Optimization Process by Generalized Genetic Algorithm

ALEXANDER ZEMLIAK[1], ANDREI OSADCHUK[2], CHRISTIAN SERRANO[3]
[1]Department of Physics and Mathematics,
Autonomous University of Puebla,
MEXICO

[2]The Arctic University Museum of Norway,
The Arctic University of Norway, Tromsø,
NORWAY

[3]Department of Electronics,
Autonomous University of Puebla,
MEXICO

Abstract: - The approach developed earlier, based on generalized optimization, was successfully applied to the problem of designing electronic circuits using deterministic optimization methods. In this paper, a similar approach is extended to the problem of optimizing electronic circuits using a genetic algorithm (GA) as the main optimization method. The fundamental element of generalized optimization is an artificially introduced control vector that generates many different strategies within the optimization process and determines the number of independent variables of the optimization problem, as well as the length and structure of chromosomes in the GA. In this case, the GA forms a set of populations defined by a fitness function specified in different ways depending on the strategy chosen within the framework of the idea of generalized optimization. The control vector allows you to generate different strategies, as well as build composite strategies that significantly increase the accuracy of the resulting solution. This, in turn, makes it possible to reduce the number of generations required during the operation of the GA and reduce the processor time by 3–5 orders of magnitude when solving the circuit optimization problem compared to the traditional GA. An analysis of the optimization procedure for some electronic circuits showed the effectiveness of this approach. The obtained results prove that the applied modification of the GA makes it possible to overcome premature convergence and increase the minimization accuracy by 3-4 orders of magnitude.

Key-Words: - generalized optimization, GA, circuit optimization, control vector, set of strategies, premature convergence.

Received: August 15, 2023. Revised: February 11, 2024. Accepted: March 9, 2024. Published: April 18, 2024.

1 Introduction

An effective way to solve problems of design and synthesis of electronic systems is to use optimization procedures. These procedures are a set of iterative algorithms that make it possible to obtain the required characteristics of the designed system by minimizing some specially developed objective functions.

Genetic algorithm (GA), which belongs to the family of stochastic algorithms and is based on a set of operators inspired by biological processes in nature (mutation, crossbreeding, and selection), has been actively used over the past two decades to find highly accurate algorithms for solving optimization problems. One area in which the genetic algorithm is widely used is the computer-aided design of electronic circuits, [1], [2], [3] , which allows one to analyze the principles, as well as the strengths and weaknesses of the GA when used to determine circuit parameters. The design of a bipolar transistor, CMOS op-amps, a CMOS op-converter, and a matching network are demonstrated here as examples. The use of GA for the automated design of analog circuits using the example of a half-wave rectifier and a bandpass filter is shown in [4]. Some interesting ideas are presented in [5] on the use of GA optimization in discrete element modeling. Modification of the topology of circuits aimed to avoid invalid circuits allowed to get improved GA efficiency, [6]. Optimization of the sizes of analog circuits was performed in [7], using GA and in [8], using evolutionary algorithms. The benefits of combining GA and particle swarm optimization (PSO) in the design of analog ICs with practical user-defined specifications are demonstrated in [9], [10]. Rule-guided genetic algorithm with a special mutation mechanism for narrow solution region cases is offered in [11].

One of the known disadvantages of GAs is premature convergence. It lies in the fact that the GA offers, as the best solution over a sufficiently long sequence of generations, an individual that provides a certain local minimum of the objective function, without further movement to the global minimum. Various ways to solve this problem have been proposed. One of them is inherent in the very design of the algorithm: the ability to regulate the probability of mutation. A combination of a dynamic genetic clustering algorithm and an elitist method is considered in [12]. In [13], a hybrid crossover method was proposed that improves convergence and at the same time maintains high quality of retrieved documents in information retrieval systems. It is based on the use of single-point crossing of ordered chromosomes; the result of this method is a daughter chromosome that combines the best genes of both parent chromosomes. The problem of premature convergence was analyzed in the framework of a Markov chain in [14], [15]. The authors prove that the degree of diversity of populations tends to zero with probability 1, causing a decrease in the searchability of the GA and, consequently, premature convergence. The relationship between premature convergence and GA parameters is shown. In [16], two mechanisms were proposed to avoid premature convergence of GAs: dynamic application of genetic operators based on average progress and partial re-initialization of the population. The authors of [17], propose a combination of the frequency crossover strategy with nine different mutation strategies to reduce the effect of premature convergence using the traveling salesman problem as an example. In [18], to overcome premature convergence, an improved adaptive GA is considered, containing dynamic adjustment of crossover and mutation operators during the evolution process, as well as a restart strategy. Other aspects of the problem of premature convergence in GA were discussed in papers, [19], [20].

One of the ways to eliminate premature convergence of GA is to use the generalized optimization methodology, [21]. This approach was created to improve the efficiency of deterministic optimization algorithms. Attempts to apply this methodology to GA have shown its effectiveness in overcoming premature convergence, [22]. Even a very limited set of strategies generated with this approach not only improved the processor time and the number of generations required to obtain the required high-precision results in achieving a given operating point of the circuit. In several cases, the use of a generalized methodology made it possible to achieve such high-quality results where the use of a separate GA did not provide the necessary convergence in principle.

One of the most important aspects of the concept of this generalized methodology is that it generates a huge number of possible optimization strategies, which naturally involves selecting the best one. This article is devoted to the study of the structural basis of strategies and some categories of composite strategies within the framework of a generalized methodology in its synthesis with GA using examples of specific electronic circuits.

The rest of the paper is organized as follows. Section 2 describes the principles for solving nonlinear programming problems using a general optimization methodology, taking into account their adaptation to GA as the main optimization method. In Sections 3 and 4, we consider solving electronic circuit optimization problems using various structural basis strategies and composite strategies. This is followed by an analysis and discussion of the results.

2 Generalized Optimization with Genetic Algorithm

We define optimization of an electronic circuit as the task of minimizing the objective function *C(X)*, $X \in R^N$. Constraints are formed using circuit model equations based on Kirchhoff's laws:

$$g_j(X) = 0,\ j = 1,2,...,M\ . \qquad (1)$$

We divide the components of vector *X* into two groups: X' and X'', $X' \in R^K$, $X'' \in R^M$, X' is the vector of independent variables, X'' is the vector of dependent variables, *K* and *M* are corresponding numbers of independent and dependent variables, $K + M = N$. The objective function is minimized using an optimization procedure, which has the following vector form:

$$X^{s+1} = \Lambda(X^s),\ s = 1,2,...\ , \qquad (2)$$

where s is the iteration number, Λ is the operator of transition from step s to step (s+1). This last operator depends on the objective function *C(X)*.

