A Multi-Objective Evolutionary Algorithm Based on Decomposition for Optimal Design of Yagi-Uda Antennas

R. Carvalho[1], R. R. Saldanha[1], B. N. Gomes[1], A. C. Lisboa[1,2], and A. X. Martins[3]

[1]Department of Electrical Engineering, Federal University of Minas Gerais, Brazil
[2]ENACOM–Handcrafted Technologies, Brazil
[3]Department of Applied and Exact Science, Federal University of Ouro Preto, Brazil

This paper presents a multi-objective evolutionary algorithm based on decomposition (MOEA/D) to design broadband optimal Yagi-Uda antennas. A multi-objective problem is formulated to achieve maximum directivity, minimum voltage standing wave ratio and maximum front-to-back ratio. The algorithm was applied to the design of optimal 3 to 10 elements Yagi-Uda antennas, whose optimal Pareto fronts are provided in a single picture. The multi-objective problem is decomposed by Chebyshev decomposition, and it is solved by differential evolution (DE) and Gaussian mutation operators in order to provide a better approximation of the Pareto front. The results show that the implemented MOEA/D is efficient for designing Yagi-Uda antennas.

***Index Terms*—MOEA/D, multi-objective optimization, decomposition problem, design Yagi-Uda antenna.**

I. Introduction

DESIGNING broadband Yagi-Uda antennas is a hard task due to the required trade-off of conflicting specifications, like directivity (D), standing wave ratio (SWR), and front-to-back ratio (FBR), which should be handled as better as possible through the entire design spectrum.

In order to design an optimal Yagi-Uda antenna, each aforementioned specification was considered at different frequencies. Thus, we have a problem with many conflicting objectives to be solved. This complex problem can be tackled using stochastic techniques [1]–[3], deterministic techniques [4], among others.

This paper presents an extension of multi-objective evolutionary algorithm based on decomposition (MOEA/D) proposed by Chen *et al.* [5]. The main contributions are the different treatment of algorithm operators, the use of external population and an update solution rule, aiming to make the algorithm more efficient, i.e., aiming to obtain a better approximation of the optimal Pareto set with fewer problem analysis. The proposed algorithm uses a decomposition method to convert a multi-objective problem into a set of mono-objective problems and it tries to approximate the optimal Pareto front by solving all these subproblems together. The implemented MOEA/D is applied to the design of 3 to 10 elements optimal Yagi-Uda antennas.

II. The Optimization Problem

The Yagi-Uda antenna is an array of k linear elements, composed by a driven element (dipole) with a reflector parasitic element in the rear and director parasitic elements in the front, as shown in Fig. 1. The element centered at the origin is the reflector, followed by the centered-fed driven element and directors. The distance between consecutive elements and the length of each element are the parameters to be optimized. These parameters are respectively represented by $d_i = x_{i+1} - x_i$, $i = 1, 2, \ldots, k-1$, and ℓ_i, $i = 1, 2, \ldots, k$.

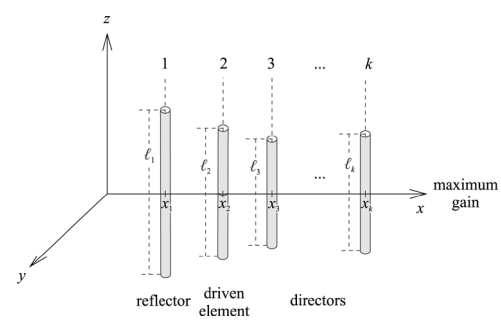

Fig. 1. Geometry of a k-elements Yagi-Uda antenna.

