

MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS APPLIED TO MICROSTRIP ANTENNAS DESIGN

ALGORITMOS EVOLUTIVOS MULTIOBJETIVO APLICADOS A LOS PROYECTOS DE ANTENAS MICROSTRIP

Juliano Rodrigues Brianeze¹ Carlos Henrique da Silva-Santos¹ Hugo Enrique Hernández-Figueroa¹

Recibido 29 de junio de 2008, aceptado 21 de octubre de 2009

Received: June 29, 2008 Accepted: October 21, 2009

RESUMEN

Este trabajo presenta tres de los principales algoritmos evolutivos: Algoritmo Genético, Estrategia Evolutiva y Programación Evolutiva, aplicados al diseño de antenas de microlíneas (microstrip). Se realizaron pruebas de eficiencia de los algoritmos, considerando el análisis de los parámetros físicos y geométricos, tipo de evolución, efecto de generación de números aleatorios, operadores evolutivos y los criterios de selección. Estos algoritmos fueron validados a través del diseño de antenas de microlíneas basado en el Método de Cavidades Resonantes y permiten optimizaciones multiobjetivo, considerando ancho de banda, razón de onda estacionaria y permitividad relativa del dieléctrico. Los resultados óptimos obtenidos fueron confirmados a través del software comercial CST Microwave Studio

Palabras clave: Antenas Microstrip, electromagnetismo computacional, algoritmos evolutivos, optimización multiobjetivo, simulación computacional.

ABSTRACT

This work presents three of the main evolutionary algorithms: Genetic Algorithm, Evolution Strategy and Evolutionary Programming, applied to microstrip antennas design. Efficiency tests were performed, considering the analysis of key physical and geometrical parameters, evolution type, numerical random generators effects, evolution operators and selection criteria. These algorithms were validated through design of microstrip antennas based on the Resonant Cavity Method, and allow multiobjective optimizations, considering bandwidth, standing wave ratio and relative material permittivity. The optimal results obtained with these optimization processes, were confirmed by CST Microwave Studio commercial package.

Keywords: Microstrip antennas, computational electromagnetism, evolutionary algorithms, multiobjective optimization, computing simulation.

INTRODUCTION

Nature-inspired algorithms have attracted attention in many knowledge areas. In this computing field, biologically inspired (evolutionary) systems are the oldest and, maybe, the most popular [1]. They are very suited for problems with high nonlinearities and several parameters.

These algorithms have shown clearly notable results in numerical optimizations in computational electromagnetism, such as bandwidth improvement [2-3], which enables them to be extensively applied in RF and microwave communication

systems [4-5]. There are also other reported applications in photonics, like optimization and analysis of new photonic effects and devices [6-7].

This work has focused on Microstrip Antennas (MSAs) design, with optimizations in single and multiobjective approaches. Convergence and performance [8-11] are important algorithms' parameters to be compared.

MSAs came from the idea of using printed circuit technology not just to the circuit components and transmission lines, but also to the radiant elements in an electronic circuit.

¹ Departamento de Microondas e Óptica. Universidade Estadual de Campinas. Av. Albert Einstein - 400, Campinas-SP, Brasil.
E-mail: jurbrian@dmo.fee.unicamp.br; henrique@dmo.fee.unicamp.br; hugo@dmo.fee.unicamp.br

Besides the compatibility to integrated circuit technology, microstrip antennas offer other advantages, such as small dimensions, low weight and cost, and easy conformation to several kinds of surfaces [12]. This type of antenna has many parameters to be optimized, what could characterize multiobjective optimization problem.

In order to perform these improvements, three of the main Evolutionary Algorithms (EAs) were chosen to be analysed in this paper: Genetic Algorithms (GA), Evolutionary Programming (EP) and Evolutionary Strategies (ES). They were implemented to solve the MSA design problem with single and multiobjective optimization approaches. First a fitness function, with parameters to be optimized, was maximized. Second, the Pareto frontier of the problem was plotted, providing a set of optimum solutions.

These EAs were applied to design a rectangular MSA to a specific frequency (2.4 GHz). The objective function was obtained using closed form expression derived from the Cavity Method to calculate the several antenna parameters. In this method, the MSA is considered a cavity with losses.

This work first presents a brief introduction to MSAs, indicating the method used to evaluate the heuristic search. Then, it shows the main necessary concepts for the developed EAs. Next, the computational optimization comparative results and their methodology are presented, considering different distributions for random numerical generation, evolutionary operators, selection criteria, EAs, and both single and multiobjective approaches. Finally conclusions are drawn.

MICROSTRIP ANTENNAS

MSAs consist of a metallic thin patch placed a small fraction of the wavelength above the ground plane. The broadside radiation pattern is obtained with an appropriated excitation from below the patch [12]. For a rectangular MSA, the length L of the patch varies usually between $\lambda_0/3$ and $\lambda_0/2$ (where λ_0 is the wavelength in free space).

The metallic patch and the ground plane are separated by a dielectric layer, the substrate. There are many kinds of substrates to be used in MSAs, with different dielectric constant. The patch may have several forms, the most common being the rectangular and circular, for

their easy analysis and manufacture, and their attractive radiation features.