The classical approach to the constrained optimization problem involves solving system (1) at each step of the iterative process. We can call this a traditional optimization strategy (TOS). There is no need to solve the system of Kirchhoff's law equations at each iteration step of the optimization procedure. The conditions imposed by system (1) must be satisfied at the end of the optimization procedure for the solution to satisfy the problem. This formulation frees us from the mandatory division of variables into independent and dependent when solving the problem (1) – (2). This idea underlies the generalized optimization methodology and is implemented through a generalized objective function, including a so-called penalty function. This structure of the objective function allows both to achieve minimization of the original objective function *C(X)* and to ensure the fulfillment of Kirchhoff's laws at the final stage of the optimization procedure. This approach to circuit optimization can be called modified traditional optimization strategy (MTOS), and it corresponds to extracting all the equations from the circuit model. The term "generalized" means that the extraction of not all equations from the circuit model is considered, but only parts of them. When starting to implement the idea of generalized optimization, we declare all components of the vector X to be independent. However, to preserve the physical meaning of the problem and fulfill the corresponding constraints at the endpoint of the optimization process, it is necessary to introduce a new objective function into consideration:

$$F(X) = C(X) + \varphi(X) \tag{3}$$

Here *φ(X)* – the penalty function which goes to zero at the end of the optimization procedure which is equivalent to the fulfillment of (1) at this point. The form of the penalty function is:

$$\varphi(X) = \sum_{j=1}^{M} g_j^2(X). \tag{4}$$

The upper limit in the sum is equal to *M*, which corresponds to MTOS. This limit does not have to be equal to *M*, it can be chosen as *Z*, $0 \le Z \le M$. In this case, we move on to the generalized approach to the optimization process when the amount of dependent variables which are declared as independent is arbitrary from the interval [0, M]. Then we exclude not all equations from the circuit model (1) but only a part of them and the amount of excluded equations is *Z*. This change is described with an additional control vector *U* introduced. The value of this vector defines how the structure of the main system changes and, thus, how processor costs are redistributed between blocks of the circuit analysis and optimization. This redistribution is the ground for the reduction of the processor time in the optimization process. The introduction of the vector $U=(u_1, u_2, \ldots, u_M)$ modifies the system (1) as follows:

$$(1-u_j)g_j(X) = 0,\ j = 1,2,...,M \tag{5}$$

where u_j components of *U* can take the value 0 or 1. The number of different strategies in this case forms the structural basis and is equal to 2^M. Expressions (3) and (4) will look out as:

$$F(X,U) = C(X) + \varphi(X,U) \tag{6}$$

$$\varphi(X,U) = \frac{1}{\sigma}\sum_{j=1}^{M} u_j g_j^2(X) \tag{7}$$

The tuning parameter σ may depend on the mathematical model of the analyzed circuit, and on the chosen optimization method, and in our examples it is equal to 1. The optimization process operator (2) also depends on the new objective function *F*(*X*, *U*), and therefore on the control vector *U*:

$$X^{s+1} = \Lambda(X^s, U), s = 1,2,..., \tag{8}$$

The vector *U* controls the structure of the circuit model equations and the optimization procedure as follows: $u_j = 0$ – the *j*th equation in (5) remains in the system, the correspondent element $g_j^2(X)$ in the sum on the right side of (7) is removed, $u_j = 1$ – the *j*th equation in (5) is excluded from the system, the $g_j^2(X)$ in the right side of (7) remains there. If all components of *U* are equal to 0, then system (5) is identical to (1), and we obtain TOS when all circuit model equations must be solved at each step of the optimization process. In the case when all u_j are equal to 1, system (5) disappears, and all information from it is transferred to the penalty function on the right side of (6). This is MTOS.

The GA crossover is organized as follows. The genes that will participate in the upcoming crossover are determined by random selection. A two-point crossover is organized in each of them. Thus, about the entire chromosome, such a scheme is a multipoint crossover with a variable number of separation points. To a certain extent, this approach was caused by the "unequal" position of initially independent genes and "new" independent genes arising as a result of extracting equations from the circuit model. The first are constantly present in the chromosome, the presence of the latter varies depending on the strategy used at the current stage of the iterative process. Therefore, it seems necessary to take special care in organizing a high-quality crossover in the second, variable group of genes. Another factor that determined the choice of this option was the desire to increase the variability of the set of chromosome regions involved in crossover, both in their number and in their location relative to each other.

3 Analysis and Discussion

When analyzing examples, the chromosome length in the algorithm varied from 12 to 20 for each variable. The number of chromosomes in the population varied from 100 to 500.

3.1 Example 1

Minimize *C(X)*

$$C(X) = 2x_1^2 - 3x_2^2 - 2x_1 \quad (9)$$

subject to:

$$(x_1 - 3)^2 + (x_2 - 2)^2 = 0 \quad (10)$$

In this example, there is only one independent variable x_1 (K=1), and parameter M=1 because there is only one constraint equation (10). Let's define variable x_2 as dependent which can be calculated from equation (10).

There is an analytical solution to this problem. Indeed, the fulfillment of the necessary constraint (10) is ensured by the solution of this equation and is achieved at the point $x_1 = 3$, $x_2 = 2$. At this point, the goal function $C(X)$ takes the minimum zero value. These values are the solution to the problem. Let us find, however, this solution by the developed approach.

Based on the generalized approach, equation (10) is transformed into the following equation:

$$(1-u)\left((x_1 - 3)^2 + (x_2 - 2)^2\right) = 0 \quad (11)$$

where u is the component of the control vector U, in this case, the only one.

Consider two main strategies: TOS which has a control vector U=(0) and MTOS which has a control vector U=(1).

Here, we analyze the results of optimization using a GA for these strategies. However, it was shown that in the case of a deterministic optimization process, a combination of several strategies can reduce both the number of steps of the optimization process and the computation time.

Table 1 shows the dynamics of changes in the number of generations and processor time (s) of the GA depending on the required precision δ of minimizing the objective function F for three strategies: TOS, MTOS, and composite strategy (0)(1) with an optimal switching point Sp from strategies (0) to strategy (1). The optimal switch point Sp improves the characteristics of the composite strategy, but in this paper it was obtained manually.

The optimal value of the switching point Sp was obtained by additional analysis. This value, as can be seen from the table, depends on the required precision δ. It is clear that when using the TOS, the number of generations and CPU time is less than for MTOS up to a certain level of precision (10^{-4}).

Table 1. Dynamics of changes in the number of generations and processor time (s) of the GA depending on the required precision δ of function F for three strategies: TOS, MTOS, and composite strategy (0)(1) with the optimal switching point Sp

	G (CPU time (s))		
δ	U=(0)	U=(1)	U=(0)(1)
10^{-1}	11 (0.048)	56 (0.125)	16 (0.044) Sp=10
10^{-2}	18 (0.091)	69 (0.143)	24 (0.059) Sp=9
10^{-3}	23 (0.073)	79 (0.162)	27 (0.066) Sp=10
10^{-4}	37 (0.113)	91 (0.182)	42 (0.095) Sp=13
10^{-5}	32883 (100.2)	96 (0.191)	53 (0.113) Sp=9
10^{-6}	-	104 (0.206)	66 (0.146) Sp=14
10^{-7}	-	114 (0.261)	75 (0.176) Sp=6
10^{-8}	-	121 (0.271)	81 (0.179) Sp=14
10^{-9}	-	129 (0.275)	104 (0.226) Sp=15
10^{-10}	-	134 (0.278)	104 (0.227) Sp=15
$5.1\ 10^{-12}$	-	1368 (2.906)	117 (0.254) Sp=10
10^{-12}	-	-	121 (0.266) Sp=10
$2.6\ 10^{-14}$	-	-	989 (1.928) Sp=5

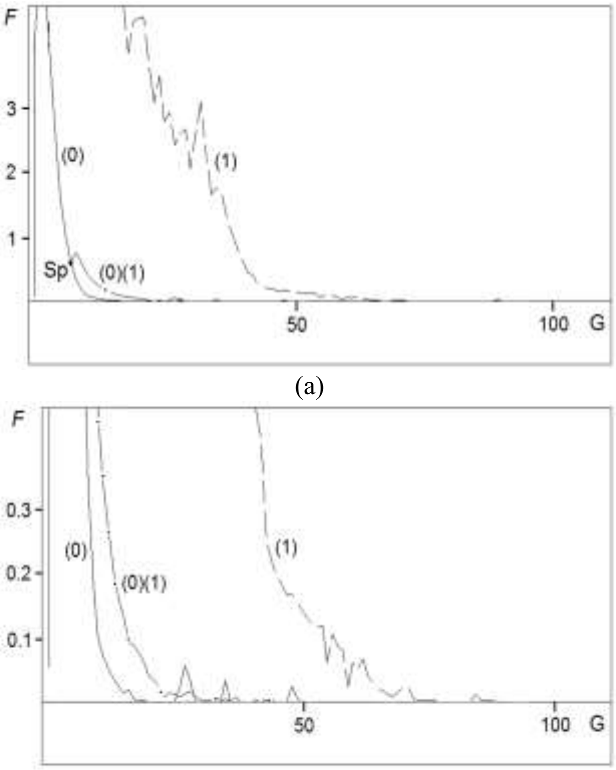
The TOS allows finding a solution up to the error level of 10^{-5}, but the number of generations increases dramatically. At the same time, this strategy cannot find a solution with higher accuracy. The MTOS with control vector (1) finds a solution with a much higher accuracy up to $5.1 \cdot 10^{-12}$.

