

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,300

Open access books available

129,000

International authors and editors

155M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Hybrid Genetic Algorithm for Fast Electromagnetic Synthesis

Artem V. Boriskin<sup>1,2</sup> and Ronan Sauleau<sup>2</sup>

<sup>1</sup>*Institute of Radiophysics and Electronics of the National Academy of Sciences of Ukraine, Kharkov,*

<sup>2</sup>*Institute of Electronics and Telecommunications of Rennes, UMR CNRS 6164, University of Rennes 1, Rennes,*

<sup>1</sup>*Ukraine*

<sup>2</sup>*France*

## 1. Introduction

Evolution strategies, implemented in numerical codes, provided researchers with powerful optimization tools capable of finding optimal solutions for a variety of real-world problems. One of the most popular representatives of this family is the Genetic Algorithm (GA) (Barricelli, 1957), which has already been well recognized by the electromagnetic (EM) community (Haupt, 1995; Johnson & Rahmat-Samii, 1997; Weile & Michielssen, 1997; Rahmat-Samii & Michielssen, 1999; Haupt & Werner, 2007; Hoorfar, 2007).

The most attractive features of GA, which are also intrinsic to other evolutionary algorithms (EA), are as follows: they can be applied given limited information about the problem, they do not require initial guesses, and they are able to produce non-intuitive solutions. These capabilities are provided thanks to a two-fold strategy that combines a stochastic global exploration and a local exploitation implemented in the form of an iterative modification and reproduction of already known individuals. The key to success here is the effective division of labour between both.

Different EAs use different ways of balancing between the global and local search, based on the corresponding evolutionary model. A possible bottleneck here is that as soon as a new evolutionary model is introduced, one starts thinking in terms and within the bounds dictated by the analogy used, whereas these bounds are not absolute. They arise from specific tasks addressed by nature and therefore are inherently adapted to “boundary conditions” of specific scenarios. For instance, genetic strategy (Barricelli, 1957) is oriented towards a huge population of diverse individuals and almost unbound time frames. This strategy is rather slow but it aims at the ultimate goal of finding the very best of all possible solutions. This is in contrast to the ant colony (Colomi et al., 1991) and particle swarm (Kennedy & Eberhart, 1995) strategies that naturally serve finding a reasonably good solution during a limited timeframe. Nevertheless, in spite of the formal differences, all population-based EAs have much in common. They share the same goal of finding the global extremum among multiple local ones; they operate with subsets of trial solutions;

they rely on stochastic decision-making mechanisms; and they manipulate with the probability in order to guide the optimization process. The latter is controlled by the selected evolutionary model and evaluation principles, which define the chances of each individual to survive and reproduce in later generations. In addition, all EAs favour improvement of the whole population instead of promoting a single leader. This protects EAs from being trapped in local minima but handicaps the solution convergence rate and may cause stagnation at the later stage of optimization. This also makes sharp distinction between stochastic global techniques and deterministic local ones. Contrary to global ones, local search techniques use cost function gradients to govern the search process. Although criticized for being slow and dependent on the initial guess, local techniques are the only means of learning (Paszkowicz, 2006; Elmihoub et al., 2006). This makes them complementary to EAs and highlights the importance and strong potential of hybridized optimization algorithms, which combine elements of different evolutionary and deterministic models. Such algorithms have been strongly advocated in a number of papers, e.g. (Renders et al., 1996; Haupt & Chung, 2003; Elmihoub et al., 2006); nevertheless they are still rarely used in electromagnetics.

In this chapter, we provide an insight into the general logic behind selection of the GA control parameters (Section 2), discuss the ways of boosting the algorithm efficiency (Section 3), and finally introduce a simple global-local hybrid GA capable of fast and reliable optimization of multi-parameter and multi-extremum functions (Section 4). The effectiveness of the proposed algorithm is demonstrated by numerical examples, namely: synthesis of linear antenna arrays with pencil-beam and flat-top patterns (Section 5).

## 2. Global and local skills of genetic algorithms

Genetic algorithm (same as any other population-based EA) can be compared with a two-handed machine that uses one hand for random selection of individuals from a given pool of possible solutions and another hand for “cheating” the first one. The cheating is realized in the form of manual weighting the probability of a favourable event to happen. In particular, this is used to promote local search in the neighbourhood of previously found fittest solutions. Different evolutionary strategies incorporate different cheating capabilities whose strength is adjusted by varying algorithms control parameters. A few examples provided below illustrate how the GA skills can be adapted in the favour of either global or local search. Similar mechanisms can be easily identified in other EAs as well.

The terminology used hereafter is borrowed from (Johson & Rahmat-Samii, 1997) whereas a recommended source for detailed information about the properties of GA operators is (Haupt & Haupt, 2004).

In most cases, GA starts with a random seed of a finite number of individuals that constitute the initial population. At this moment any solution within the given design space can be selected and probability of this event to occur equals reverse of the pool volume (the total number of all possible solutions or combinations of parameters). The situation changes for the second and subsequent generations. Here, the number and locations of potential offspring are limited and determined by the previous population. This happens because offspring always preserve properties of parents (at least partially) and therefore they can occur in a limited number of locations dictated by their parents and the crossover/mutation

schemes used (Haupt & Haupt, 2004, Chapter 5). Thus, except for the initial step, one never deals with a complete pool of solutions. Instead, as soon as the initial population is randomly generated, one has access only to a subset which includes the current population and its potential offspring. During optimization this trial subset is gradually transformed in a way to include individuals from the most promising regions of the original design space.

To reach this goal, the following sequence of operations is performed at each step of the optimization process. First, the trial subset is expanded by adding new individuals produced via reproduction of already known ones or randomly generated. Then, the quality of new individuals is evaluated and all individuals are ranked according to their cost function values. Finally, the worst individuals are discarded. Hopefully, each iteration moves search towards a region holding the global extremum, thanks to the continuous discarding of individuals which belong to less promising parts of the original design space.

Convergence of this process depends on two factors: (i) the rule that defines interrelation between a population at hand and the corresponding trial subset, and (ii) the criteria used for estimating the individuals' quality, which affects chances of individuals to survive and reproduce. To boost convergence, an additional weighting of individuals in populations can be introduced based on the cost function value or some additional criterion, e.g. taboo (Ji & Klinowski, 2009) or penalty (Paszkowicz, 2009) principles.

The influence of different factors on the GA convergence rate is discussed below.

The role of the population size seems obvious: the larger the size, the more uniform the exploration (or sampling) of the design space is provided. On the other hand, an oversized population slows convergence due to degeneracy of individuals that causes a strong offspring dispersion. This hinders local search because offspring often escape the parent solution's basin. Therefore some optimal size always exists, although it depends on the landscape of the fitness function and properties of the GA operators used. Useful hints on this subject are given in (Linden, 1999).

Two main GA operators are crossover and mutation. They define the size and structure of the trial subset accessible at each step. For instance, if a single-point crossover is implemented in a binary GA, all possible offspring are limited to a few choices that occur along the lines coinciding with the edges of a hyper rectangle with two parents on opposite vertices. The size of this hyper rectangle depends on the distance between parents, whereas sampling density is proportional to the number of crossing points. For instance, the number of potential offspring increases if a double-point crossover is used, whereas a uniform sampling can only be provided if uniform crossover is implemented. For numerical examples the reader could refer to (Haupt & Haupt, 2004, Chapter 5).