There are many possible configurations to feed a MSA. The four more popular are by microstrip line, by coaxial probe, by aperture coupling, and by proximity coupling [12]. This work considers only MSAs with coaxial probe feeding. In this case, the internal conductor is connected to the patch, while the external is connected to the ground plane. With this kind of feeding, the desired input impedance may be obtained by appropriately placing the internal conductor.

Optimized parameters from a rectangular MSA, except dielectric constant of substrate, are shown in Figure 1, where x_{in} is the distance from the feed point to the nearest resonant side, L is the non-resonant side length, W is the resonant side length, and h is the substrate height.

In simulations it was imposed that: 1) The value of h is limited by the interval of 0.1mm to 10mm; 2) The relative dielectric constant (ϵ_r) of substrate is limited to the range 1 to 10; 3) To obtain an MSA for 2.4 GHz, W and L are limited between 7 and 4 cm; 4) As input impedance is symmetrical around the centre of the patch, x_{in} can be no greater than $L/2$; it should therefore be smaller than 3.5 cm, and any solution with $x_{in} > L$ must be eliminated. A CAD procedure, based on equations derived from the Cavity Method, is used to calculate the antenna's several parameters [4].

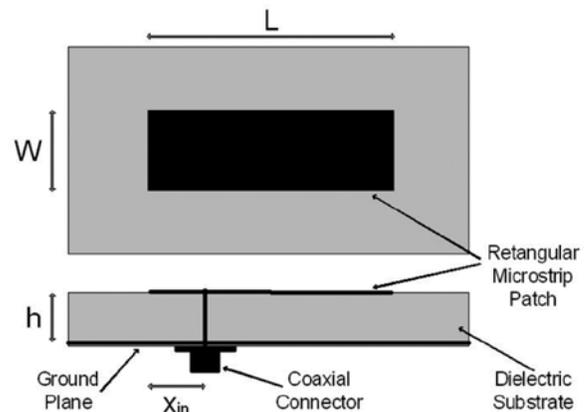


Figure 1. Rectangular microstrip antenna and parameters optimized by EAs. Top view (above), and side view.

The validity of the parameters' expressions was previously guaranteed to every considered situation.

One of the MSA configurations obtained with the EAs was also simulated in a commercial electromagnetic simulation software [13], as verification. The results for some antenna parameters are compared in Table 1.

Table 1. Comparison between some MSA parameters calculated by the CAD Model based on the Cavity Method, and obtained by simulation in a commercial EM software f_0 is resonant frequency, D is directivity.

<i>MSA Parameters</i>	<i>CAD Model</i>	<i>Commercial EM Simulation Software</i>
F_0 (GHz)	2.40563	2.326
BW (%) ($S_{11} \leq -6$ dB)	6.30184	2.49
er (%)	96.6614	100
D (dB)	6.07199	8.059
SWR (minimum)	1.72268	1.544

One can notice in Table 1 that some parameters received different values. The greater differences occur for BW and D. It however does not devalue the evolutionary optimization and the accomplished comparison among methods.

EVOLUTIONARY ALGORITHMS: CONCEPTS

As biologically inspired computing is basically aimed at solving problems, most of the approaches are not concerned with the creation of accurate models [14]. They are search and optimization procedures that have their origin in biological world.

Every search heuristics applied to solve a particular problem requires two major steps, which are representation and evaluation function (fitness) [14]. In many engineering and science areas bio-inspired algorithms have been welcomed and became a popular tool [15].

Genetic Algorithm: Concepts

GA is a method based on biological evolution. Basically, GAs have at least the following elements in common: population of chromosomes, selection method according to fitness, crossover and random mutation operators [4]. Parameters responsible for the features in the individuals are mapped onto the chromosomes, with binary or real numbers.

The selection operation is based on objective (fitness) function and is applied to select chromosomes (individuals)

in the population for reproduction. Crossover operator chooses one or more loci in the chromosomes to make a recombination of genes from there, and to create one or more children individuals as offspring. Next, mutation changes some of the genes in chromosomes. Crossover and mutation are important operators responsible for improving diversity at fitness surface.

Roulette Wheel, Tournament and Ranking are the most common selection criteria used on GAs. One of them is chosen according to fitness function and individual representations. The effect of mutation and crossover operators depends on this choice as well.

Evolutionary Programming: Concepts

EP was introduced in the 60s as an evolutionary technique to develop an alternative form of artificial intelligence, evolving finite state machines. It was later extended in the 90s to operate with real-valued vectors subject to Gaussian mutation, similar to ES [16-17]. More recently it has been applied to electromagnetic problems [9].

Selection method is always tournament, a rather different than what is adopted by GA. After generating an offspring, which then suffers the action of the genetic operator, parents and children are grouped and compete together in a tournament. Next, the individuals with the highest scores are kept in an usually fixed-size population, and are then transmitted to the next generation. It is thus equivalent to the selection ($\mu + \lambda$) in ES.

EP employs only mutation as its genetic operator. Reproduction process is therefore asexual, namely each single parent produces one single offspring, and the algorithm consequently operates in species level. Mutation is originally Gaussian, with standard deviation been also evolved. This turns the algorithm self-adaptive, an important feature for the evolution process.