At the same time a composite strategy consisting of two, (0) and (1) with an optimal switching point between them, gives a solution with an accuracy of $2.6 \cdot 10^{-14}$ and, importantly, with a smaller number of generations.

Figure 1(a) and 1(b) show the dependence of the generalized objective function F under successive generational change for strategies (0), (1), and composite strategy (0)(1) for two scales; (a) - scale 1, (b) – scale 2.

The best strategy for minimizing the fitness function is the composite strategy (0)(1), which, in the case of the optimal switching point Sp, solves the problem in the best way compared to other strategies.

Variables x_1 and x_2 take the values 3 and 2, respectively, but with different degrees of accuracy for different strategies.



(a)

(b)

Fig. 1: Dependence of the generalized objective function *F* under successive generational change for strategies (0), (1) and composite strategy (0)(1) for two scales; (a) - scale 1, (b) – scale 2

3.2 Example 2

Minimize $C(X)$

$$C(X) = (x_3 - 0.15)^2 \quad (12)$$

subject to:

$$\begin{aligned} x_1 - x_2 + 2x_3 - 6 &= 0 \\ x_1 - 2x_2 - 8 &= 0 \end{aligned} \quad (13)$$

In this case, M=2, that is, system (13) is determined by two dependent variables, and the third is an independent parameter. We define x_1 as an independent parameter. In this case, x_2 and x_3 are dependent.

This test problem also has an analytical solution. It can be seen that the objective function, being non-negatively defined, reaches the minimum, zero value at the point x_3 = 0.15. In this case, to fulfill the restrictions (13), the variables x_1 and x_2 take the following values: x_1=3.4, x_2= - 2.3. Let us find a solution to the problem by the developed approach. Using the generalized optimization approach, system (13) is transformed into the following system:

$$\begin{aligned} (1-u_1)(x_1 - x_2 + 2x_3 - 6) &= 0 \\ (1-u_2)(x_1 - 2x_2 - 8) &= 0 \end{aligned} \quad (14)$$

The control vector for this example has two components: U=(u_1,u_2). Table 2 shows the dynamics of changes in the number of generations and processor time (s) of the GA depending on the required precision δ of minimizing the objective function *F* for three strategies: TOS, MTOS, and composite strategy (00)(11) with optimal switching point Sp between strategies (00) and (11).

Table 2. Dynamics of changes in the number of generations and processor time (s) of the GA depending on the required precision δ of function *F* for three strategies: TOS, MTOS, and composite strategy (00)(11) with optimal switching point Sp

δ	G (CPU time (s))		
	U=(00)	U=(11)	U=(00)(11)
$4 \cdot 10^{-2}$	29 (0.043)	22 (0.06)	14 (0.038) Sp=2
$2 \cdot 10^{-2}$	1017 (1.472)	25 (0.067)	16 (0.043) Sp=2
10^{-2}	4118 (5.959)	25 (0.067)	18 (0.047) Sp=2
$5 \cdot 10^{-3}$	165741 (240.53)	27 (0.07)	19 (0.05) Sp=2
10^{-3}	-	32 (0.085)	21 (0.063) Sp=2
10^{-4}	-	39 (0.107)	37 (0.105) Sp=2
$5 \cdot 10^{-5}$	-	69 (0.186)	45 (0.124) Sp=5
10^{-5}	-	-	51 (0.142) Sp=16
10^{-6}	-	-	75 (0.207) Sp=27
10^{-7}	-	-	82 (0.226) Sp=27
$3 \cdot 10^{-8}$	-	-	149 (0.416) Sp=27
$2 \cdot 10^{-8}$	-	-	-

The traditional strategy requires many more generations than modified or composite strategies while obtaining the same precision.

Analyzing the results in the table, one can see that TOS can find a solution with a precision of 10^{-3} and no higher. At the same time, the MTOS with the control vector (11) and the composite strategy with the control vector (00)(11) makes it possible to find a solution with a precision of $5 \cdot 10^{-5}$ and $3 \cdot 10^{-8}$, respectively.

It can be seen that MTOS with U = (11) and a combined strategy with U = (00)(11) find a solution for a smaller number of generations and a smaller processor time than TOS with U=(00).

We see that MTOS and the composite strategy solve an optimization problem with two orders of magnitude fewer generations than TOS for 10^{-2} precision and four orders of magnitude less for $5\cdot10^{-3}$ precision.

TOS solves the optimization problem in 5.959 s with a precision of 10^{-2} and 240.53 s with a precision of $5\cdot10^{-3}$. The composite strategy solves this problem in 0.047 s with an accuracy of 10^{-2} and 0.05 s with an accuracy of $5\cdot10^{-3}$. In this case, the CPU time gain is 126 times for 10^{-2} precision and 4810 times for $5\cdot10^{-3}$ precision.

The dependence of the generalized objective function F is shown in Figure 2 under successive change of generations for strategies (00), (11), and composite strategy (00)(11).

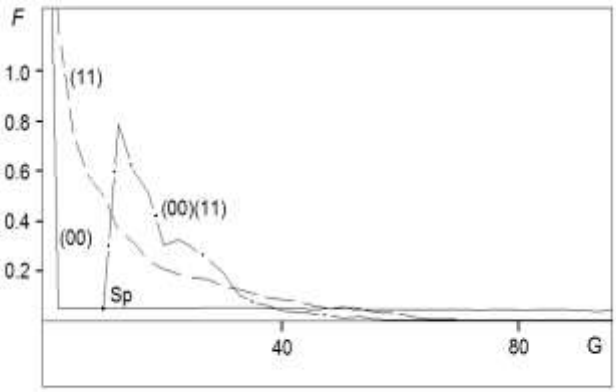


Fig. 2: Dependence of the function F under successive generational change for strategies (00), (11), and composite strategy (00)(11)

It can be seen that for the three presented strategies, different behavior of the function F is observed. MTOS and the composite strategy provide a large gain in generation number and CPU time to ensure the desired precision.

Variables x_1, x_2, and x_3 take values of 3.4, -2.3, and 0.15, respectively, but with different degrees of accuracy for the three studied strategies.

