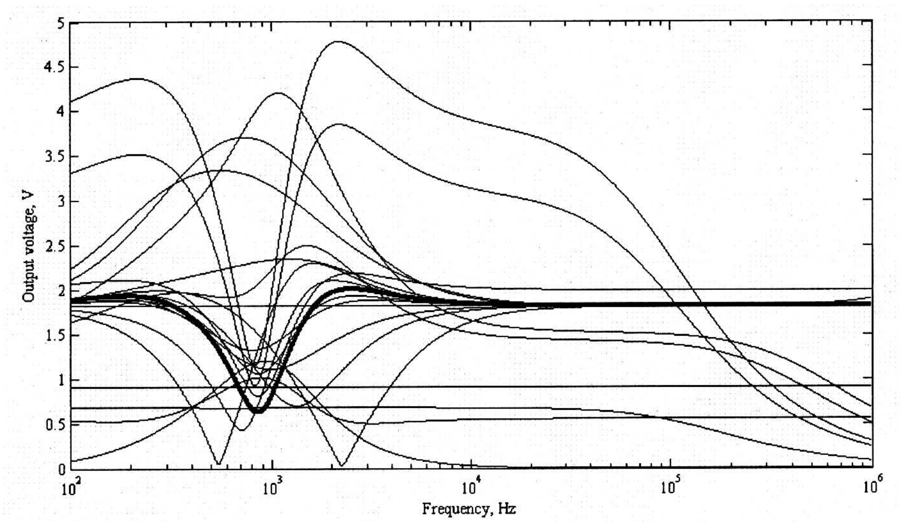

Fig. 12. The intact and damaged CUT amplitude characteristics

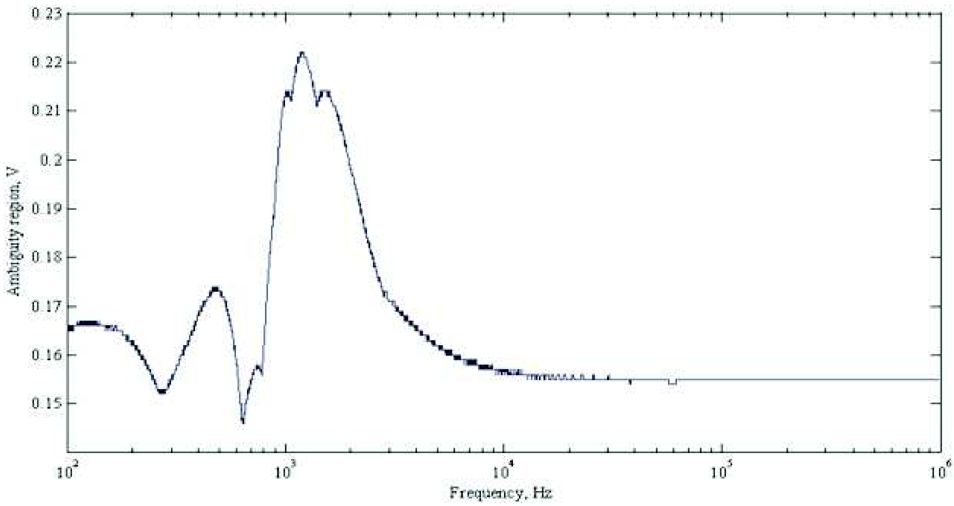

Fig. 13. The computed ambiguity region

Short elements have been simulated by the parallel conductance of 10 [S] and open elements with serial resistance of 1 [TΩ].

The global solution (with the use of all possible frequencies) allowed for location of 15 hard faults. The rest of faults were grouped into 6 ambiguity set (look Table 3).