Mutation operators also suffer from the problems related to non-uniform sampling (Haupt & Haupt, 2004; Paszkowicz, 2006). Although usually positioned as a source of new genetic material, in practice a binary mutation operator (similarly to the crossover operator) is capable of producing only a finite number of offspring, called mutants, confined to orthogonal lines parallel to the axes. Furthermore, the strength of mutation (spread of mutants' locations) cannot be controlled easily because the change of a single bit in the binary string used for storing optimization parameters (called chromosome) has a different impact on the parameters values depending on the bit position. A partial remedy for the latter is in the Gray coding (Taub & Schilling, 1986) or a continuous representation of

variables. But this remedy has a side effect because a non-uniform distribution of potential offspring produced by standard binary crossover and mutation operators has its own hidden sense: it provides denser distribution of potential offspring and mutants in the neighbourhood of their parents that enhances local skills of the algorithms.

Finally, a weighting mechanism is implemented in GA in the form of selection principles that define chances of the fittest individuals to survive and reproduce. Among the popular selection principles (Johnson & Rahmat-Samii, 1997; Haupt & Haupt, 2004), the strongest one (local-search oriented) is the roulette wheel with cost-function weighting. This scheme heavily promotes the best individuals and stimulates local search in their neighbourhood. This increases chances for population degeneracy and thus may negatively affect optimization process by premature convergence to a local extremum. To counterbalance this, a permanent inflow of new genetic material should be provided. This is usually done by choosing a larger population size, higher mutation rate, and/or periodic injection of randomly generated individuals. Sharing the GA searching efforts among several most promising individuals can be realized if a so-called tournament principle is used. This scheme deals with randomly selected sub-groups (instead of the whole population) and in such a manner improves chances of next-to-the-best individuals to survive and reproduce in later generations.

Summarizing the discussion, we would like to highlight the following. Although there are many factors affecting the GA performance, they all serve the same reason: to effectively share the algorithm efforts between the stochastic global exploration and local exploitation. Thus variation of any control parameter can be considered as a contribution towards the enhancement of either global or local skills of the algorithm. This simplification helps a lot when adjusting GA control parameters for a specific problem at hand. Finally, one should remember that GA control parameters constitute a system of counterbalances; therefore variation of any parameters usually requires some adjustment of the others (e.g. a smaller population size should be compensated by a larger inflow of new genetic material, etc.). The additional opportunities for boosting the algorithm efficiency are discussed in the following section.

### **3. On boosting the algorithm efficiency**

There are two complementary approaches for boosting the performance of an optimization algorithm. The first one is based on adaptation of the algorithm control parameters during optimization. The second one is based on the amelioration of the design space landscape. The advantages proposed by each approach are summarized below.

#### **3.1 Adaptation of the algorithm control parameters**

As it was discussed in Section 2, selection of the algorithm control parameters (e.g. crossover and mutation schemes and rates) and selection mechanisms affects the global and local skills of GAs. Thus, adaptation of these parameters during simulations enables one to gradually shift the search efforts from the global exploration to local exploitation. The adaptation can be carried out based on different time-varying quantities such as iteration number, population diversity, solution quality, or relative improvement. The numerical examples revealing the capabilities of this approach, as well as an exhaustive review of the literature on this subject, can be found in (Eiben et al., 1999; Boeringer et al., 2005).

### 3.2 Hybridization of different optimization techniques

An additional degree of freedom for adjusting the algorithm capabilities for global and local search can be gained via hybridization of different optimization techniques. Both global-global and global-local hybrids have been reported so far. The former are typically used to compensate for intrinsic weak points of evolutionary algorithms that come from their natural analogues (Robinson et al., 2002; Salhi & Queen, 2004; Paszkowicz, 2006; Grimaccia et al., 2007); whereas the latter are used for boosting the algorithm efficiency at the later stage of optimization and/or learning purposes (Chelouah & Siarry, 2003; Ishibuchi, 2003; Haupt & Chung, 2003; Elmihoub et al., 2006; Paszkowicz, 2006; Ngo et al., 2007; Quevedo-Teruel et al., 2007; Boriskin & Sauleau, 2011a).

The great potential of global-local hybrids is explained by the distinction and complementarity between the local and global search techniques. Both these features come from the decision-making mechanism implemented in local and global search techniques. The former defines direction where to go (based on the cost function gradient), whereas the latter relies on the elimination principle implemented in the form of a successive dismissal of less promising individuals. In such a way, EAs give preference to the gradual improvement of the entire population instead of promoting a single individual. This is contrary to the local gradient-based algorithms that start from a given initial guess and perform a down-hill movement towards a nearest minimum following the shortest trajectory. Finally, local techniques are the only means of learning. If hybridized with EAs, they can supply the latter with information about cost function gradients, which can be used for introducing additional weights for individuals with better potential for improvement.

The aforementioned tactics are not new. Their pros and cons are well described in (Elmihoub et al., 2006). Nevertheless, the importance of global-local hybridization is still often underestimated, although the marriage of two is a simple and elegant way to achieve the optimal balance between the global and local skills of GA (or another EA).

### 3.3 Multi-extremum search capabilities

An important feature of EAs is their intrinsic capability for the multi-extremum search. On the way to an optimal solution, EAs sequentially investigate a number of local extrema. Most often, this information is lost as soon as the corresponding individuals are discarded due to lower fitness values or achievement of a stopping criterion. However, some of the next-to-the-best individuals can belong to basins of optimal solutions (or at least the most feasible ones due to some technical constraints not accounted for in the mathematical model). Therefore, search for multiple extrema and proper usage of the optimization history opens the door for development of advanced global optimization algorithms (Moret et al., 1998; Chelouah & Siarry, 2003).

### 3.4 Modification of the design space landscape

A deciding factor for the solution convergence rate of any optimization problem is a landscape of the corresponding fitness function. Usually it is accepted as something predefined and therefore invariable, although this is not true. Definitely, above all the landscape depends on the problem at hand, but it can also be affected by the style of parameter representation. This includes the chromosome structure (Weile & Michielssen,

1997) and parameter encoding (ODonnell et al., 2003; Boriskin & Sauleau, 2011a). Therefore adjustment of the fitness function landscape can also be considered as a part of the optimization strategy. Indeed, the landscape can be easily modified via mapping (not obviously identical) accounting for the problem-specific information. If properly done, a new design space becomes more optimization-friendly thanks to a reduced dimensionality, smaller size, and/or smoother landscape.

Note that mapping does not require any modifications of the cost-function itself. This constitutes an important distinction compared to (Ioan et al., 1998) and (Farina & Sykulski, 2001), where it was proposed to reduce the computational load by replacing the original cost function by a simplified or approximated cost function. Instead, the mapping only assumes a change in the way of storing optimization parameters that facilitates integration of such an algorithm with external electromagnetic solvers, even those operating in a “black box” mode. Such an approach can be especially effective if many identical parameters are involved, e.g. geometrical parameters describing an antenna topology (Fernandes, 2002; Robinson et al., 2002; Godi et al., 2007; Boriskin et al., 2010; Rolland et al., 2010; Boriskin & Sauleau, 2011b) or phase/amplitude weights in the antenna aperture (Johnson & Rahmat-Samii, 1997; Pérez & Basterrechea, 2007). A few examples of mapping realized on the basis of different encoding schemes are given in Section 5.