Evolutionary Strategies: Concepts

ES appeared in the 60s and is contemporaneous to GA and EP. It is focused on continuous and parametric problems. Furthermore, these algorithms are commonly applied to problems without differentiability in the optimization criteria, and to multimodality, which are problems complex enough to not be solved by conventional methods.

Performance of this method depends largely on adjustment of internal parameters, and in almost every case mutation is the main operator [18].

Furthermore, many representations, selection criteria and operators can be used [18], but a usual ES goal is to optimize some given function $F(s)$, according to many control parameters or decision variables $y=(y_1, y_2, \dots)$.

ES operates with a population β of individuals α . Each individual α_k is composed of a parameter set or vector data structure y_k , an objective function and, sometimes, a usually called endogenous strategy parameters, s_k , also referred to as fitness:

$$\alpha_k = (y_k, s_k, F(y_k)) \quad (1)$$

Endogenous strategy parameters are a peculiarity of ES, and they are used to control certain statistical properties of operators.

In addition, ES employs exogenous parameters at evolutionary operators and selection control, which are applied to identify the number of parents (μ), the number of parents recombined to create an offspring (ρ) and the number of offspring individuals (λ).

Basically, there are two types of implementation in a standard ES [16]:

- $\left(\frac{\mu}{\rho} + \lambda\right)_{EE}$: μ parents generate λ children as offspring and the whole population ($\mu + \lambda$) is then reduced to μ individuals. Selection operates on the set formed by parents and offspring. However, parents survive until offspring becomes better adapted to the environment.
- $\left(\frac{\mu}{\rho}, \lambda\right)_{EE}$: μ parents generate λ children as offspring. In this case selection only operates on offspring set. For this reason, it is necessary that $\lambda > \mu$.

Both $\left(\frac{\mu}{\rho} + \lambda\right)_{EE}$ and $\left(\frac{\mu}{\rho}, \lambda\right)_{EE}$ have the same structure.

Multiobjective Optimization

To effectively deal with multiobjective optimization problems, an appropriate approach should be taken. The best one is to find the so called Pareto optimum set or Pareto frontier of the problem. If a solution is in this group, there is no other feasible solution which has some better criterion without having a simultaneous worsening one in at least one of others criteria [19, 20]. These are the so

called dominant solutions [21]. Not all Pareto optimum solutions may constitute an acceptable one, but a Pareto optimum solution should always be a better compromise solution than any other it dominates [22].

MSAs OPTIMIZATION: PROCEDURES AND RESULTS

All bio-inspired algorithms use random numbers in their working mechanism, mainly at the genetic operators. And random numbers with different probability density functions (PDFs) result in different effects on algorithms, turning them more appropriate to solve different problems.

In implemented EAs from this work, effect of random numbers with four different PDFs was assessed. PDFs were approximately Uniform, Gaussian, Cauchy or a Hybrid Gaussian-Cauchy, as it is seen in Figure 2.

In the hybrid case, random number is a linear combination of a Gaussian and a Cauchy random numbers, given by:

$$H(\mu, \sigma^2) = N(\mu, \sigma^2) + \beta \cdot C(\mu, \sigma^2) \quad (2)$$

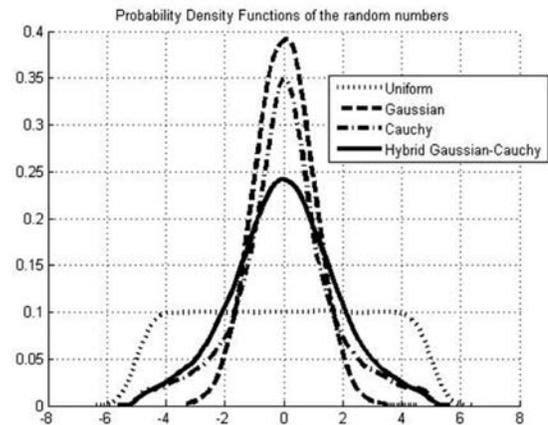


Figure 2. Probability density functions of random numbers used in EAs, with $\mu = 0$, $\sigma^2 = 1$ and limited between -5 and 5.

where $N(\mu, \sigma^2)$ is a Gaussian random number with mean μ and variance σ^2 , $C(\mu, \sigma^2)$ is a Cauchy random number with median μ and scale parameter σ^2 , and β is the ratio between both random numbers in the combination.

In EAs' implementation, each individual was represented by a vector with the value of variables to be optimized:

$I = [x_m, L, W, h, \epsilon_r]$. In all accomplished tests the total number of generations was 1000 and 2.4 GHz was the desired frequency, with 5% acceptable limit error.

Chromosomes in the population were always initialized with uniform random values, between limits -5 and 5. During the following generations these values were modified by genetic operators. Before each fitness evaluation, these chromosome values were converted to a value between limits of each MSA variable under optimization and proportional to its previous position in the range [-5, 5].

The following optimization targets were adopted: maximization of standing wave ratio (SWR), radiation efficiency (ϵ_r) and bandwidth (BW), besides tuning antenna to 50 Ω input impedance and to the desired operational frequency. This is, therefore, a multiobjective optimization.