Table 2
The results obtained for heuristic systems

Parameter	System		
	GEP	GA	SA
U	15	15	15
K	2	2	2
f_1, $\Delta_{\min}$	592.9 Hz, 160 mV	998.0 Hz, 298 mV	628.1 Hz, 149 mV
f_2, $\Delta_{\min}$	2.951 KHz, 171 mV	3.936 KHz, 156 mV	6.370 KHz, 160 mV

The results obtained with each of the presented systems are gathered in the Table 2. All of the algorithms allowed for finding solutions with the same level of detectability as with the use of all possible frequencies (Table 3). Moreover, the number of frequencies in each case was 2. The set of fully separated states contains state S_0 of healthy circuit, so 100% level of fault detection is possible (test go/no go). Fault location and identification are precise for 44% states.

7. Conclusions

Heuristic methods for sinusoidal stimuli test selection have been proposed. The algorithms allow for diagnosing a CUT with a single accessible node. The implementation of ambiguity sets has made the methods robust to a CUT tolerances and practical inaccuracies (e.g. test measure errors, simulation models imprecise). If diagnosis rate is not satisfactory it allows introduce either other node(s) or additional frequencies. All methods reduce analogue fault dictionary significantly where only a few signatures (amplitude responses) have to be stored. The nearest neighbour measure has been tested but in case of other artificial dictionary construction, diagnostic results may be even higher.

Table 3
Diagnostic results for stimuli found by hybrid system

CUT state	Recognized state(s)	No. of isolated states
S_0 (healthy)	S_0	**32**
S_1 (R_1 short)	S_1	**32**
S_2 (R_2 short)	S_2	**32**
S_3 (R_3 short)	S_3	**32**
S_4 (R_4 short)	$S_4S_8S_{16}S_{19}S_{21}S_{23}$	27
S_5 (R_5 short)	$S_5S_6S_{13}S_{14}S_{17}S_{18}S_{31}S_{32}$	25
S_6 (R_6 short)	$S_5S_6S_{13}S_{14}S_{17}S_{18}S_{31}S_{32}$	25
S_7 (R_7 short)	S_7	**32**
S_8 (R_8 short)	$S_4S_8S_{16}S_{19}S_{21}S_{23}$	27
S_9 (R_9 short)	S_9	**32**
S_{10} (R_{10} short)	S_{10}	**32**
S_{11} (R_{11} short)	$S_{11}S_{28}$	31
S_{12} (R_{12} short)	$S_{12}S_{27}$	31
S_{13} (C_1 short)	$S_5S_6S_{13}S_{14}S_{17}S_{18}S_{31}S_{32}$	25
S_{14} (C_2 short)	$S_5S_6S_{13}S_{14}S_{17}S_{18}S_{31}S_{32}$	25
S_{15} (C_3 short)	S_{15}	**32**
S_{16} (C_4 short)	$S_4S_8S_{16}S_{19}S_{21}S_{23}$	27
S_{17} (R_1 open)	$S_5S_6S_{13}S_{14}S_{17}S_{18}S_{31}S_{32}$	25
S_{18} (R_2 open)	$S_5S_6S_{13}S_{14}S_{17}S_{18}S_{31}S_{32}$	25
S_{19} (R_3 open)	$S_4S_8S_{16}S_{19}S_{21}S_{23}$	27
S_{20} (R_4 open)	S_{20}	**32**
S_{21} (R_5 open)	$S_4S_8S_{16}S_{19}S_{21}S_{23}$	27
S_{22} (R_6 open)	S_{22}	**32**
S_{23} (R_7 open)	$S_4S_8S_{16}S_{19}S_{21}S_{23}$	27
S_{24} (R_8 open)	S_{24}	**32**
S_{25} (R_9 open)	S_{25}	**32**
S_{26} (R_{10} open)	S_{26}	**32**
S_{27} (R_{11} open)	$S_{12}S_{27}$	31
S_{28} (R_{12} open)	$S_{11}S_{28}$	31
S_{29} (C_1 open)	S_{29}	**32**
S_{30} (C_2 open)	S_{30}	**32**
S_{31} (C_3 open)	$S_5S_6S_{13}S_{14}S_{17}S_{18}S_{31}S_{32}$	25
S_{32} (C_4 open)	$S_5S_6S_{13}S_{14}S_{17}S_{18}S_{31}S_{32}$	25

The proposed methods have been tested with the use of the exemplary circuit (Fig. 11) with good results (equal for each of the systems). The algorithms' assets are: an ease of implementation, a short proceeding time and a self-adapting optimal frequencies' set. It may suggest that there is a sense of developing industrial applications based on presented heuristic algorithms.