### 3.5 Summary: Recipe for an efficient global optimizer

Summarizing the discussion, a general recipe for an efficient global optimization algorithm can be outlined as follows. Start with an EA, whose control parameters are selected in a way to promote an exhaustive global exploration. Then gradually shift the algorithm efforts in the favour of the pseudo-local search. This can be done via a gradual adaptation of the algorithm control parameters and/or via a switching between different selection mechanisms. In addition, the learning capabilities of local optimizers can be used for determining the improvement potentials of selected individuals, based on the cost function gradients. This knowledge can be used to guide the selection process. If the optimization process shows signs of stagnations, a switching between different EAs can be performed. Finally, top- $N$  individuals (if possible, selected from different solution basins) should be extracted and fine-tuned using a local gradient-based optimizer. Such a complementary strategy offers an optimal division of labour between the global and local search, as well as reaching the very bottom of all identified extrema. The latter is very useful for collecting the problem-specific information. To illustrate the discussion, a simple global-local hybrid GA is introduced in Section 4.

## 4. Hybrid genetic algorithm

In this section we present a global-local hybrid genetic algorithm (HGA) built in line with recommendations outlined in Section 3.5. The algorithm combines a binary GA and a steepest descent gradient (SDG) algorithm. The former is used for the global exploration, whereas the latter is used for tuning the top- $N$  individuals produced by GA (hereafter labelled as “GA top-runners”) and considered as initial guesses for local optimization.

A distinctive feature of the proposed algorithm is that it aims not only at a single best solution but instead identifies a given number of GA top-runners that are investigated at the

later stage with the aid of the gradient-based SDG optimizer. Such a two-step approach enables us to reduce significantly the GA stagnation period at the later stage of optimization and also to guarantee achievement of the very bottom of multiple extrema whose basins are identified by GA. The final solution is then selected among those produced by SDG.

A flowchart of the proposed HGA is shown in Fig. 1. The purpose of each block of the algorithm is discussed below.

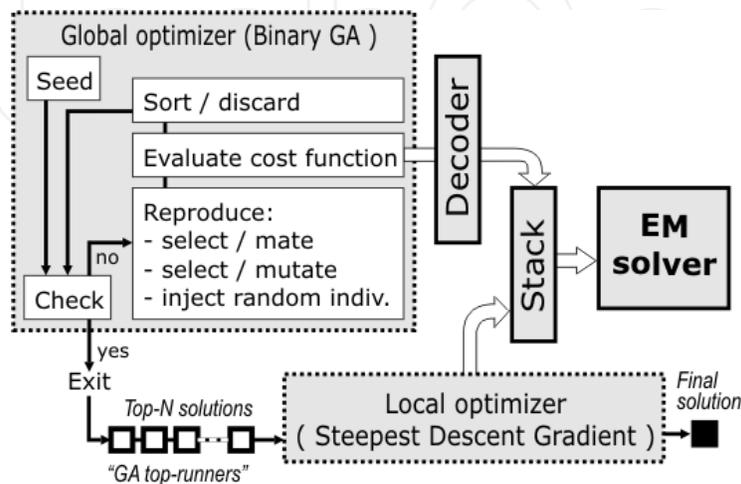


Fig. 1. Flowchart of the HGA

#### 4.1 Global optimizer

The global-local hybridization enables us to let GA concentrate mostly on the global exploration. For this purpose, GA parameters are selected in the following extreme manner that, on the one hand, enhances its global-search capabilities and, on the other hand, strongly promotes pseudo-local search around best individuals: small population size; high mutation rate; periodic injection of randomly generated individuals; no identical individuals (so-called twins) allowed; double-point crossover; uniform mutation. As a counterbalance in the favour of the GA local-searching capabilities, the elitism principle and the roulette wheel cost-function weighted selection mechanism are implemented. The logic behind such a selection of the GA control parameters was discussed in Section 3.

#### 4.2 Local optimizer

For the reported study, the local optimizer is used only for tuning a given number of GA top-runners. To simplify comparison with a standard binary GA, we make SDG algorithm move using the same mesh as for GA, where it is defined by the binary representation of optimization parameters. For simplicity, we disable the “learning function” of the SDG, which means that there are no additional weights introduced in GA selection mechanism and there is no feedback between SDG and GA algorithms after switching between the two.

#### 4.3 Decoder

An important feature of the proposed HGA is a decoder, which is used for communication between GA and EM solver. In contrast to a binary decoder which is an essential part of any

binary EA, this additional decoder is used for mapping between the original design space and a new one, which appears due to implementation of specific encoding schemes used for representation of optimization parameters. The decoder is not used for communication between the local optimizer and EM solver because at the final stage of optimization one needs access to the complete pool of parameter combinations corresponding to the original design space.

#### 4.4 Stack

Finally, to avoid the recalculation of the cost function for already known individuals, a stack has been implemented in the form of an array storing parameters of the recently evaluated individuals and their cost function values. This is in line with recommendations given in (Linden, 1999). The optimal size of the stack depends on the complexity of the optimization problem. Our experience shows that a stack with size of three to five populations is usually sufficient. The content of the stack can be updated cyclically: each time a new individual appears it replaces the oldest one in the stack.

#### 4.5 Summary

A combination of the aforementioned features guarantees a high efficiency and reliability of the proposed HGA when solving various optimization problems. It is worth being noted that the performance characteristics of HGA are boosted by letting each algorithm do what it is best suited for, rather than trying to push the optimization process by implementing some deterministic rules, which may cause a conflict with the stochastic nature of the evolutionary strategy. This makes the proposed algorithm very stable and universal. In addition, the performance of the algorithm is strongly facilitated by the amelioration of the design space landscape and elimination of redundant simulations. The effectiveness of the algorithm is demonstrated by solving two multi-parameter optimization problems, typical for EM synthesis (Section 5).

### 5. Linear antenna array synthesis using HGA

The optimization of antenna arrays has already become classics of the electromagnetic synthesis due to a simple formulation and practical importance, e.g (Haupt, 1995; Johnson & Rahmat-Samii, 1997; Weile & Michielssen, 1997; Rahmat-Samii & Michielssen, 1999; Isernia, et al., 2004; Boeringer et al., 2005; Haupt & Werner, 2007).

To illustrate the performance of the developed HGA, two simple linear array optimization problems are considered, namely phase-only optimization aimed at the minimum side-lobe level (Sections 5.1) and amplitude-phase optimization aimed at a flat-top beam pattern (Section 5.2). In both cases the HGA features are adjusted in a way to let GA perform an exhaustive global search, aiming to identify 10 top-runners to be used as initial guesses for the SDG algorithm. The control parameters are selected as follows: (i) double-point crossover, (ii) uniform mutation with linearly decreasing rate of 20% to 10%, (iii) cost-function-weighted roulette wheel selection mechanism, and (iv) permanent inflow of randomly generated individuals with a rate of 10%. Furthermore, we avoid twins which are replaced by randomly generated individuals each time when identical offspring appear. Finally, to preserve the continuous progress, a few best individuals (~5%) are stored from

previous generations (elitism principle). This set of parameters has been approbated on several standard test functions and found to be suitable for various optimization scenarios (these data are skipped for brevity).