Always when a variable X reached a value beyond its maximum or minimum allowed limits, through genetic operators, a repair procedure was adopted. Variables then received a value $rep_1 = X_{max} - N$ or $rep_2 = X_{min} + N$, respectively if the maximum or minimum limit was exceeded, where N is a uniform random number between variable's limits.

Each EA has other specific features as well, which will be detailed next.

Genetic Algorithm: Implementation

In GA case, crossover and mutation taxes were fixed at 70% and 10% respectively. Chromosomes were real-valued, population was composed of 80 individuals and selection method used was tournament, with 10 opponents. Each generation, for 80 times, 10 individuals were chosen randomly (uniform distribution) and their fitness were compared. Then the individual with the highest fitness was chosen to join the group which would suffer the action of genetic operators and generate an offspring.

The implemented algorithm was elitist, which is a memory resource where, if the best individual of the next generation is worse than the best one of the current generation, the worst individual of the next generation is substituted by the best individual of the current generation. Crossover was arithmetic and occurred in only one point of chromosome. Each parent's gene from a generation (x_i) generated a child's gene (x_{i+1}) by Equation (3), with equal probability using addition or subtraction.

$$x_{i+1} = x_i \pm N \quad (3)$$

In Equation (3) N represents a random number of any of the four cases considered, with PDF centred in zero and null outside the interval [-5, 5], and also with variance or scale parameter equal to one, if the random number is not uniform.

Evolutionary Programming: Implementation

As the GA case, population was composed of 80 individuals and tournament was the adopted selection method, with 10 opponents. Now each individual fitness was compared to fitness from randomly chosen opponents (uniform distribution) in population, generating a score to each individual (number of opponents which had fitness value smaller). The 80 individuals with the highest scores were then chosen to suffer the action of genetic operator and to generate next generation.

As the algorithm was self-adaptive, for each gene in chromosome (x), there was another one to be used in self-adaptation (η). By mutation, the only operator present, each child's gene (x_{i+1} or η_{i+1}) was generated from the equivalent parent's one (x_i or η_i) through Equations (4, 5).

$$x_{i+1} = x_i + \eta_i \cdot N_1(0,1) \quad (4)$$

$$\eta_{i+1} = \eta_i \cdot e^{[\tau' \cdot N_2(0,1) + \tau \cdot N_1(0,1)]} \quad (5)$$

$N_1(0, 1)$ represents a Gaussian random number with mean equal to zero and variance equal to one, and was generated to each gene in chromosome. $N_2(0, 1)$ also represents a Gaussian random number with mean equal to zero and variance equal to one, but now it was generated once each generation. Scale factors τ and τ' are defined respectively by Equations (8, 9), where n is the dimension of the search space [9], that is, the number of optimized variables in EA. In hybrid random numbers case, β was also evolved [9].

$$\tau = \left(\sqrt{2\sqrt{n}} \right)^{-1} \quad (6)$$

$$\tau' = \left(\sqrt{2n} \right)^{-1} \quad (7)$$

Evolutionary Strategies: Implementation

ES follows other EAs presented before. This optimization was done with 50 parents (μ) generating 30 offsprings (λ).

Both standard ESs, $(\mu + \lambda)$ ES and (μ, λ) ES, were used. In order to reduce the mutation effect and to improve diversity on fitness surface, a recombination operator (crossover) was introduced. It is different from GAs', and recombines 3 random parents to generate an offspring.

Each individual was represented by the vector $\alpha_k = (y_k, \theta, \sigma)$, where y_k is attribute input vector, θ and σ are recombination and mutation taxes, respectively [16]. This can be considered an endogenous data structure representation. $\theta = 70\%$ was used, the same as in GA. Mutation tax used was higher ($\sigma = 25\%$), because it is the main genetic operator in ES.

The standard ES mutation [1] was applied, where the selected attribute to be mutated was increased, or decreased, by an independent random sample with a chosen distribution function.

Single Objective Approach

To turn the MSA optimization problem into a single objective problem (the simplest approach to multiobjective problems), a fitness function F (Equation (8)) was empirically adopted.

$$F = \frac{e_r + (1 / \text{SWR}) + 20 \times \text{BW} + F_1}{4 \cdot F_2} \quad (8)$$

In (9)

$$F_1 = 1 - \frac{|50 - R_{in}|}{50} \quad (9)$$

if $|50 - R_{in}| \leq 50$ or $F_1 = 0$ otherwise, and $F_2 = 0$ if $f_0 < f - (f \cdot df) / 2$ or $f_0 > f + (f \cdot df) / 2$, or $F_1 = 1$ otherwise.

In these expressions e_r is the radiation efficiency, BW is bandwidth, R_{in} is the input resistance, f is desired operational frequency, f_0 is calculated operational frequency.

Each of added elements in Equation (8) receives an approximate equal weight. Employment of F_2 imposes a penalty to the individual whose f_0 is out of the supplied acceptable percent error, and F_1 reinforces the objective of minimizing SWR.

Figure 3 presents the mean evolution of the maximum fitness values after 30 trials obtained with the four different EAs (GA, EP, $(\mu / \rho + \lambda)$ ES and $(\mu / \rho, \lambda)$ ES), for each of the random numbers distributions considered. It shows the reachability of each method.