This paper investigates the design of broadband Yagi-Uda antennas considering three conflicting objectives: directivity (D), standing wave ratio (SWR) and front-to-back ratio (FBR). These should be handled as better as possible through the entire design spectrum to guarantee the desired broadband characteristic. The antenna must be able to operate over a band and not only on a single frequency. In order to take this into account, the objective functions were defined as a function of the lower, central and upper frequencies f of the desired operation band, which results in an optimization problem with 9 conflicting objective functions. The problem can be mathematically formulated as

$$\min_s \max_i y(s, f_i) = \begin{bmatrix} -D(s, f_i) \\ -\mathrm{FBR}(s, f_i) \\ \mathrm{SWR}(s, f_i) \end{bmatrix} \tag{1}$$

where $s = (d, \ell)$ are the decision variables composed by the elements length $\ell \in \mathbb{R}^k$ and the distance $d \in \mathbb{R}^{k-1}$ between adjacent elements. An antenna with k elements defines a problem with $n = 2k-1$ variables. The goal is to optimize the worst case performance over the entire bandwidth. When antennas are optimized to operate over a frequency range, they are, somehow, robust to perturbations. The antenna is numerically analyzed using the method of moments (MoM) [6] with a delta gap feed model.

Manuscript received July 07, 2011; revised October 08, 2011; accepted October 22, 2011. Date of current version January 25, 2012. Corresponding author: R. Carvalho (e-mail: rodrigo@rclink.com.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMAG.2011.2174348

III. MOEA/D

A. Chebyshev Decomposition

The fundamental idea of the proposed MOEA/D is to decompose a multi-objective optimization problem into a set of mono-objective problems (subproblems), using evolutionary computation techniques in the process of searching for optimal solutions.

To decompose the multi-objective problem, it is used the Chebyshev decomposition represented by

$$g(s, \lambda, z^*) = \max_i \lambda_i |y_i(s) - z_i^*| \tag{2}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)$ is a weight vector associated to the objective functions, m is the number of objective functions, and z^* is the set of reference points given by

$$z_i^* = \min_{s \in \Omega} y_i(s), \quad i = 1, \ldots, m. \tag{3}$$

The weight vectors belong to the set

$$\mathbb{T} = \left\{ \lambda \in \mathbb{D} = \left\{ \frac{0}{A}, \frac{1}{A}, \ldots, \frac{A}{A} \right\}^m \middle| \sum_{i=1}^m \lambda_i = 1 \right\} \tag{4}$$

where $A \in \{1, 2, \ldots\}$ is a parameter that defines the number of subproblems and also determines the possible values for the weight vectors in $\mathbb{T}$. The number of generated subproblems is given by

$$|\mathbb{T}| = \binom{A + m - 1}{A} = \frac{(A + m - 1)!}{A!(m-1)!}. \tag{5}$$

For instance, consider $A = 4$ and $m = 2$. Then $\mathbb{D} = \{0, 0.25, 0.50, 0.75, 1\}^2$, $\mathbb{T} = \{(0,1), (0.25, 0.75), (0.5, 0.5), (0.75, 0.25), (1, 0)\}$ and $|\mathbb{T}| = 5$.

MOEA/D has a population of subproblems, where each individual represents a different problem, defined by its unique weight vector in relation to other individuals. These weight vectors are assigned in a uniform way to subproblems under the condition of belonging to $\mathbb{T}$.

Under mild conditions, a reasonable large number of evenly distributed weight vectors usually lead to a set of Pareto optimal vectors, which may not be evenly spread but could approximate the Pareto optimal front very well [7].

The weight vectors are also used to define a neighborhood structure based on Euclidean distance. This structure is used in the optimization process because the subproblems consider neighbors having similar fitness (landscape) and their respective utopian solutions are close in the decision space.

Through the optimization process, the neighbors are used in two distinct forms. In the first case, they are used in the selection of individuals to apply the evolutionary operators, because the crossover probability is greater between neighboring individuals. The second case is to try to update a predetermined number of neighboring subproblems ($N_{\max}$) during operators application. The subproblem is updated based on the condition $g(s', \lambda_i, z) \geq g(s, \lambda_i, z)$, i.e., if the variables of a subproblem that is being optimized returns a better result to a neighboring subproblem, then the variables of the neighboring subproblem are updated to s'.