In the reported study, we approbate three different encoding schemes, namely: “direct”, “relative”, and “envelope” ones. The former is a standard encoding scheme when optimization parameters are stored as they are. In the relative scheme, optimization parameters are encoded as differences between neighbours. For most practical cases, this difference does not exceed a half of the parameter variation range. Thus the search domain for new parameters can be reduced by a factor of 2, which means reduction of the entire design space by  $2^N$  parameter combinations, where  $N$  is the number of optimization parameters. In case of the envelope encoding scheme, optimization parameters are represented using an envelope line, defined by a polynomial. For the current study the envelope line is constructed as a sum of a few Gaussians. This enables us to replace the original design space with  $N$  dimensions by a new one with  $3M$  dimensions, where  $3M$  parameters are the amplitude, central value, and half-width of each Gaussian, and  $M$  is the number of Gaussians used. An empirical rule for selecting the latter parameter is  $M = NINT(N/10)$ , where  $NINT$  returns the nearest integer value of the argument. For high-dimensional problems ( $N \geq 10$ ), the reduction of the design space becomes really significant, which strongly facilitates the search for the global extremum.

It is important to note, that in both non-direct encoding schemes the reduction of the design space is obtained via truncation of the original space according to some template defined by the encoding scheme used. Thus it is important to assure that this template “filters” poor solutions and preserve better ones. Definition of such a template is a tricky question. To some extent this is similar to guessing for a class of optimal solutions. It might look like the introduction of a template brings us back to a deterministic optimization scenario, criticised for its strong dependence on the quality of the initial guess, but it does not. As it will be shown below, the selected encoding schemes preserve flexibility sufficient for identification of optimal solution basins for various optimization problems. Once identified, these basins can be effectively studied using a gradient-based local optimizer.

### 5.1 Test-case 1: Low-sidelobe via phase tapering

The first test problem is the synthesis of a linear array aimed at reduction of the array factor (AF) side-lobe level (SLL) via phase weights optimization. A symmetrical linear array of 31 equally spaced feeds with uniform amplitude weights is considered. The cost function returns a square of difference between the AF SLL for a given phase taper and its desired value. The elements of the array are spaced  $0.5\lambda$  apart and phase weights are symmetric about the centre of the array with the central element having a phase of zero. Quantization of the phase weights is 4-bit. A trustable reference solution for this test problem can be found in (Haupt, 2007), whereas the one found by the proposed HGA is shown in Fig. 2.

To assess the efficiency of HGA, its performance is superimposed with that of a binary GA whose control parameters are selected in line with general recommendations (Johnson & Rahmat-Samii, 1997). For convenience, parameters of both algorithms are summarized in Table 1.

A typical run of HGA is illustrated in Fig. 3. Here the best and average cost function values are denoted by solid thick lines, whereas cost function values of each individual at each iteration step are denoted by circles. Switching between GA and SDG occurs after 40 generations. The family of ten colour lines shown after 40<sup>th</sup> generation illustrates the process of tuning the ten GA top-runners by means of the SDG algorithm.

Control parameter	HGA	GA
Population size	50	50
Probability of crossover	Linearly increasing: 65 → 75 %	90 %
Probability of mutation	Linearly decreasing: 20 → 10 %	5 %
Inflow of random individuals	10 %	--
Number of preserved best individuals	5 %	5 %
Selection mechanism	Cost-function weighted roulette wheel	Tournament with sub-population size of 10%
Stopping criterion	40 iterations for GA + as much as needed for SDG	200 iterations

Table 1. Control parameters of HGA and GA algorithms

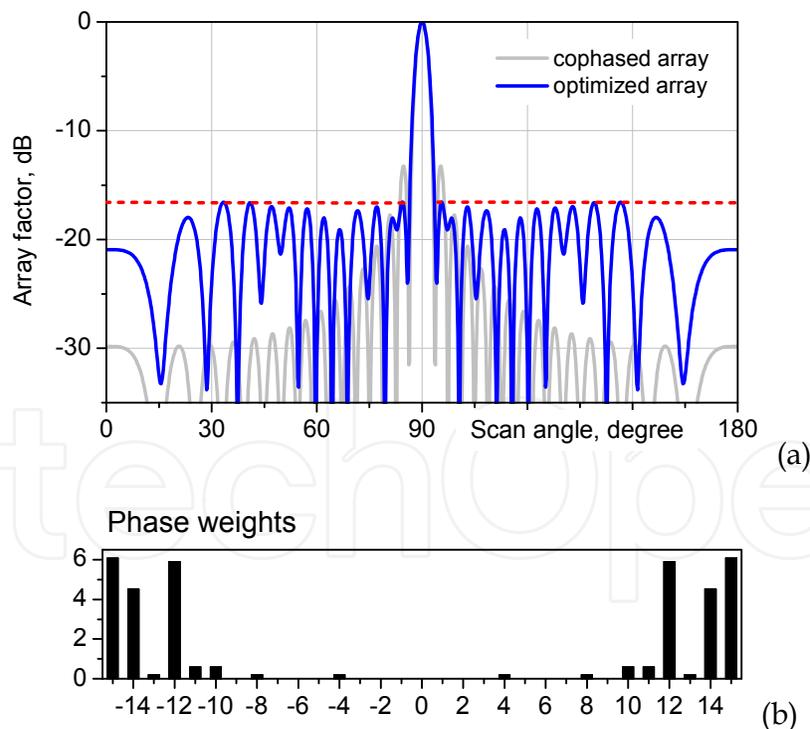


Fig. 2. Optimal solution found by HGA: (a) Array factors of the cophased and optimized arrays, (b) phase weights corresponding to the optimized solution.

The same optimization run represented in terms of AF SLL is shown in Fig. 4. Here it is superimposed with the curve which represents the averaged solution produced by a standard GA. As we can see, the standard GA quickly reaches the AF side-lobe level of

approximately -15 dB and spends twice more time to improve solution for another half dB. Such behaviour is typical for GA that continues to explore the entire design space (more or less exhaustively) during all simulation time. This protects GA from “hanging” in local minima but slows down the convergence rate at the later stage of optimization. The proposed HGA is free from this drawback because here GA is used only to identify the optimal solution basins whereas their exploitation is performed in a straight-forward manner using the SDG algorithm, whose performance is based on the cost function gradient. Indeed, we can see that at the initial stage, while HGA is focused on the global exploration, standard GA performs better. Nevertheless, as soon as HGA switches for local optimizer, it catches up and outruns GA in a very few steps.