In all cases, GA had the faster convergence and the highest average maximum fitness values. Its performance seemed to suffer not much influence from random number distributions. This EA usually explores efficiently wider areas of the search space in the first stages of the optimization [23], what explains its good convergence. Its elitist implementation, however, might attract evolution to a local optimum in more multimodal problems.

On the other hand, EP presented slower convergence in every situation. As it is reported in literature [9], this EA is more efficient in the middle and later stages, but usually it keeps a constant progress towards a global solution.

Except for $(\mu + \lambda)$ ES, random numbers with uniform distribution were harmful to EAs' performance in this MSA optimization problem. EP and GA had their worse efficiency in this case.

(μ, λ) ES showed a high level of noise at fitness surface when its evolutionary operators used uniform or hybrid random numbers. It can be explained by the absence of memory in (μ, λ) selection criterion, and by the wider PDF of random numbers. These factors did not guarantee a good convergence.

EP and (μ, λ) ES had the most considerably changed convergence.

Therefore, Figure 3 indicates the importance of random number distribution used by evolutionary operators, in application of EAs to a specific optimization problem.

Multiobjective Approach

The single objective approach does not appropriately deal with the multiobjective problem of optimizing e_r , SWR and BW in a MSA. Nothing guarantees that relation

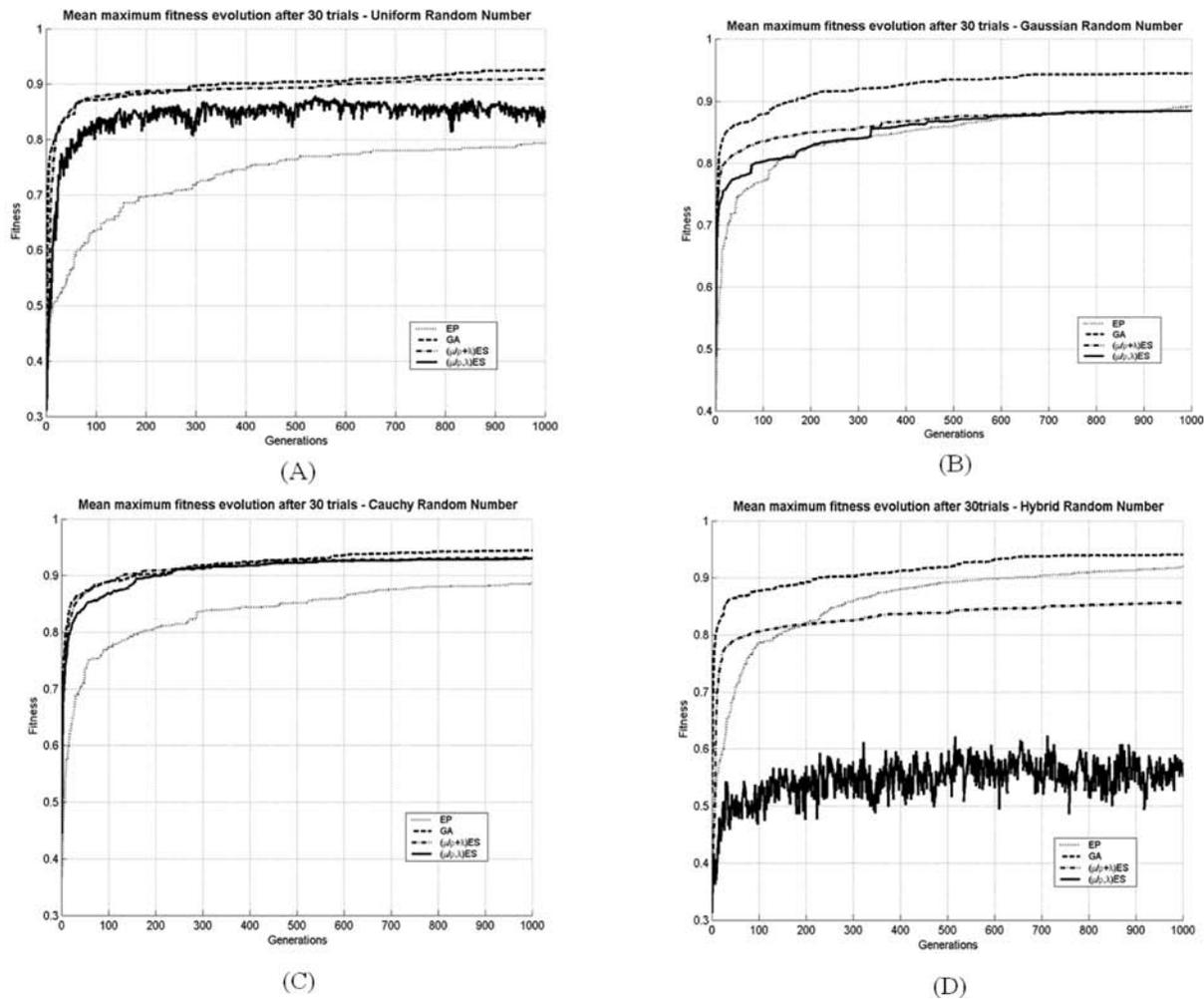


Figure 3. Average evolution of maximum fitness values after 30 trials, obtained with four different EAs (GA, EP, $(\mu/\rho + \lambda)$ ES, and $(\mu/\rho, \lambda)$ ES), for each random number distribution considered. A) Uniform, B) Gaussian, C) Cauchy, D) Hybrid.

among these parameters is correctly expressed in the fitness function adopted.