MOEA/D optimizes the set of subproblems simultaneously. Thus, each optimal solution for these subproblems is a Pareto solution of the multi-objective problem.

B. Diversified Initial Population and External Population

Aiming to obtain a diversified initial population and a better domain mapping, the search space Ω is divided into segments for each variable in $s_{i,j}$, $i = 1, \ldots, n$, $j = 1, \ldots, |\mathbb{T}|$, based on the Latin hypercube [8]. Consider, for example, a search space $\Omega_i = [-1, 1]$ divided into 4 segments. Therefore, the i-th variable for the j-th subproblem will be generated inside of one of the subintervals $[-1.0, -0.5)$, $[-0.5, 0.0)$, $[0.0, 0.5)$ and [0.5, 1.0], preventing the i-th variable from assuming values in only one interval for all subproblems.

An external population ($\mathbb{T}_E$) ensures that the nondominated points are preserved over the optimization process. For increasing the performance based on algorithm time, the actual population is not inserted in the external population in all iterations. Furthermore, at each insertion of the current population into external population, the non-domination criterion is applied.

C. Neighborhood Relationship

Another important characteristic of MOEA/D is the update mechanism. When this mechanism is totally random, a similar situation that happens in the process scheduling of operational systems, known as starvation, can occur. In the MOEA/D case, this situation may trap the subproblems away from other points in the decision space, slowing down the convergence.

To prevent starvation, an update rule is used. Consider $\mathbb{V}_j$ a set of neighboring subproblems of the subproblem j, and $p_r = 1/|\mathbb{V}_j|$ the initial probability of choosing a neighbor r in $\mathbb{V}_j$. If an individual r' is randomly chosen in the iteration $i + 1$, $p_{r'}$ is decremented by $p_{r'}/2$, and the probability of the remaining individuals $r \in \mathbb{V}_j \backslash \{r'\}$ is incremented by $p_{r'}/[2(|\mathbb{V}_j| - 1)]$, so that $\sum_r p_r = 1$ at all iterations. This process is carried out at each iteration. To illustrate this process, let $|\mathbb{V}_j| = 4$, p_r be the probability of choosing the neighboring subproblem $r \in \mathbb{V}_j$, and 1, 2 be the chosen neighbor sequence for updating. Thus, initially, $p_1 = p_2 = p_3 = p_4 = 0.25$. In the first iteration, the neighbor 1 is chosen and the probabilities become $p_1 = 0.125$ and $p_2 = p_3 = p_4 \cong 0.2916$. The neighbor 2 is chosen in the next iteration and the probabilities become $p_1 \cong 0.1736$, $p_2 \cong 0.1458$ and $p_3 = p_4 \cong 0.3402$.

D. Differential Evolution

The differential evolution (DE) operator proposed by Storn and Price [9] is used within the MOEA/D proposed in this paper. It is a simple and efficient optimization operator widely used in nonlinear optimization with continuous variables.

The DE initially selects three different individuals in the population, named s_1, s_2 and s_3. It then performs the operation known as differential mutation, where a new individual is created by displacing the base vector s_1 towards the vectorial

difference between s_2 and s_3, i.e.,

$$s' = s_1 + \mu(s_2 - s_3), \tag{6}$$

where μ is a random number with uniform probability inside the interval [0, 1].

The mutant vector s' is then crossed over with the vector s_1 according to the rule

$$s'_i = \begin{cases} s'_i, & r \le p_c \text{ or } i = r' \\ s_{1,i}, & \text{otherwise} \end{cases}, \quad i = 1, \ldots, n \tag{7}$$

where $r \in [0, 1]$ and $r' \in \{1, \ldots, n\}$ are random numbers with uniform probability inside their ranges. The number r' ensures that at least one element of s' will be mutated relative to s_1.