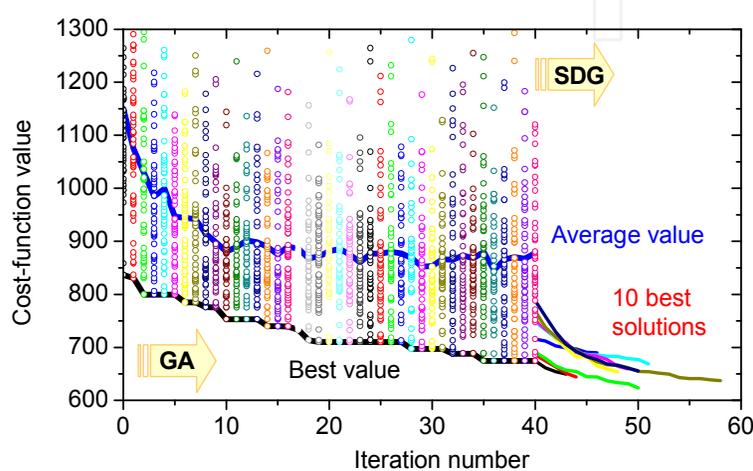


Fig. 3. A typical run of HGA when applied for the linear array synthesis aimed at minimum AF SLL. The AF pattern of the optimized array are shown in Fig. 2.

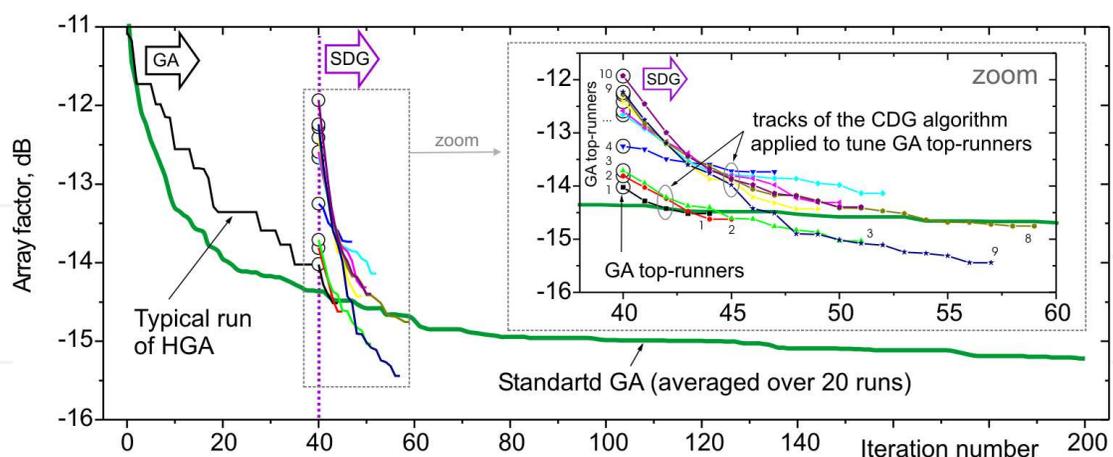


Fig. 4. Comparison between HGA and a standard GA when applied for the linear array synthesis aimed at minimum AF SLL. The HGA run is the same as shown in Fig. 3 but represented in terms of AF SLL. The GA curve is averaged over 20 trials. The inset zooms in on the local optimization stage of HGA.

As already explained, in the proposed hybrid algorithm, GA is used to generate a few best solutions (top-runners) to be then refined using the SDG algorithm. To this end, it is interesting to note that refinement of only the best GA solution (let's label it as a “GA

winner”) usually does not give much advantage. This is because the winner often belongs to a wide and gently slopping basin, which is optimization-friendly and thus already well examined by GA, whereas most promising solutions are usually located on sides of deep and narrow valleys whose exploration using GA is troublesome. An illustration to such a situation is given in Fig. 4. As we can see, the bottom of the winner’s solution basin is reached in four iterations with no significant improvement achieved, whereas the 3-rd, 8-th, and 9-th top-runners demonstrate much better improvement. In particular, refinement of the 9-th top-runner resulted in SLL of -15.4 dB which is approximately 1 dB lower than the final solution found with the GA winner taken as the initial guess.

To get more statistical data, 20 trials have been performed with the same set of parameters (Fig. 5). The obtained data clearly evidence that GA winners rarely appears to be the best initial guess for local search. Therefore evaluation of several top-runners is strongly recommended in order not to waste GA efforts in a hunt for a single winner, which often belongs to a local solution basin. Note that this recommendation remains valid for all tree encoding schemes.

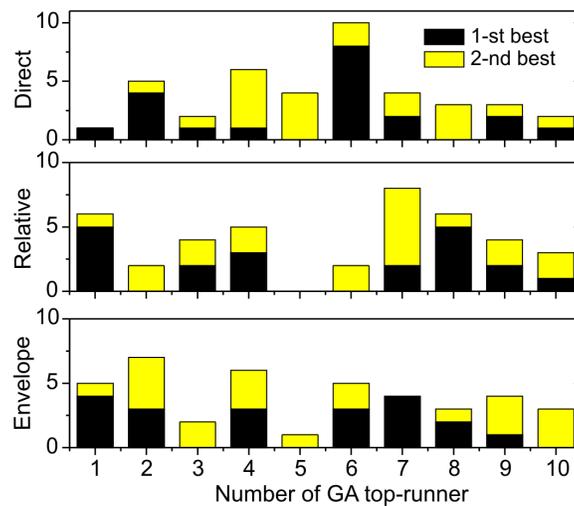


Fig. 5. Number of trials when each of ten GA top-runner, tuned by SDG, finished with the best or second-best result. The total number of trials is 20.

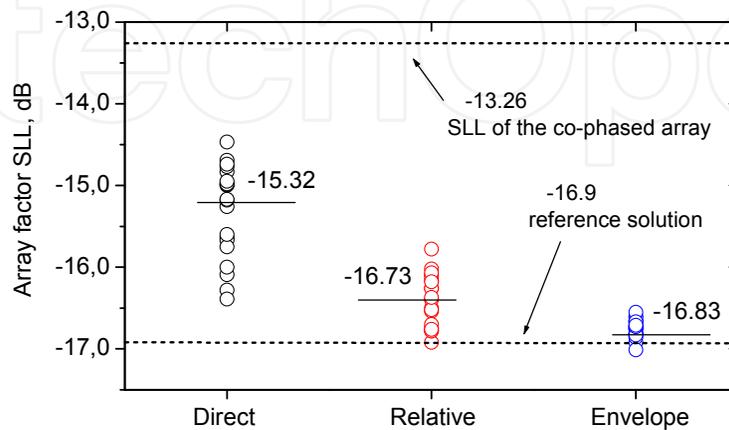


Fig. 6. The final solutions found by HGA in 20 trials applied with three different encoding schemes. The reference solution is borrowed from (Haupt, 1997).

Finally, the impact of different encoding schemes on the algorithm performance is illustrated in Fig. 6. As one can see, the relative and envelope (i.e. Gaussian with  $M=2$ ) encoding schemes provide much better grouping of final solutions around an improved average value. The better quality of the final solution and the significant reduction of cost function evaluations (see Table 2) are of the primary importance for the EM synthesis because solution of direct EM problems is usually very time consuming.

Algorithm type	No. of optimization parameters	No. of bits per parameter	GA encoding scheme	No. of iterations * (SDG)	Cost function evaluations * (GA + SDG)	Final solution * (SLL, dB)
GA	15	4	Direct	--	10000	-15.18
HGA	15	4	Direct	9.6	3895	-15.32
HGA	15	3	Relative	10.7	4216	-16.73
HGA	6	4	Gauss ( $M = 2$ )	11.0	4300	-16.83

\* Data averaged over 20 trials

Table 2. Statistical data: HGA vs. standard GA.