3.3 Example 3

This example analyzes the process of optimizing a function for one of the reference problems - finding the global minimum of the modified Shekel function. This function is given by the next formula:

$$C(X) = -\sum_{i=1}^{m}\left(\sum_{j=1}^{N}\left(x_j - a_{ij}\right)^2 + c_i\right)^{-1} + c_0 \quad (15)$$

where m is the number of possible minima of the function, N is the total number of variables, a_{ij} are the coordinates of possible minima, and c_i are the coefficients that determine the values of possible minima. There is no coefficient c_0 in the standard definition of the Shekel function. Such assignment of the Shekel function is typical for the problem of unconstrained optimization. Possible minima of function (15) are located in the negative area and the global minimum corresponds to the deepest dip. Let us define the following coefficients in formula (15): N = 2, m = 5. For this example, the Shekel function depends on two variables x_1 and x_2, and is defined by five possible minima given by the following coordinates: a_{11} = 1.10, a_{12} = 0.0316, a_{21} = 2.0, a_{22} = 1.0, a_{31} = 3.0, a_{32} = 2.828427, a_{41} = 3.5, a_{42} = 3.952847, a_{51} = 4.0, a_{52} = 5.196. Each pair of coefficients determines the coordinates of the minima. The values of the minima correspond to the coefficients c_1, c_2, c_3, c_4, and c_5, which are defined below. Since the optimization problem is being solved in the presence of constraints, we set constraints in the following form:

$$\left(x_1 - 1\right)^3 - x_2^2 = 0\,, \quad (16)$$

$$x_1 \geq 0,\; x_2 \geq 0. \quad (17)$$

Equation (16) is a relationship equation between variables, being a model of some system and when an independent variable x_1 is specified, the dependent variable x_2 is uniquely determined.

A feature of optimizing an electronic circuit and applying a generalized approach is that the objective function can be set to be non-negative and its global minimum, therefore, has a value of 0. In this case, some modification of the Shekel function is required, which consists of adding the coefficient c_0 in formula (15), which is equal to the absolute value of the global minimum. In this case, the entire function "rises" by the value of the global minimum and is non-negative.

The presence of one independent variable and one dependent corresponds to K=1, M=1. Using a generalized approach to optimization, equation (16) is transformed into the following equation:

$$\left(1-u\right)\left(\left(x_1 - 1\right)^3 - x_2^2\right) = 0 \quad (18)$$

In this case, only two main strategies TOS and MTOS, and possible compound strategies can be defined.

Numerical analysis of the Shekel function (15) for given coefficients and c_0 = 0 made it possible to reveal the presence of four minima, one of which is global, at the points corresponding to the first four pairs of coefficients a_{ij}.

Let us consider three variants of the distribution of the minima of the Shekel function.

3.3.1 Option 1

The minima correspond to the following coefficients: $c_1 = 0.1$, $c_2 = 0.2$, $c_3 = 0.3$, $c_4 = 0.2$, $c_5 = 0.3$. The values of the minima are as follows: $C_{min1} = -10.6454$, $C_{min2} = -5.8458$, $C_{min3} = -4.2235$, and $C_{min4} = -5.6889$. The first minimum is global and corresponds to the coordinates: $x_1 = 1.1$, $x_2 = 0.0316$. The coefficient c_0 in formula (15) is taken as equal to 10.6454.

Function optimization results (15) under constraints (16)-(17) for TOS, MTOS, and composite strategies that include two main strategies with a control vector (0)(1) are given in Table 3, Figure 3 and Figure 4.

Table 3. Dynamics of changes in the number of generations and processor time (s) of the GA depending on the required precision δ of function *F* for two strategies: MTOS and composite strategy (0)(1) with the optimal switching point Sp=3

δ	G (CPU time (s))	
	U=(1)	U=(0)(1)
10^{-1}	22 (0.05)	24 (0.058)
10^{-2}	38 (0.086)	38 (0.087)
10^{-3}	45 (0.10)	42 (0.098)
10^{-4}	65 (0.144)	55 (0.128)
10^{-5}	79 (0.174)	56 (0.131)
10^{-6}	80 (0.18)	58 (0.136)
10^{-7}	91 (0.202)	79 (0.175)
10^{-8}	909 (2.014)	812 (1.802)
$3. 10^{-9}$	-	37659 (83.573)
$2. 10^{-9}$	-	-

The traditional TOS strategy comes to a local minimum with *F*=4.75 and coordinates $x_1 = 2.0$, $x_2 = 1.0$. That is, we can state that this strategy does not find a solution to the problem. At the same time, the MTOS and composite strategy find a global minimum equal to zero with coordinates $x_1 = 1.10$, $x_2 = 0.0316$. The table shows the results of the optimization process for different accuracy δ of minimizing the objective function *F* for MTOS and a composite strategy with control vector (0)(1) and switching point Sp=3.

A comparison of these strategies shows a slight advantage of the composite strategy while increasing the required accuracy of solving the problem.

Figure 3 shows the trajectories of the optimization process, including two components x_1 and x_2 of the vector *X*, calculated by the formula (12) for three strategies, TOS, MTOS, and a composite strategy with a control vector (0)(1).

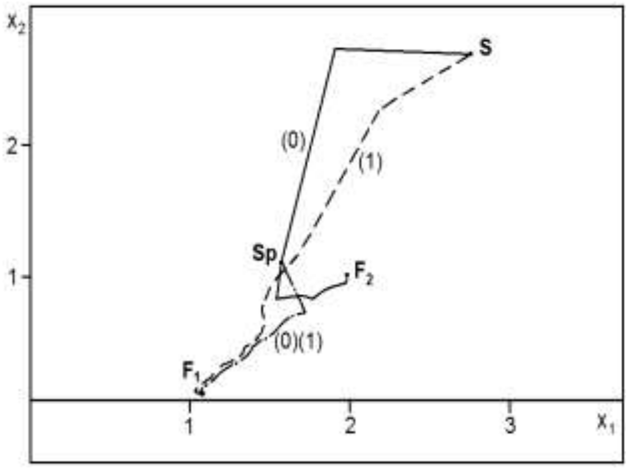


Fig. 3: Trajectories of the optimization process for three strategies (0), (1) and composite strategy (0)(1)

Point S corresponds to the starting point of the optimization process, F_1 is the final point of the optimization process, corresponding to MTOS and the composite strategy (0)(1) and is the global minimum point, F_2 is the final point of the optimization process, corresponding to TOS and being one of the local minima.

Sp is the switching point from strategy (0) to strategy (1). It is important to emphasize that the TOS corresponding to the control vector (0) has a "hard" trajectory in the sense that condition (16) must always be satisfied on this trajectory. At the same time, the other two strategies work under the conditions of two independent variables x_1 and x_2, and condition (16) may not be satisfied on the entire trajectory, except for the final point. In this sense, these two strategies are more stochastic, which ultimately leads to the possibility of "skipping past" local minima and finding a global one.

The dependence of the generalized objective function *F* on the number of generations is shown in Figure 4 for three strategies TOS, MTOS, and a composite strategy with a control vector (0)(1) for an accuracy of $\delta=10^{-5}$. Sp is the switching point from one strategy to another.

The function *F* for TOS decreases to 4.75 and then does not change, which corresponds to a local minimum.

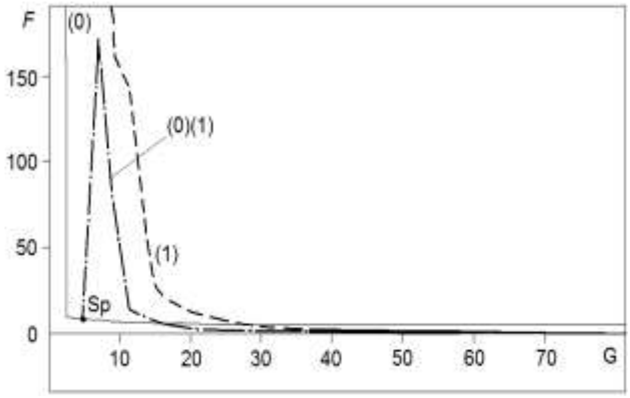


Fig. 4: Dependence of the function F under successive generational change for strategies (0), (1), and composite strategy (0)(1)

At the same time, for the other two strategies, the function F decreases to the values 10^{-8}–10^{-9} giving a high accuracy of the optimization process implementation, since it corresponds to the global minimum.