Acknowledgements. This work was supported by the Ministry of Science and Higher Education under grant no 3 T11B031 29.

REFERENCES

[1] M. Catelaniand and A. Fort, "Soft fault detection and isolation in analog circuits: some results and a comparison between a fuzzy approach and radial basis function networks", *IEEE Trans. on Instrumentation and Measurement* 51(2), 196–202 (2002).

[2] L. S. Milor, "A tutorial introduction to research on analog and mixed-signal circuit testing", *IEEE Trans. on Cir. and Syst.-II, Analog and Dig. Sig. Proces.* 45(10), 1389–1407 (1998).

[3] Pen-Min Lin and Y.S. Elcherif, "Computational approaches to fault dictionary", in *Analog Methods for Computer-Aided Circuit Analysis and Diagnosis*, pp. 325–364, M.Dekker, New York, 1998.

[4] W. Hochwald and J.D. Bastian, "A DC dictionary approach for analog fault dictionary determination", *IEEE Trans. on Cir. and Syst.* 26, 523–529, (1979).

[5] C. Ferreira, *Gene Expression Programming: Mathematical Modelling by an Artificial Intelligence*, ISBN: 9729589054.

[6] C. Ferreira, "Gene expression programming: a new adaptive algorithm for solving problem", *Complex Systems* 13(2), 87–129 (2001).

[7] P. Jantos, D. Grzechca, T. Golonek, and J. Rutkowski, "Gene expression programming-based method of optimal frequency set determination for purpose of analogue circuits' diagnosis", *V Int. Conf. on Computer Recognition Systems 2*, CORES'07, 794–801 (2007).

[8] D.E. Goldberg, *Genetic Algorithms in Search/Optimization and Machine Learning*, Addison Wesley, London, 1989.

[9] Z. Michalewicz, *Genetic Algorithms+Data Structures= Evolution Programs*, Springer-Verlag, Berlin, 1996.

[10] T. Golonek and J. Rutkowski, "Genetic-algorithm-based method for optimal analog test points selection", *IEEE Trans. on Cir. and Syst.-II.* 54(2), 117–121 (2007).

[11] T. Golonek, D. Grzechca, and J. Rutkowski, "Evolutionary system for analog test frequencies selection with fuzzy initialization", *IEEE Design and Diagnostics of Electronic Circuit and Systems* DDECS 1, 353–356 (2007).

[12] M.H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis", *IEEE Transaction on Computers* C-26 (12) 1182–1191 (1977).

[13] L.A. Zadeh, "Fuzzy sets", *Information and Control* 8, 338–353 (1965).

[14] L. A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages", *Computers and Mathematics* 9, 149–184 (1983).

[15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing", *Science* 220(4598), 671–680 (1983).

[16] L. Ingber, "Simulated annealing: practice versus theory", *Mathl. Comput. Modelling* 18(11),. 29–57 (1993).

[17] D. Grzechca and T. Golonek, "Simulated annealing with fuzzy fitness function for test frequencies selection", *IEEE Conference on Fuzzy Systems, FUZZ-IEEE 2007 1*, 438–443 (2007).

[18] P. Jantos, D. Grzechca, T. Golonek, and J. Rutkowski, "Optimal test frequencies choice with the use of Gene Expression Programming", *VI State Conf. on Electronics 1*, 143–148 (2007), (in Polish).

[19] T. Golonek, D. Grzechca, and J. Rutkowski, "Evolutionary method for analog test frequencies selection with fuzzy initialization", *VI State Conf. on Electronics 1*, 101–106 (2007), (in Polish).

[20] D. Grzechca, T. Golonek, and J. Rutkowski, "The use of simulated annealing to optimal excitation selection of analogue electronic circuits", *VI State Conf. on Electronics 1*, 83–88 (2007), (in Polish).