## 5.2 Test-case 2: Flat-top beam via complex weighting

The second test problem is the synthesis of a linear array with a flat-top beam via joint phase and amplitude weights optimization (Fig. 7). This time an even symmetrical linear array of 30 isotropic feeds spaced half lambda apart ( $d = 0.5\lambda$ ) is considered. The weights have 4-bit quantization and are symmetric about the centre of the array with the central elements having phase of zero. The pattern template is defined as follows: the flat-top beam parameters are  $\theta_1 = 28^\circ$  and  $\theta_2 = 30^\circ$ , the ripples level in the main beam is restricted by  $F_1 = -2$  dB, and the highest allowed SLL is -20 dB. The cost function equals the sum of penalties charged for crossing the given corridor (Fig. 7b):

$$F = \sum_{k=1}^K \left[ (F_1 - F(\theta_k)) \Big|_{\theta_k \in [0, \theta_1] \cup F < F_1} + (F(\theta_k) - F_2) \Big|_{\theta_k \in [\theta_2, \pi/2] \cup F > F_2} \right], \quad (1)$$

where  $K$  is the number of sampling points,  $K = 90$ .

The radiation pattern of the optimized array and its amplitude/phase weights are shown in Fig. 8, and a reference solution is available in (Galan et al., 2011).

To demonstrate the efficiency of the HGA for the considered optimization problem, we compare its performance with that of a standard GA. Parameters of the algorithms are the same as shown in Table 1, except the following: (i) for HGA, the switching between GA and SDG occurs after 50 iterations, (ii) population size for the standard GA has been increased up to 200 individuals in order to compensate for the larger number of optimization parameters (i.e. total of 29, which corresponds to 15 amplitude weights and 14 phase weights). The number of trials has been also increased up to 100.

The statistical data presented in Fig. 9 clearly demonstrate that the proposed HGA significantly outperforms a standard GA in terms of the final solution quality even if the same direct encoding is used, whereas implementation of the advanced encoding schemes leads to further improvement of the stability in the algorithm performance evidenced by the improved quality and superior grouping of final solutions. Once again, the best efficiency is

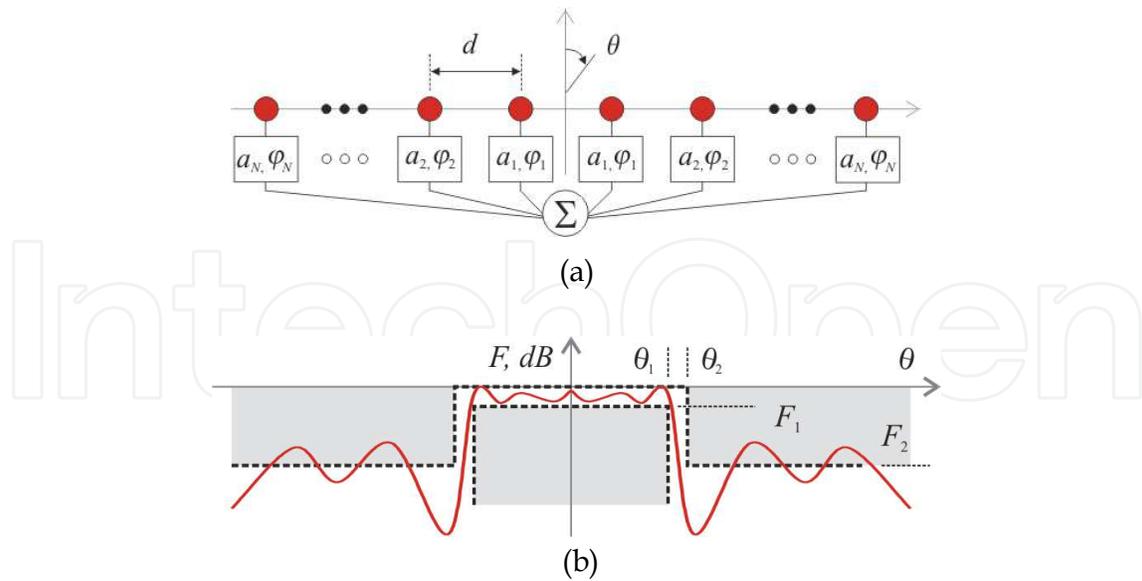


Fig. 7. Linear antenna array under consideration: (a) geometry and notations of the problem, (b) template for the flat-top beam radiation pattern.

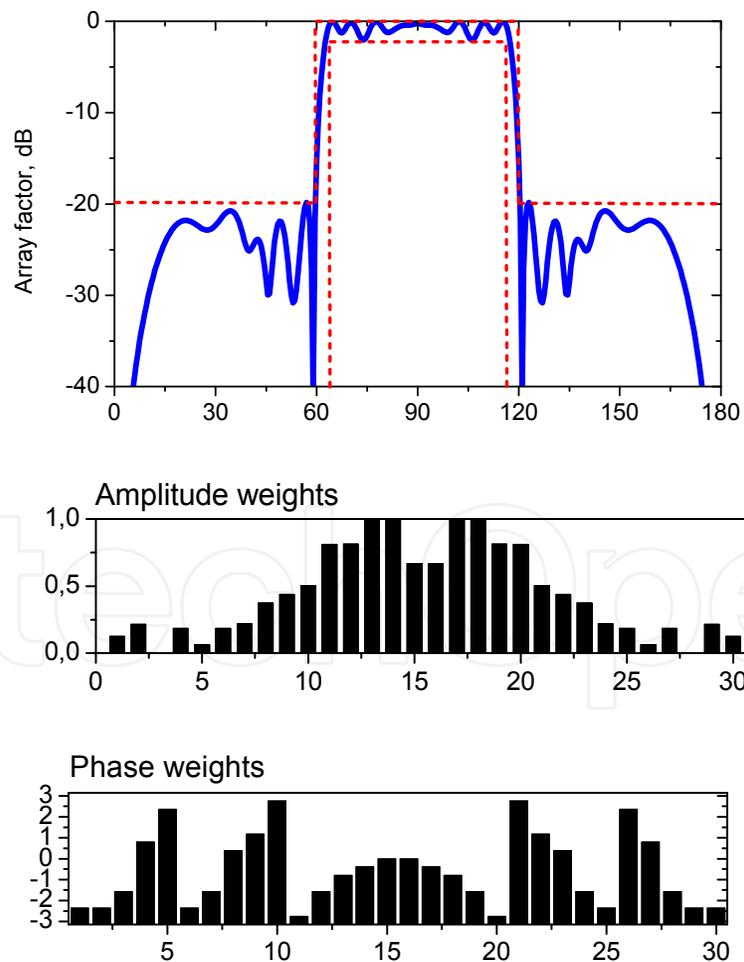


Fig. 8. Optimal solution found by HGA: array factor of the 30-element linear array with optimized phase and amplitude weights. Dashed line denotes the pattern template.

observed for the envelope encoding scheme. Note that this time parameters of two types are involved (i.e. phase and amplitude), therefore the envelope lines for the phase and amplitude weights are reconstructed independently, which explains the increase of the number of optimization parameters up to 12 (two envelope lines build of two Gaussians each). Finally, it is worth being mentioned that the overall computational time (measured in terms of a number of cost function evaluations) is nearly the same for all runs (Table 3), which means that the improved performance is achieved thanks to a more effective optimization strategy.