3.3.2 Option 2

The minima correspond to the following coefficients: $c_1 = 0.15$, $c_2 = 0.1$, $c_3 = 0.3$, $c_4 = 0.2$, $c_5 = 0.3$. The values of the minima are as follows: $C_{min1} = -7.3399$, $C_{min2} = -10.8316$, $C_{min3} = -4.2280$, and $C_{min4} = -5.6896$. The second minimum is global and corresponds to the coordinates: x_1=2.0, x_2=1.0. The coefficient c_0 in formula (15) was set equal to 10.8316.

Optimization results of function (15) under constraints (16)-(17) for strategies TOS, MTOS, and a composite one that includes two main strategies with a control vector (0)(1) are given in Table 4.

Table 4. Dynamics of changes in the number of generations and processor time (s) of the GA depending on the required precision δ of function F for three strategies: TOS, MTOS, and composite strategy (0)(1) with optimal switching point Sp=1

δ	G (CPU time (s))		
	U=(0)	U=(1)	U=(0)(1)
10^{-2}	26 (0.041)	32 (0.073)	43 (0.098)
10^{-3}	31 (0.048)	60 (0.137)	76 (0.174)
10^{-4}	37 (0.058)	97 (0.222)	79 (0.181)
10^{-5}	350 (0.549)	99 (0.227)	85 (0.194)
10^{-6}	1212 (1.903)	949 (2.173)	201 (0.460)
$3.\ 10^{-7}$	-	9179 (21.020)	666 (1.525)
10^{-7}	-	-	8998 (20.605)
$2.\ 10^{-8}$	-	-	13457 (30.816)
10^{-8}	-	-	-

All three strategies find the global minimum corresponding to the point with coordinates $x_1 = 2.0$, and $x_2 = 1.0$, however, the accuracy of finding this minimum is different for these strategies. TOS finds the minimum with a marginal accuracy of 10^{-6}, MTOS with an accuracy of $3\cdot10^{-7}$, and a compound strategy with an accuracy of $2\cdot10^{-8}$.

3.3.3 Option 3

The minima correspond to the following coefficients: $c_1 = 0.2$, $c_2 = 0.1$, $c_3 = 0.07$, $c_4 = 0.15$, $c_5 = 0.3$. The values of the minima are as follows: $C_{min1} = -5.6751$, $C_{min2} = -10.8296$, $C_{min3} = -15.1971$ and $C_{min4} = -7.4365$. The third minimum is global and corresponds to the coordinates: x_1=3.0, x_2=2.828. The coefficient c_0 in formula (15) was set equal to 15.1971.

The results of optimization of function (15) under constraints (16)-(17) for two strategies: MTOS and composite, including two main strategies of the structural basis with control vector (0)(1), are given in Table. 5.

Table 5. Dynamics of changes in the number of generations and processor time (s) of the GA depending on the required precision δ for two strategies: MTOS and composite strategy (0)(1) with the optimal switching point Sp=1

δ	G (CPU time (s))	
	U=(1)	U=(0)(1)
10^{-1}	33 (0.075)	32 (0.073)
10^{-2}	52 (0.119)	48 (0.110)
10^{-3}	61 (0.140)	61 (0.140)
10^{-4}	66 (0.151)	79 (0.181)
10^{-5}	73 (0.167)	79 (0.181)
$5.\ 10^{-6}$	6655 (15.240)	81 (0.185)
$4.\ 10^{-6}$	94449 (216.288)	82 (0.186)
10^{-6}	-	366 (0.838)
$2.\ 10^{-7}$	-	29672 (67.949)
10^{-7}	-	-

In this case, as well as in the first variant, the traditional strategy does not find a global minimum but stops in a local minimum with coordinates x_1=2.0, x_2=1.0. MTOS and the composite strategy find the global minimum corresponding to the point with coordinates x_1= 3.0, x_2 = 2.828. At the same time, the composite strategy finds a minimum with a maximum accuracy of $2\cdot10^{-7}$, which is an order of magnitude better than the MTOS strategy.

The analysis of this example allows us to understand the specifics of optimizing a multi-extremal function in the presence of restrictions. In this case, the use of the traditional strategy does not always allow one to find the global minimum, since the process can loop in local minima. At the same time, some strategies emerging from the generalized approach can overcome this problem and find the global minimum with a high degree of accuracy.

3.4 Example 4

Let us optimize the circuit of a four-node nonlinear voltage divider shown in Figure 5. The conductivities y_1, y_2, y_3, y_4, y_5 are positive and represent a set of parameters for a given circuit (*K*=5) that are defined as independent. Voltages in circuit nodes V_1, V_2, V_3, V_4 are dependent parameters (*M*=4). Circuit optimization aims to obtain the required values of all nodal voltages $V_{10}, V_{20}, V_{30}, V_{40}$ by selecting conductivities.

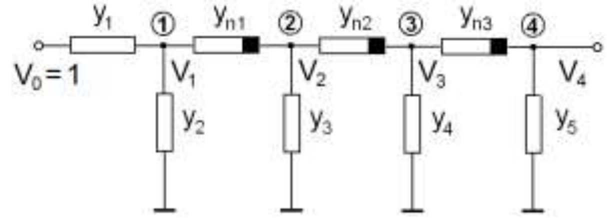


Fig. 5: Four-node nonlinear passive circuit

Given that the voltage at the input of the divider is 1 V, these constants in the normalized form have the following values: V_{10}=0.7, V_{20}=0.4, V_{30}=0.2, V_{40}=0.1.

In mathematical terms, this problem can be represented as a problem of minimizing some objective function. Let us define the objective function of the optimization process using the following formula:

$$C(X) = \sum_{i=1}^{M} \left[\left(V_i - V_{i0} \right)^2 \right] \qquad (19)$$

The mathematical model of the circuit in this case acts as a set of restrictions.

Let's define non-linear elements by the following expressions: $y_{n1} = a_{n1} + b_{n1} \cdot (V_1 - V_2)^2$, $y_{n2} = a_{n2} + b_{n2} \cdot (V_2 - V_3)^2$ and $y_{n3} = a_{n3} + b_{n3} \cdot (V_3 - V_4)^2$, where $a_{n1} = a_{n2} = a_{n3} = 1$, and $b_{n1} = b_{n2} = b_{n2} = 0.9$. Vector *X* includes nine components $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$, where: $x_1^2 = y_1$, $x_2^2 = y_2$, $x_3^2 = y_3$, $x_4^2 = y_4$, $x_5^2 = y_5$, $x_6 = V_1$, $x_7 = V_2$, $x_8 = V_3$ and $x_9 = V_4$. These formulas for the components x_1, x_2, x_3, x_4, x_5 always make it possible to obtain positive conductivities. This removes the problem of the mandatory positive definiteness of each component of the vector *X*. The first five components of this vector can have both positive and negative values. In this case, the conductivities are always positive.

Formula (19) is transformed into the following form:

$$C(X) = \sum_{i=1}^{M} \left[\left(x_{K+i} - V_{i0} \right)^2 \right] \qquad (20)$$

Taking into account the Kirchhoff laws, the mathematical model of the circuit can be represented by four equations of the nodal voltage method, and the functions $g_j(X)$ are given using the following formulas:

$$g_1(X) \equiv -x_1^2 + \left(x_1^2 + x_2^2 \right) x_6 + \left\{ a_{n1} + b_{n1} \left(x_6 - x_7 \right)^2 \right\} \left(x_6 - x_7 \right) = 0$$

$$\begin{aligned} g_2(X) \equiv x_3^2 x_7 &+ \left\{ a_{n1} + b_{n1} \left(x_6 - x_7 \right)^2 \right\} \left(x_7 - x_6 \right) \\ &+ \left\{ a_{n2} + b_{n2} \left(x_7 - x_8 \right)^2 \right\} \left(x_7 - x_8 \right) = 0 \end{aligned} \qquad (21)$$

$$\begin{aligned} g_3(X) \equiv x_4^2 x_8 &+ \left\{ a_{n2} + b_{n2} \left(x_7 - x_8 \right)^2 \right\} \left(x_8 - x_7 \right) \\ &+ \left\{ a_{n3} + b_{n3} \left(x_8 - x_9 \right)^2 \right\} \left(x_8 - x_9 \right) = 0 \end{aligned}$$

$$g_4(X) \equiv x_5^2 x_9 + \left\{ a_{n3} + b_{n3} \left(x_8 - x_9 \right)^2 \right\} \left(x_9 - x_8 \right) = 0$$

Therefore, we must minimize the function *C(X)* given by expression (20) with additional conditions (21). The control vector *U* has four components: $U = (u_1, u_2, u_3, u_4)$.

Applying formulas (6) and (7), gives the following formula for the generalized objective function *F*:

$$F(X,U) = C(X) + \frac{1}{\sigma} \sum_{j=1}^{4} u_j g_j^2(X). \qquad (22)$$

The number of structural basis strategies is quite large and equals 16. Of course, there are a large number of possible combinations of different strategies, but, as was shown in [21], when using deterministic optimization methods, the best results should be expected from a combination of TOS and MTOS strategies with the control vector (00...0) and (11...1).

Table 6 shows the dynamics of changes in the number of generations and processor time (s) of the GA depending on the required precision δ for three strategies: TOS, MTOS, and composite strategy (0000)(1111) with the optimal switching point Sp=6 between strategies (0000) and (1111).

Table 6. Dynamics of changes in the number of generations and processor time (s) of the GA depending on the required precision δ for three strategies: TOS, MTOS, and composite strategy (0000)(1111) with the optimal switching point Sp=6

	G (CPU time (s))		
δ	U=(0000)	U=(1111)	U=(0000)(1111)
4.10^{-3}	77 (0.298)	72 (0.074)	68 (0.087)
5.10^{-4}	80 (0.31)	77 (0.079)	69 (0.088)
$3.765\ 10^{-4}$	176809 (684.25)	81 (0.083)	70 (0.089)
$3.76\ 10^{-4}$	-	81 (0.083)	70 (0.089)
3.10^{-4}	-	84 (0.086)	72 (0.091)
10^{-4}	-	93 (0.096)	75 (0.094)
10^{-5}	-	111 (0.114)	82 (0.101)
10^{-6}	-	126 (0.130)	84 (0.104)
2.10^{-7}	-	148 (0.152)	86 (0.106)
10^{-7}	-	-	88 (0.108)
4.10^{-8}	-	-	164 (0.186)
$3.9\ 10^{-8}$	-	-	-

It can be stated that the use of MTOS and the composite strategy makes it possible to obtain a significant gain compared to TOS both in terms of the number of generations and processor time to achieve an accuracy of $3.765\cdot10^{-4}$. It should be noted that this is the ultimate accuracy that a traditional optimization strategy can achieve.

MTOS with a control vector (1111) and a composite strategy with a control vector (0000)(1111) has an advantage over TOS of more than 2000 times in the number of generations and more than 8000 times in processor time. TOS does not find a solution if the required error is reduced to a value less than $3.765\ 10^{-4}$. In contrast, MTOS and the composite strategy find solutions up to a precision of $2\cdot10^{-7}$ or $4\cdot10^{-8}$ for the first and second strategies, respectively. The number of GA generations as a function of the position of the switching point Sp for the composite strategy (0000)(1111) for accuracy $\delta = 10^{-5}$ is presented in Table 7.

The optimal value of the switching point between strategies Sp = 6. That is, the strategy with the control vector (0000) works for the first five steps and the subsequent ones with the vector (1111).

Table 7. Number of generations as a function of the switching point Sp of the composite strategy (0000)(1111)

Switch point Sp	4	5	6	7	8	9	10	11
Number of generation G	98	85	82	84	89	112	109	119

The dependences of the generalized objective function *F* on the successive change of generations for the strategies with the control vector (0000), (1111) and the composite strategy (0000)(1111) with a given error $\delta = 2\cdot10^{-7}$ are shown in Figure 6.

Figure 6 shows the dependence of the generalized objective function *F* under successive generational change for strategies with the control vector (0000), (1111), and composite strategy (0000)(1111) for a given error $\delta = 2\cdot10^{-7}$.

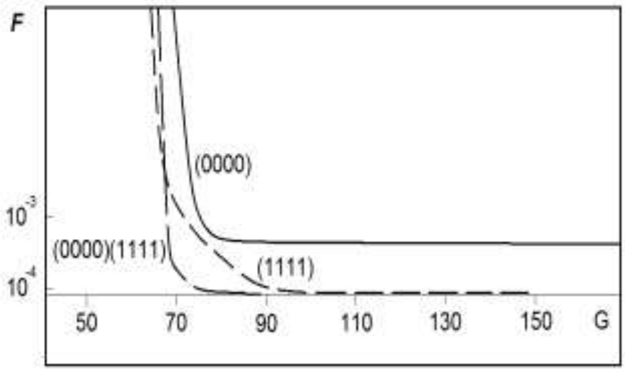


Fig. 6: Dependence of the generalized objective function *F* under successive generational change for strategies (0000), (1111), and composite strategy (0000)(1111)

It can be seen from the figure that TOS does not provide good accuracy of the solution, unlike MTOS and the composite strategy. Conversely, the MTOS and the composite strategy give a solution to the problem with high accuracy ($2\cdot10^{-7}$) in a relatively small number of generations. It is important to emphasize that TOS cannot solve the problem with such accuracy in a foreseeable period.

A new population with different properties is formed for a composite strategy at the switching point Sp. At this point, the population structure changes drastically and the optimization process leaves the local minimum trap. For this reason, this strategy achieves the minimum of the objective function with greater precision than other strategies.

3.5 Example 5

The next example demonstrating the procedure for optimization considered is the two-cascade transistor amplifier shown in Figure 7.