That is why an approach similar to the Strength Pareto Evolutionary Algorithm [24] was implemented next. At each generation, the number of individuals dominated by each one in population was counted, and stored as the “strength” of each individual. Then, the ones with the highest strength at each generation were checked to see if they could be included in an external pool of individuals, aimed to keep dominants found during the whole optimization. If there was not yet any individual there with the same values for er , SWR and BW , and if there was not any individual there which dominated the one being analyzed, it was included in pool. A limit number of 500 individuals were allowed in this external population. With this procedure, after some generations

individuals in pool should lie over the true Pareto frontier of the problem. Then the user can choose an individual among the optimum.

This multiobjective approach was implemented in the four EAs built previously. Before using them in MSA optimization problem, they were applied to a known 2D problem for assessment: Minimize Equations (23-18), subject to $0 \leq x_i \leq 6$ [23].

$$f_1(x) = \sum_{i=1}^5 \sin(\pi x_i) \tag{10}$$

$$f_2(x) = \sum_{i=1}^5 \cos(\pi x_i) \tag{11}$$

Results obtained are shown in Figure 4. In comparison with results presented in [9], one can notice GA and EP practically reach the true Pareto frontier, while both ES configurations do not. This shows that, although ES results get near to the position and shape expected for the true Pareto frontier, it is still necessary to improve the ES implementation to be used in a really multiobjective optimization approach.

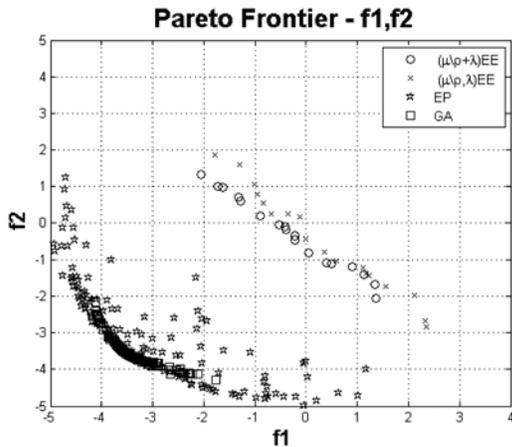


Figure 4. Pareto Frontier obtained for the 2D test problem.

These EAs were next applied to MSA optimization. Pareto frontiers obtained in each EA, with different random number distributions, were generally similar, presenting approximately the same shape and position. With EP however, the width of the Pareto frontier found was broader, what might have been result of a better coverage.

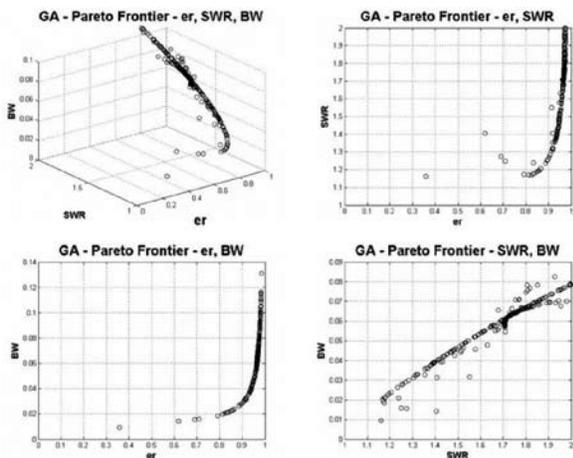


Figure 5. Pareto Frontier with Gaussian random numbers for MSA optimization with Genetic Algorithm.

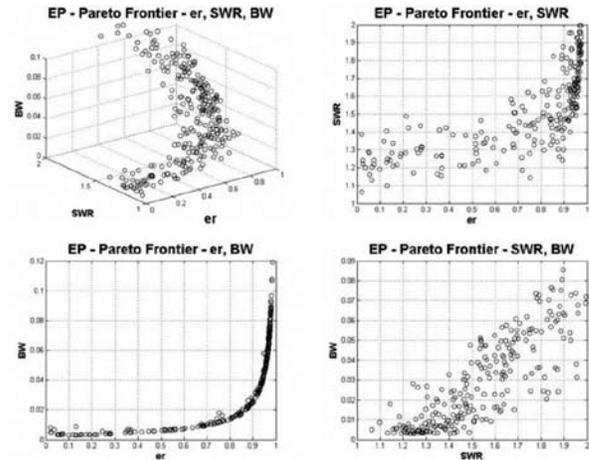


Figure 6. Pareto Frontier with Gaussian random numbers for MSA optimization with Evolutionary Programming.

Figures 5 to 8 show Pareto frontiers obtained with each analysed EA, in case of Gaussian genetic operators. Both ESs implementations picked fewer individuals for the Pareto frontier.

Although the evaluation test presented earlier (Figure 4) suggested ES was inefficient to find the Pareto frontier, in this multiobjective approach for MSA optimization both of them presented better results, finding individuals concentrated in regions of the frontiers found by GA and EP.

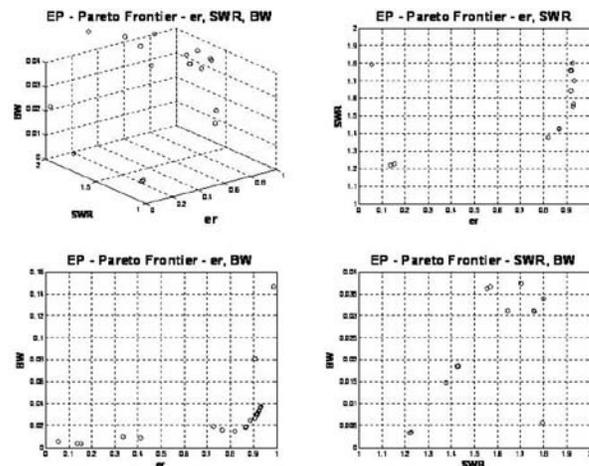


Figure 7. Pareto Frontier with Gaussian random numbers for the MSA optimization with, $(\mu + \lambda)$ ES.

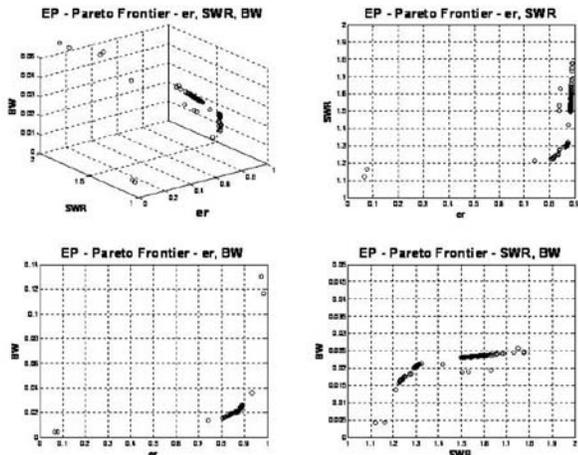


Figure 8. Pareto Frontier with Gaussian random numbers for the MSA optimization with, (μ, λ) ES.

With the Pareto frontier revealed, one can choose one or more solutions which offer an acceptable compromise among SWR, BW and er.

Finally, for a more accurate and objective comparison among EAs' multiobjective performances, all individuals in all Pareto frontiers were grouped and strength of each one relative to the others was computed. Figure 9 presents the average strength of individuals from each combination (EA and random number distribution), normalized by the sum of strengths from all individuals. The highest the bar, more dominant individuals were found by the combination, relative to others.

Mean Strength of the individuals in the Pareto Frontier

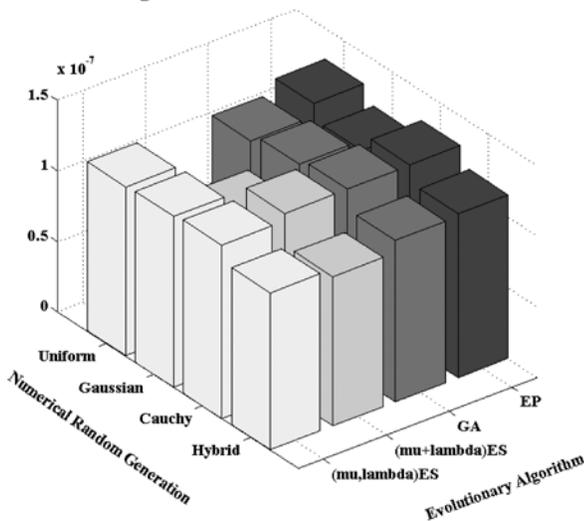


Figure 9. Mean strength of each combination (EA, random number), relative to the sum of strengths from all individuals in Pareto Frontier.

EP found the most dominant solutions when it used uniform or Cauchy random numbers.

Furthermore, $(\mu + \lambda)$ ES showed the most stable average strength, for every random number distributions. On the other hand, (μ, λ) ES was the most inefficient EA in this analysis. When it used uniform or Gaussian random numbers, the Pareto frontier it found presented more dominated individuals, related to others. However, with Cauchy and Hybrid random numbers, its efficiency on finding the true Pareto frontier was similar to other EAs'.

CONCLUSIONS

One can conclude from obtained results that the employment of random numbers with different distributions in operators used by EAs considerably influences their convergence and efficiency on finding global optimum solutions.

For MSA optimization design, with parameters calculated through a CAD model based on Cavity Method, GA was always the best EA in convergence and in efficiency to find global solutions. GA was also relatively immune to effects derived from evolutionary operators with different random number distributions. In a truly multiobjective approach, however, where the Pareto frontier for the problem is found, EP seems to supply a broader coverage of it. This is a better result because it provides more options of solutions which offer acceptable compromises among optimized parameters.