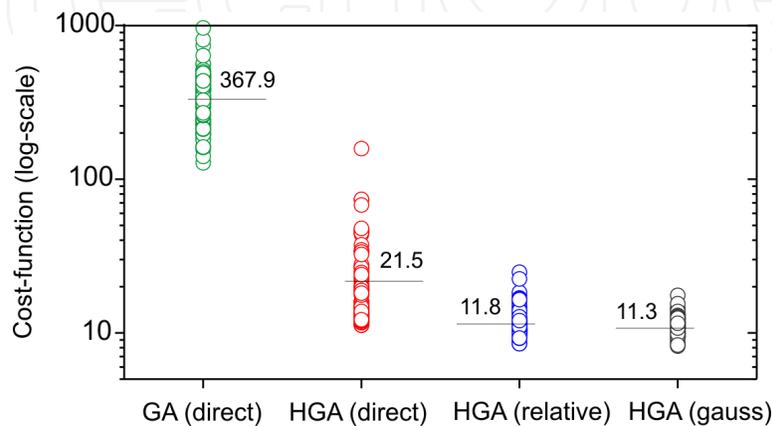


Fig. 9. Final solutions found in 100 trials by a standard GA and HGA with three different encoding schemes. The average values of the final solutions produced are shown nearby.

Algorithm type	No. of optimization parameters	No. of bits per parameter	GA encoding scheme	No. of iterations * (SDG)	Cost function evaluations * (GA + SDG)	Final solution * (cost function)
GA	29	4	Direct	--	20000	367.9
HGA	29	4	Direct	26	17062	21.5
HGA	29	3	Relative	34	21541	11.8
HGA	12	4	Gauss (M = 2)	37	23226	11.3

\* Data averaged over 100 trials

Table 3. Statistical data: HGA vs. standard GA.

## 6. Conclusion

In this chapter, the factors affecting the performance of genetic algorithms have been discussed and a few hints on boosting the algorithm efficiency have been provided. In particular, three complementary options have been outlined, namely: adjustment of the algorithm control parameters, hybridisation of different global and local algorithms, and amelioration of the design space implemented in the form of mapping. The discussion has been illustrated by presentation of a global-local hybrid genetic algorithm, whose efficiency in solving multi-parameter problems has been demonstrated through numerical examples.

The main benefits achieved thanks to hybridization of a binary GA and a SDG algorithms are as follows: (i) improved convergence rate, (ii) better quality of the final solution, and (iii) the possibility to investigate multiple local extrema during a single run of the algorithm. These features are of the primary importance for the electromagnetic synthesis. Although the optimal values of the algorithm control parameters may vary for different optimization problems, the general recommendations regarding the logic behind the selection of these parameters are applicable for various optimization scenarios and different evolutionary algorithms.

## 7. Acknowledgment

This work was supported jointly by Ministry of Science and Education (MSE), Ukraine and Ministère des Affaires Étrangères et Européennes (MAEE), France in the frame of the DNIPRO Program (project N° 24748UH).

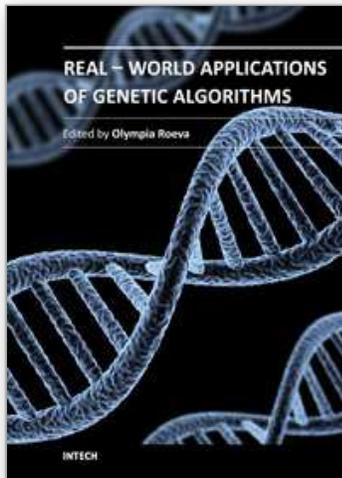
## 8. References

- Barricelli, N.A. (1957). Symbiogenetic evolution processes realized by artificial methods, *Methodos*, Vol. 9, pp. 143–182.
- Boeringer, D.W.; Werner, D.H. & Machuga, D.W. (2005). A simultaneous parameter adaptation scheme for genetic algorithms with application to phased array synthesis, *IEEE Tran. Antennas Propag.*, Vol. 53, No. 1, pp. 356–371.
- Boriskin, A.V.; Balaban, M.V., Galan, A.Yu. & Sauleau, R. (2010). Efficient approach for fast synthesis of phased arrays with the aid of a hybrid genetic algorithm and a smart feed representation, *IEEE Int. Symp. on Phased Array Systems and Technol.*, Boston, MA, pp. 827–832.
- Boriskin, A.V. & Sauleau, R. (2011a). Recipe for an efficient hybrid genetic algorithm, *J. Telecommunications and Radio Engineering*, vol. 70, no. 13, pp. 1143–1158.
- Boriskin, A.V. & Sauleau, R. (2011b). Numerical investigation into the design of shaped dielectric lens antennas with improved angular characteristics, *Progress in Electromagnetics Research B*, Vol. 30, pp. 279–292.
- Chelouah, R. & Siarry, P. (2003). Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multim minima functions, *Eur. J. Operational Research*, Vol. 148, No. 2, pp. 335–348.
- Coloni, A. ; Dorigo M. & V. Maniezzo, (1991), Distributed optimization by ant colonies, *Proc. 1<sup>st</sup> Eur. Conf. On Artificial Life*, Paris, France, pp. 134–142.
- Eiben, A.E.; Hinterding, R. & Michalewicz, Z. (1999). Parameter control in evolutionary algorithms, *IEEE Trans. Evol. Comput.*, Vol.3, pp. 124–141.
- Elmihoub, T.; Hopgood, A. A.; Nolle, L. & Battersby, A. (2006). Hybrid genetic algorithms - a review. *Engineering Letters*, Vol. 13, pp. 124 - 137.
- Farina, M. & Sykulski, J. K. (2001). Comparative study of evolution strategies combined with approximation techniques for practical electromagnetic optimization problems. *IEEE Trans. Magn.*, Vol. 37, No. 9, pp. 3216–3220.
- Fernandes, C.A. (2002) *Shaped-beam antennas*, Chapter 15 in *Handbook of Antennas in Wireless Communications*, L.C. Godara, Ed., CRC Press, New York.