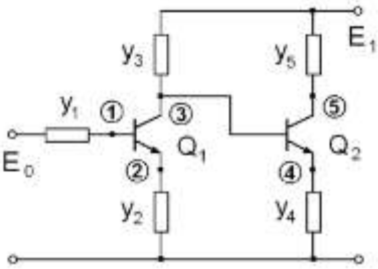


Fig. 7: Two-cascade transistor amplifier

We have five independent variables for this circuit: y_1, y_2, y_3, y_4 and y_5 (K=5) and five dependent variables: V_1, V_2, V_3, V_4 and V_5 (M=5). All the components of the vector X are defined with the following formulas: $x_1^2 = y_1$, $x_2^2 = y_2$, $x_3^2 = y_3$, $x_4^2 = y_4$, $x_5^2 = y_5$, $x_6 = V_1$, $x_7 = V_2$, $x_8 = V_3$ and $x_9 = V_4$ and $x_{10} = V_5$. The static Ebers – Moll model, [23], is chosen for the approximation of transistor characteristics. The objective function has the same form as in Example 4:

$$C(X) = \sum_{i=1}^{M} \left[(V_i - V_{i0})^2 \right] \qquad (23)$$

We set the required node voltages as (in volts): V_{10}=1.75, V_{20}=1.0, V_{30}=3.2, V_{40}=2.5, V_{50} =5.6. The control vector U is formed with five control functions: U = (u_1, u_2, u_3, u_4, u_5). There are 32 optimization strategies on the structural basis. The mathematical model of the circuit (24) consists of five equations:

$$g_1(X) \equiv I_{B1} + (x_6 - E_0)x_1^2 = 0$$

$$g_2(X) \equiv I_{E1} + x_7 x_2^2 = 0$$

$$g_3(X) \equiv I_{E2} + x_9 x_4^2 = 0 \qquad (24)$$

$$g_4(X) \equiv I_{C2} + (x_{10} - E_1)x_5^2 = 0$$

$$g_5(X) \equiv I_{C1} + I_{B2} + (x_8 - E_1)x_3^2 = 0$$

where I_{B1}, I_{B2}, I_{E1}, I_{E2}, I_{C1}, I_{C2} – are the base, emitter, and collector currents of the first and the second transistor. According to the generalized approach considered the system is converted to the following one:

$$(1 - u_j)g_j(X) = 0, \; j = 1,2,3,4,5. \qquad (25)$$

The function $F(X)$ has the form:

$$F(X,U) = C(X) + \frac{1}{\sigma}\sum_{j=1}^{5} u_j g_j^2(X) \qquad (26)$$

One can try algorithm schemes with different amounts of switching points between strategies to achieve better results of the algorithm efficiency. For this scheme, we choose a variant with two switching points. Table 8 shows the generation numbers and processor time when the function F(X) achieves the required accuracy δ for various strategies: TOS with control vector (00000), MTOS with control vector (11111), and composite strategy with control vector (11111)(00000)(11111) and two switching points Sp1=5 and Sp2 = 9 giving the best result for processor time.

Table 8. Dependencies of the number of generations and processor time (s) on the required precision δ for TOS, MTOS, and the composite strategy (11111)(00000)(11111) with switching points Sp1=5 and Sp2 = 9

	G (CPU time (s))		
δ	U=(00000)	U=(11111)	U=(1...1)(0...0)(1...1)
5. 10^{-2}	28563 (931.89)	52 (0.32)	38 (0.235)
10^{-2}	389533 (12708)	56 (0.344)	43 (0.251)
5. 10^{-3}	1691364 (55181)	59 (0.364)	47 (0.268)
10^{-3}	-	65 (0.408)	52 (0.278)
10^{-4}	-	80 (0.492)	62 (0.309)
10^{-5}	-	88 (0.542)	66 (0.321)
10^{-6}	-	94 (0.578)	78 (0.358)
1.7 10^{-7}	-	134 (0.824)	87 (0.385)
1.03 10^{-7}	-	-	114 (0.469)
1.02 10^{-7}	-	-	-

One can see that amounts of generations in which MTOS and the composite strategy need to achieve some definite accuracy is much less than those corresponding amounts for TOS. In addition, the best accuracy achieved by TOS over 15 hours of CPU time is not within the range of the desired accuracy levels of the optimization process. The time that TOC requires to achieve an accuracy above $5 \cdot 10^{-3}$ is clearly beyond reasonable values. Accuracy which is achieved with MTOS and the composite strategy has the magnitude orders -11… -12. Results shown in Table 8 demonstrate that the composite strategy allows achieving the stationary mode with lesser processor time than MTOS.

The result of the work of the algorithm is influenced by the position of switching points. Table 9 shows how this influence manifests itself in dependencies of the number of generations and processor time on the second switching position Sp2 when the first point Sp1=5.

Table 9. Number of generations and processor time as a function of the second switching point Sp2 for the composite strategy (11111)(00000)(11111), Sp1 = 5. The accuracy achieved is 10^{-6}

Switch point Sp2	6	7	8	9	10	11	12	13
Number of generation G	85	82	92	78	81	87	99	118

It is clear that the switching point ultimately determines the number of generations needed for a given precision. The results show that the minimum number of generations is 78 and corresponds to the optimal second switching point Sp2 = 9.

Dependencies of the objective function on the successive generation change for strategies (00000), (11111), and the composite strategy (11111)(00000)(11111) with Sp1 = 4, Sp2 = 8 are shown in the Figure 8. Sp2 = 8 is chosen as the best switching point value for the number of generations for achieving accuracy 10^{-5}.

Despite the truly huge number of generations of more than 10^6, TOC does not allow obtaining an accuracy of 10^{-3}. MTOS and the composite strategy achieve an accuracy of 10^{-5} within the first hundred generations. For the accuracy $5 \cdot 10^{-3}$ MTOS has a time gain of 151596 times compared to TOS.

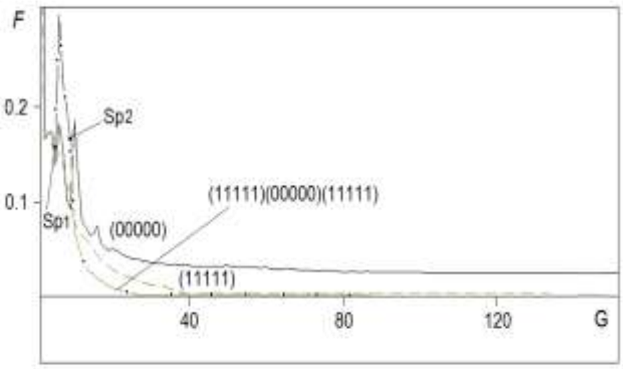


Fig. 8: Dependence of the generalized objective function *F* under successive generational change for strategies (00000), (11111) and composite strategy (11111)(00000)(11111)

The combined strategy has a gain of 205899 times compared to TOS.

Using the generalized optimization approach within the genetic algorithm is a mechanism that contributes to changing the internal structure of the vector *X*, and at the same time, changing the structure of the principal function of the GA - the fitness function. This effect manifests itself within the optimization process since it depends on the structure of the control vector *U*, which can be changed at any step of the optimization process. In this case, the GA has the opportunity to get around local minima and continue the search for a global minimum.

New strategies that appear within the idea of generalized optimization help to increase the accuracy of the solution and reduce the processor time. This can be seen from a comparison of the results obtained using TOS, MTOS, and a combined strategy.

The results obtained in this section show that changing the mechanism for calculating the fitness function during the operation of the GA leads to the exit from local minima and overcoming premature convergence. In this case, the accuracy of the solution can be significantly increased, which can be transformed into both a reduction in the number of possible generations and a reduction in processor time.

4 Conclusion

Previously, based on control theory, a generalized approach to the problem of optimizing electronic circuits was developed using such deterministic methods as the gradient method, Newton's method, etc. This made it possible to determine many different optimization strategies by introducing a control vector and to formulate the problem of finding the optimal strategy by optimizing the structure of this vector. It was shown that this approach provides a significant acceleration of the optimization procedure through the use of various strategies and the formation of composite strategies.