E. Gaussian Mutation

The MOEA/D proposed in this paper uses Gaussian mutation. It consists in assigning, to each variable s_i, $i = \{1, \ldots, n\}$, a random value of a normal distribution $G(s_i, \sigma)$ with mean s_i and standard deviation σ.

F. The Algorithm

Algorithm 1 outlines the MOEA/D proposed in this paper. It is fundamentally composed of 3 phases: initialization, evolutionary process and elitism. In the initialization phase (lines 1 to 4), the subproblems population and their respective neighbor list are created. The lines 6 to 36 of the algorithm summarize the evolutionary process, which involves the application of evolutionary operators to each subproblem and an update of the reference point z^*, subproblems and neighboring solutions. Finally, the elitism phase, represented in the lines 37 to 39 of the algorithm, contains the external population update.

G. Benchmark Tests

To evaluate the performance of the proposed method, testing functions defined in the CEC2009 were used [10]. Since the optimal Pareto fronts of these functions are known, the IGD metric [5] was used to measure the distance between the obtained solutions and their respective optimal Pareto front. For the experiments, 8 testing problems were chosen, of which exactly half were constrained.

Table I shows the average results after 300.000 functions evaluations of 10 independent runs of the MOEA/D (see Algorithm 1) set to $p_c = 0.95$, $p_{cn} = 0.95$, $p_m = 0.30$, $N = 5$, $N_{\max} = 3$ and $Q_{\max} = 650$ iterations. As it can be seen, the proposed algorithm outperforms (worse only for UF2) another version of the MOEA/D, which was introduced by Chen *et al.* [5] and it is one of the best in the competition. We believe that this improvement comes from a better sampling of the search space, especially because of a more diversified initial population and the update rule, which implies in a more unbiased search. Unfortunately, this diversity may slow down convergence for complex Pareto optimal sets, like the one of benchmark UF2. The external population allows a better sampling of the Pareto front, since the non-dominated points are preserved over the optimization process. A common behavior of these algorithms is their deficiency caused by decomposition: when the weight vectors are not uniformly distributed, the points tend to be accumulate into certain regions.

Algorithm 1 Outline of the implemented MOEA/D

Input:

an m-objective problem: $\min_s y(s), s \in \mathbb{R}^n, y : \mathbb{R}^n \mapsto \mathbb{R}^m$

p_c: crossover probability

p_m: mutation probability

p_{cn}: neighbor update probability

N: number of neighboring subproblems

$N_{\max}$: maximum number of neighbors that can be updated

λ: set of weight vectors

$Q_{\max}$: maximum number of iterations

Output:

$\mathbb{T}_E$: external population

1: $\mathbb{T} \leftarrow$ subproblems for each input weight vector λ_j
2: $\mathbb{V}_j \leftarrow$ neighbor list of subproblem $j, |\mathbb{V}_j| = N, \forall j$
3: $\mathbb{T}_E \leftarrow$ nondominated of $\mathbb{T}$ ▷ external population
4: $z_i^* \leftarrow \min_j y_i(s_j), i = 1, \ldots, m$ ▷ reference point
5: **for** $Q = 1 \rightarrow Q_{\max}$ **do**
6: **for** $j \in \mathbb{T}$ **do**
7: choose $\mu \in [0, 1]$
8: choose $j_1, j_2, j_3 \in \mathbb{T}$
9: $s'_j \leftarrow s_{j_1} + \mu(s_{j_2} - s_{j_3})$ ▷ differential mutation
10: choose $r' \in \{1, \ldots, n\}$
11: choose $r_i \in [0, 1], i = 1, \ldots, n$
12: $s'_{j,i} \leftarrow \begin{cases} s'_{j,i}, & r_i \le p_c \text{ or } i = r' \\ s_{j_1,i}, & \text{otherwise} \end{cases}, i = 1, \ldots, n$
13: choose $r_i \in [0, 1], i = 1, \ldots, n$
14: $s'_{j,i} \leftarrow \begin{cases} G(s'_{j,i}, \sigma), & r_i \le p_m \\ s'_{j,i}, & \text{otherwise} \end{cases}, i = 1, \ldots, n$
15: $z_i^* \leftarrow \min\{z_i^*, y_i(s'_j)\}, i = 1, \ldots, m$
16: **if** $g(s'_j, \lambda_j, z^*) \le g(s_j, \lambda_j, z^*)$ **then**
17: $s_j \leftarrow s'_j$ ▷ overwrite subproblem
18: **end if**
19: $c \leftarrow 0$
20: **while** $c < N_{\max}$ **do**
21: choose $r \in [0, 1]$
22: **if** $r \le p_{cn}$ **then**
23: choose $i \in \mathbb{V}_j$ ▷ neighboring subproblem
24: **if** $g(s'_j, \lambda_i, z^*) \le g(s_i, \lambda_i, z^*)$ **then**
25: $s_i \leftarrow s'_j$ ▷ overwrite neighbor
26: $c \leftarrow c + 1$
27: **end if**
28: **else**
29: choose $i \in \mathbb{T}$ ▷ any subproblem
30: **if** $g(s'_j, \lambda_i, z^*) \le g(s_i, \lambda_i, z^*)$ **then**
31: $s_i \leftarrow s'_j$ ▷ overwrite subproblem
32: $c \leftarrow c + 1$
33: **end if**
34: **end if**
35: **end while**
36: **end for**
37: **if** Q is multiple of 10 **then**
38: $\mathbb{T}_E \leftarrow$ nondominated of $\mathbb{T}_E \bigcup \mathbb{T}$
39: **end if**
40: **end for**
41: **return** $\mathbb{T}_E$

IV. Results for Antenna Design

Table II shows the solution to (1) whose SWR is the lowest for the respective k, for a Yagi-Uda antenna with k elements

TABLE I
RESULTS FOR TESTING FUNCTIONS

function	this work	Chen et al. [5]
UF1	0.0016	0.0063
UF2	0.0091	0.0067
UF3	0.0286	0.0561
UF8	0.0790	0.1325
CF1	0.00003	0.0110
CF2	0.0015	0.0126
CF3	0.2271	0.4336
CF8	0.0336	0.4061

TABLE II
WORSE CASE OBJECTIVE FUNCTIONS OF 3 TO 10 ELEMENTS OPTIMAL ANTENNAS AMONG THE 3 SAMPLE FREQUENCIES

Antenna	D (dBi)	VSWR	FBR (dB)
3 elements	7.03	1.48	13.66
4 elements	8.27	1.15	11.84
5 elements	8.61	1.13	13.20
6 elements	9.17	1.10	12.88
7 elements	10.89	1.16	12.36
8 elements	11.05	1.18	11.90
9 elements	10.66	1.08	11.66
10 elements	11.08	1.24	13.56

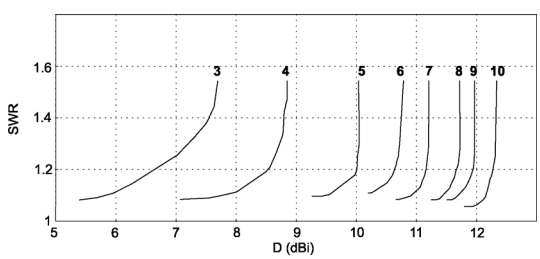

Fig. 2. Approximate Pareto optimal front for directivity and stationary wave ratio for antennas with 3 to 10 elements.