Integration of other powerful numerical method for electromagnetic simulation with EAs can provide a better accuracy and efficient convergence results. Furthermore, some other features implemented for multiobjective optimization, such as *crowding* [5], could result in more complete and covered Pareto frontiers.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the assistance and support of FAPESP (06/57074-2), CAPES and Laboratório de Eletromagnetismo Computacional e Aplicado (LEMAC) - Universidade Estadual de Campinas (UNICAMP), Campinas - SP, Brasil.

REFERENCES

- [1] L.N. de Castro and F.J. Von Zuben. "Recent Developments in Biologically Inspired Computing". Idea Group Inc. USA. 2005.
- [2] S.D. Targonski, R.B. Waterhouse and D.M. Pozar. "Design of wide-band aperture-stacked patch microstrip antennas". *IEEE Trans. Antennas Propagation*. Vol. 46 N° 9, pp. 1245-1251. 1998.
- [3] K. Ghorbani and R.B. Waterhouse. "Ultrabroadband printed (UBP) antenna". *IEEE Trans. Antennas Propagation*. Vol. 50 N° 12, pp. 1697-1705. 2002.
- [4] K.F. Lee and W. Chen. "Advances in Microstrip and Printed Antennas". J. Wiley & Sons. New York. 1997.
- [5] R. Garg, P. Bhartia, P. Bahl and A. Ittipiboon. "Microstrip Antenna Design Handbook". Artech House. Boston. 2001.
- [6] J. Smajic, C. Hafner and D. Erni. "Optimization of photonic crystal structures". *Journal of the Optical. Society of America A*. Vol. 21 N° 11, pp. 2223-2232. 2004.
- [7] S.F. Preble, H. Lipson and M. Lipson. "Novel two-dimensional photonic crystals designed by evolutionary algorithms". *Nanophotonics for Communication: Materials and Devices. Proceedings of SPIE*. Vol. 5597, pp. 118-128. 2004.
- [8] J.R. Brianeze, C.H.S. Silva-Santos and H.E. Hernández-Figueroa. "Evolutionary Algorithms Applied to Microstrip Antennas Design". *The Second European Conference on Antennas and Propagation (EuCAP 2007)*. Edinburgh, UK. 2007.
- [9] A. Hoorfar. "Evolutionary programming in electromagnetic optimization: a review". *IEEE Trans. Antennas Propagation*. Vol. 55 N° 3, pp. 523-537. 2007.
- [10] F.J. Ares-Pena, J.A. Rodriguez-Gonzalez, E. Villanueva-Lopez and S.R. Rengarajan. "Genetic algorithms in the design and optimization of antenna array patterns". *IEEE Trans. Antennas Propagation*. Vol. 47 N° 3, pp. 506-510. 1999.
- [11] R.L. Haupt. "Antenna design with a mixed integer genetic algorithm". *IEEE Trans. Antennas Propagation*. Vol. 55 N° 3, pp. 577-582. 2007.
- [12] C.A. Balanis. "Antenna Theory: Analysis and Design". John Wiley & Sons, Inc. 2005.
- [13] CST Microwave Studio 2006B. Date of visit: July, 2007. URL: www.cst.com
- [14] A.E. Eiben, R. Hinterding and Z. Michalewicz. "Parameter Control in Evolutionary Algorithms". *IEEE Transactions on Evolutionary Computation*. Vol. 3 N° 2, pp. 124-141. 1999.
- [15] M. Tomassini. "Evolutionary Algorithms", in E. Sanchez, M. Tomassini (Eds), *Towards Evolvable Hardware: The evolutionary engineering approach*, LNCS 1062, pp. 19-47. Springer-Verlag. 1996.
- [16] L.N. de Castro. "Fundamentals of Natural Computing Basic Concepts, Algorithms, and Applications". Chapman & Hall/CRC Taylor and Francis Group, USA. 2006.
- [17] T. Bäck, D.B. Fogel and Z. Michalewicz. "Evolutionary Computation 1: Basic Algorithms and Operators". Institute of Physics Publishing, USA. 2000.
- [18] H.G. Beyer and H.P. Schwefel. "Evolution Strategies a comprehensive introduction". *Natural Computing 1*, Kluwer Academic Publishers, pp. 3-52. 2002.
- [19] E. Zitzler, L. Thiele and K. Deb. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", *Evolutionary Computation*. Vol. 8, issue 2, pp. 173-195. 2000.
- [20] J. Teo and H.A. Abbas. "Multiobjectivity and Complexity in Embodied Cognition". *IEEE Transactions on Evolutionary Computation*. Vol. 9 N° 4, pp. 337-360. 2005.
- [21] C.A.C. Coello. "An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of Art and Future Trends". *Proceedings of the Congress on Evolutionary Computation IEEE Press*. Vol. 1, pp. 3-13. 1999.
- [22] T. Bäck, D.B. Fogel and Z. Michalewicz. "Evolutionary Computation 2: Advanced Algorithms and Operators". Institute of Physics Publishing, USA. 2000.

- [23] G.P. Coelho and F.J. Von Zuben. "Omni-aiNet: An Immune-Inspired Approach for Omni Optimization". ICARIS-2006, pp. 294-308. 2006.
- [24] C.A.C. Coello. "Evolutionary Multi-objective Optimization: A Historical View of the Field". IEEE Comp. Intelligence Mag. Vol. 1 N° 1, pp. 28-36. 2006.