The application of a similar approach in the case of using a genetic algorithm as the basis of an optimization procedure leads to a change in the structure and main parameters of this algorithm. The results of the study demonstrate the possibility of introducing the idea of generalized optimization into the body of the genetic algorithm, which leads to a change in the structure of chromosomes and the fitness function during the operation of the algorithm and the formation of a set of different optimization strategies. In turn, the emergence of a set of strategies inside the GA makes it possible to use various strategies of this set, as well as to form their combinations, which can significantly improve the characteristics of the optimization process. The results obtained show that changing the main parameters of the GA makes it possible to bypass local minima and overcome premature convergence. An analysis of the optimization procedure for some electronic circuits showed the effectiveness of this approach. In this case, it becomes possible to increase the optimization accuracy by 3–4 orders of magnitude and reduce processor time by 3-5 orders of magnitude compared to traditional GA. Thus, it can be emphasized that new optimization strategies that appear within the framework of the presented methodology have good prospects both for improving the process of solving a nonlinear programming problem in general, and especially for optimizing electronic systems. It can be assumed that such a methodology for solving the optimization problem, based on a generalized approach, can be extended to other stochastic optimization methods, which may be the subject of future research. In this case, an improvement in the performance of the optimization process is also expected.

References:

[1] R. S. Zebulum, M.A. Pacheco and M.M. Be Vellasco, *Evolutionary electronics, Automatic Design of Electronic Circuits and Systems by Genetic Algorithms*, Taylor & Francis Group, eBook Published 2017, p.320, https://doi.org/10.1201/9781420041590.

[2] M. W. Cohen, M. Aga, and T. Weinberg, Genetic Algorithm Software System for Analog Circuit Design, *Procedia CIRP*, Vol.36, 2015, pp. 17-22. DOI: 10.1016/j.procir.2015.01.033

[3] P. Das, B. Jajodia, Design Automation of Two-Stage Operational Amplifier Using Multi-Objective Genetic Algorithm and SPICE Framework, *International Conference on Inventive Computation Technologies (ICICT)*, Lalitpur, Nepal, 2022, pp. 166-170.

[4] Y. C. Wong , D.W.F. Yap, N. Mazlina, A.R. Syafeeza and L.S. Pang, Optimization of Electrical Circuits Using Genetic Algorithms (GAs), *Solid State Science and Technology*, Vol. 1, No. 1, 2007, pp.1-11.

[5] T. Ueda, J. Katagiri, T. Oki and S. Koyanaka, Genetic algorithm optimization in discrete element simulation of electric parts separation from printed circuit board, *Structural and Multidisciplinary Optimization*, Vol.64, 2021, pp. 2763-2771, https://doi.org/10.1007/s00158-021-02982-4.

[6] A. P. Vaze, Analog Circuit Design using Genetic Algorithm: Modified, *International Journal of Electronics and Communication Engineering*, Vol.2, No.2, 2008, pp. 301-303.

[7] P. Prem Kumar, An Optimized Device Sizing of Analog Circuits using Genetic Algorithm, *European Journal of Scientific Research*, Vol.69, No.3, 2012, pp.441-448.

[8] A. Lberni, M.A. Marktani, A. Ahaitouf and Al. Ahaitouf, Analog circuit sizing based on Evolutionary Algorithms and deep learning, *Expert Systems with Applications*, V.237, Part B, No.3, 2024, pp. 121480-121482, DOI:10.1016/j.eswa.2023.121480.

[9] M. Barari, H.R. Karimi and F. Razaghian, Analog Circuit Design Optimization Based on Evolutionary Algorithms, *Math. Problems in Engineering*, Vol.2014, 2014, pp. 1-12, https://doi.org/10.1155/2014/593684.

[10] R. A. de Lima Moreto, C.E. Thomaz, S.P. Gimenez, A customized genetic algorithm with in-loop robustness analyses to boost the optimization process of analog cmos ics, *Microelectronics Journal*, Vol.92, 2019, pp.12. DOI:10.1016/j.mejo.2019.07.013.

[11] R. Zhou, P. Poechmueller and Y. Wang, An Analog Circuit Design and Optimization System With Rule-Guided Genetic Algorithm, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.41, No.12, 2022, pp. 5182-5192, DOI: 10.1109/TCAD.2022.3166637.

[12] S. Malik and S. Wadhwa, Preventing Premature Convergence in Genetic Algorithm Using DGCA and Elitist Technique, *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol.4, No.6, 2014, pp. 410-418.

[13] A. S. Aldallal, Avoiding Premature Convergence of Genetic Algorithm in Informational Retrieval Systems, *International Journal of Intelligent Systems and Applications in Engineering,* Vol.2, No.4, 2014, pp. 80-85, DOI:10.18201/ijisae.78975.

[14] F. W. Yang, H.J. Lin and S.H. Yen, An Improved Unsupervised Clustering Algorithm Based on Population Markov Chain, *International Journal of Computers and Applications*, Vol.29, No.3, 2015, pp. 253–258, https://doi.org/10.1080/1206212X.2007.11441855.

[15] A. E. Eiben, E. Aarts, K.M. van Hee, Global convergence of genetic algorithms: A Markov chain analysis, *International Conference on Parallel Problem Solving from Nature*, In book: *Parallel Problem Solving from Nature*, pp. 3-12, 2006. DOI:10.1007/BFb0029725

[16] E. S. Nicoara, Mechanisms to Avoid the Premature Convergence of Genetic Algorithms, *BULETINUL Universitatea Petrol – Gaze din Ploiesti, Seria Matematica -*

Informatica – Fizica, Vol. 61, No.1, 2009, pp. 87–96.

[17] S. Ramadan, Reducing Premature Convergence Problem in Genetic Algorithm: Application on Travel Salesman Problem, *Computer and Information Science*, Vol.6, No.1, 2013, pp. 47-57. DOI:10.5539/cis.v6n1p47.

[18] H. Fu, Y. Xu, G. Wu, H. Jia, W. Zhang and R. Hu, An Improved Adaptive Genetic Algorithm for Solving 3-SAT Problems Based on Effective Restart and Greedy Strategy, *International Journal of Computational Intelligence Systems*, Vol.11, 2018, pp. 402–413. DOI:10.2991/ijcis.11.1.30.

[19] N. G. A. P. H. Saptarini, P.I. Ciptayani, N.W. Wisswani, I.W. Suasnawa, N.E. Indrayana, Comparing Selection Method in Course Scheduling Using Genetic Algorithm, *International Conference on Science and Technology (ICST 2018),* Yogyakarta, Indonesia. *Atlantis Highlights in Engineering (AHE),* Vo.1, pp. 574-578, 2018.

[20] G. Takasao, T. Wada, H. Chikuma, P. Chammingkwan, M. Terano, T. Taniike, Preventing Premature Convergence in Evolutionary Structure Determination of Complex Molecular Systems: Demonstration in Few-Nanometer-Sized $TiCl_4$-Capped $MgCl_2$ Nanoplates, The *Journal of Physical Chemistry A.*, Vol.126, No.31, 2022, pp. 5215-5221. doi: 10.1021/acs.jpca.2c02112.

[21] A. Zemliak, Analog circuit optimization on basis of control theory approach, *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol.33, No.6, 2014, pp. 2180–2204, https://doi.org/10.1108/COMPEL-10-2013-0324.

[22] A. Zemliak, A modified genetic algorithm for system optimization, *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol.41, No.1, 2022, pp. 499-516, https://doi.org/10.1108/COMPEL-08-2021-0296.

[23] G. Massobrio and P. Antognetti, *Semiconductor Device Modeling with SPICE*, Mc. Graw-Hill, Inc.: New York, 1993.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- Alexander Zemliak formulated the idea of research and the structure of the article and also carried out the analysis and identification of results.
- Andrei Osadchuk participated in checking the calculation results and searching for optimal solutions.
- Christian Serrano was involved in the software implementation of the algorithms and participated in the analysis and discussion of the results.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.