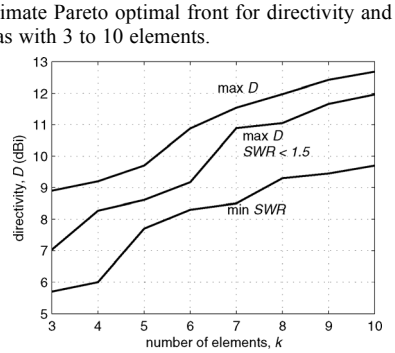

Fig. 3. Upper and lower bounds for Pareto optimal directivity, and also an intermediate Pareto point for maximum directivity with SWR no greater than 1.5.

of cross-section radius 3.175 mm designed to operate over a 7% bandwidth centered at 859 MHz. The results were achieved with the proposed MOEA/D using the same settings as in the previous benchmark tests.

Fig. 2 presents the approximate Pareto optimal fronts of 3 to 10 elements Yagi-Uda antennas for directivity (D) and standing wave ratio (SWR). As expected, an optimal Pareto front related to antennas with higher number of elements dominates a Pareto optimal front related to antennas with lower number of elements. Fig. 3 shows the lower and upper bounds for Pareto optimal directivity shown in Fig. 2, and also a typical practical choice, with maximum directivity with SWR no higher than 1.5.

V. CONCLUSION

The MOEA/D introduced in this work is based on a framework to solve multi-objective problems and evolutionary techniques that are widely used in the literature. Its resulting Pareto front samples were fairly accurate for the number of function analysis, both in benchmark tests and in designing of Yagi-Uda antennas.

The behavior of the proposed MOEA/D can by customized by its input weight vectors, the number of neighbors that can be updated and the update rule itself, the external population and the evolutionary operators. When the number of neighbors that can be updated is large, the search space is not explored very well, once the points become concentrated. When this number is small, some points may be pushed away from others during the optimization process and, hence, which may lead to a slower convergence. Another factor that impacts onto population diversity is the update rule. The proposal in this work allowed all refresh rates of neighbors to be balanced, which decreases an undesired stagnation.

ACKNOWLEDGMENT

This work was supported by CAPES, FAPEMIG, and CNPq, Brazil. It was developed in the Graduate Program in Electrical Engineering of the Federal University of Minas Gerais.

REFERENCES

[1] R. M. Ramos, R. R. Saldanha, R. H. C. Takahashi, and F. J. S. Moreira, "The real-biased multiobjective genetic algorithm and its application to the design of wire antennas," *IEEE Trans. Magn.*, vol. 39, no. 3, pp. 1329–1332, 2003.

[2] J. D. Lohn, W. F. Kraus, and S. P. Colombano, "Evolutionary optimization of Yagi-Uda antennas," in *Proc. 4th Int. Conf. Evolvable Syst.*, 2001, pp. 236–243.

[3] S. K. Goudous, K. Siakavara, E. Vafiadis, and J. N. Sahalos, "Pareto optimal Yagi-Uda antenna design using multi-objective differential evolution," *Progr. Electromagn. Res.*, vol. 105, pp. 231–251, 2010.

[4] A. C. Lisboa, D. A. G. Vieira, J. A. Vasconcelos, R. R. Saldanha, and R. H. C. Takahashi, "Monotonically improving Yagi-Uda conflicting specifications using the dominating cone line search method," *IEEE Trans. Magn.*, vol. 45, no. 3, pp. 1494–1497, 2009.

[5] C. Chen, Y. Chen, and Q. Zhang, "Enhancing MOEA/D with guided mutation and priority update for multi-objective," in *CEC2009 MOEA Competition*, Trondheim, Norway, 2009.

[6] G. Thiele, "Analysis of Yagi-Uda-type antennas," *IEEE Trans. Magn.*, vol. 17, no. 1, pp. 24–31, 1969.

[7] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[8] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics (American Statistical Association)*, vol. 21, no. 2, pp. 239–245, 1979.

[9] R. Storn and K. Price, "Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces," ICSI, Tech. Rep. TR-95-012, 1995.

[10] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition," Sch. Comput. Sci. Electr. Eng., Univ. Essex, U.K., Tech. Rep. CES-887, 2008.