- Galan, A.Y., Sauleau, R. & Boriskin, A.V. (2011). Parameter selection in particle swarm optimization algorithm for synthesis of linear arrays with flat-top beams, *J. Telecommunications and Radio Engineering*, Vol. 70, No. 16, pp. 1415–1428.
- Godi, G.; Sauleau, R.; Le Coq, L. & Thouroude, D. (2007). Design and optimization of three-dimensional integrated lens antennas with genetic algorithm, *IEEE Trans. Antennas Propag.*, Vol. 55, No. 3, pp. 770-775.
- Grimaccia, F.; Mussetta, M. & Zich, R.E. (2007). Genetical swarm optimization: self-adaptive hybrid evolutionary algorithm for electromagnetics, *IEEE Trans. Antennas Propag.*, Vol. 55, No. 3, pp. 781-785.
- Hoorfar, A. (2007). Evolutionary programming in electromagnetic optimization: a review, *IEEE Trans. Antennas Propag.*, Vol. 55, No. 3, pp. 523-537.
- Haupt, R.L. (1995). An introduction to genetic algorithms for electromagnetics, *IEEE Antennas Propag. Mag.*, Vol. 37, No. 2, pp. 7-15.
- Haupt, R. & Chung, Y.C. (2003). Optimizing backscattering from arrays of perfectly conducting strips, *IEEE Trans. Antennas Propag.*, Vol. 45, No. 2, pp. 26-33.
- Haupt, R.L. & Haupt, S.E. (2004). *Practical genetic algorithms*, 2nd ed., Wiley & Sons, ISBN 0-471-45565-2, Hoboken NJ.
- Haupt, R.L. & Werner, D.H. (2007). *Genetic algorithms in electromagnetics*, John Wiley & Sons, New York.
- Haupt, R.L., (2007), Antenna design with a mixed integer genetic algorithm, *IEEE Trans. Antennas Propag.*, Vol. 55, No. 3, pp. 577-582.
- Ioan, D.; Ciuprina, G. & Dumitrescu, C. (1998) Use of stochastic algorithms for distributed architectures in the optimization of electromagnetic devices. *IEEE Trans. Magn.*, Vol. 34, No. 9, pp. 3000–3003.
- Isernia, T.; Pena, F.J.A.; Bucci, O.M.; D'Urso, M.; Gomez, J.F. & Rodriguez, J.A. (2004). A hybrid approach for the optimal synthesis of pencil beams through array antennas, *IEEE Trans. Antennas Propag.*, Vol. 52, No. 11, pp. 2912-2918.
- Ishibuchi, H.; Yoshida, T. & Murata, T. (2003). Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling, *IEEE Trans. Evol. Comput.*, Vol. 7, No. 2, pp. 204-223.
- Ji, M. & Klinowski, J. (2009). Taboo evolutionary programming: a new method of global optimization, *Proc. Royal Society A*, Vol. 462, pp. 3613-3627.
- Johnson, J.M. & Rahmat-Samii, Y. (1997). Genetic algorithms in engineering electromagnetics, *IEEE Antennas Propag. Mag.*, Vol. 39, No. 4, pp. 7-21.
- Kennedy, J. & Eberhart, R.C. (1995). Particle swarm optimization, *Proc. IEEE Conf. Neural Networks IV*, vol. 52, pp. 1942-1948.
- Linden, D.S. (1999). Rules of thumb for GA parameters, GA monitoring, GA parameter optimization, and enhancement of GA efficiency, Chapter 3 in Rahmat-Samii, Y. & Michielssen, E., Eds. (1999). *Electromagnetic optimization by genetic algorithms*, John Wiley & Sons, New York.
- Moret, M.A.; Bisch, P.M. & Vieira, F.M.C. (1998). Algorithm for multiple minima search, *Phys. Rev. E*, Vol. 57, pp. R2535–R2538.
- Ngo, N.Q.; Zheng, R.T.; Ng, J.H.; Tjin, S.C. & Binh, L.N. (2007). Optimization of fiber bragg gratings using a hybrid optimization algorithm, *J. Lightwave Technol.*, Vol. 25, No. 3, pp. 799-802.

- O'Donnell, T.H.; Altshuler, E.E. & Best, S.R. (2003). The significance of genetic representation in genetic antenna design, *Proc. IEEE Int. Symp. Antennas Propag.*, Columbus, OH, Vol. 1, pp. 149-152.
- Quevedo-Teruel, Ó.; Rajo-Iglesias, E. & Oropesa-García, A. (2007). Hybrid algorithms for electromagnetic problems and the no-free-lunch framework, *IEEE Trans. Antennas Propag.*, Vol. 55, No. 3, pp. 742-749.
- Paszkwicz, W. (2009). Properties of a genetic algorithm equipped with a dynamic penalty function, *Computational Material Science*, Vol. 45, pp. 77-83.
- Paszkwicz, W. (2006). Properties of a genetic algorithm extended by a random self-learning operator and asymmetric mutations: a convergence study for a task of powder-pattern indexing, *Analytica Chimica Acta*, Vol. 566, pp. 81-98.
- Pérez, J.R. & Basterrechea, J. (2007). Comparison of different heuristic optimization methods for near-field antenna measurements, *IEEE Trans. Antennas Propag.*, Vol. 55, No. 3, pp. 549-555.
- Rahmat-Samii, Y. & Michielssen, E., Eds. (1999). *Electromagnetic optimization by genetic algorithms*, John Wiley & Sons, New York.
- Renders, J.M. & Flasse, S.P. (1996). Hybrid method using genetic algorithms for the global optimization, *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 26, No. 2., pp. 243-258.
- Robinson, J.; Sinton, S. & Rahmat-Samii, Y. (2002). Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna, *Proc. Inter. Symp. Antennas Propag.* Vol. 1, pp. 314-317.
- Rolland, A.; Ettorre, M.; Drissi, M.; Le Coq, L. & Sauleau, R. (2010). Optimization of reduced-size smooth-walled conical horns using BoR-FDTD and genetic algorithm, *IEEE Trans. Antennas Propag.*, Vol. 58, No. 9, pp. 3094-3100.
- Salhi, S. & Queen, N.M. (2004). A hybrid algorithm for identifying global and local minima when optimizing functions with many minima, *Eur. J. Operational Research*, Vol. 155, pp. 51-67.
- Taub, H. & Schilling D.L. (1986). *Principles of Communication Systems*, McGraw-Hill, NY.
- Weile, D.S. & Michielssen, E. (1997). Genetic algorithm optimization applied to electromagnetics: a review, *IEEE Trans. Antennas Propag.*, Vol. 45, No.3, pp. 343-353.

IntechOpen



## **Real-World Applications of Genetic Algorithms**

Edited by Dr. Olympia Roeva

ISBN 978-953-51-0146-8

Hard cover, 376 pages

**Publisher** InTech

**Published online** 07, March, 2012

**Published in print edition** March, 2012

The book addresses some of the most recent issues, with the theoretical and methodological aspects, of evolutionary multi-objective optimization problems and the various design challenges using different hybrid intelligent approaches. Multi-objective optimization has been available for about two decades, and its application in real-world problems is continuously increasing. Furthermore, many applications function more effectively using a hybrid systems approach. The book presents hybrid techniques based on Artificial Neural Network, Fuzzy Sets, Automata Theory, other metaheuristic or classical algorithms, etc. The book examines various examples of algorithms in different real-world application domains as graph growing problem, speech synthesis, traveling salesman problem, scheduling problems, antenna design, genes design, modeling of chemical and biochemical processes etc.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Artem V. Boriskin and Ronan Sauleau (2012). Hybrid Genetic Algorithm for Fast Electromagnetic Synthesis, Real-World Applications of Genetic Algorithms, Dr. Olympia Roeva (Ed.), ISBN: 978-953-51-0146-8, InTech, Available from: <http://www.intechopen.com/books/real-world-applications-of-genetic-algorithms/hybrid-genetic-algorithm-for-fast-electromagnetic-synthesis>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen