



UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
<http://www.dit.unitn.it>

SYNTHESIS OF SUM AND DIFFERENCE PATTERNS
FOR MONOPULSE ANTENNAS BY AN HYBRID
REAL/INTEGER-CODED DIFFERENTIAL
EVOLUTION METHOD

Salvatore Caorsi, Andrea Massa, Matteo Pastorino,
and Andrea Randazzo

July 2003

Technical Report # DIT-03-043

Also: submitted for consideration for publication IEEE Trans. on
Antennas Propagation.

Synthesis of sum and difference patterns for monopulse antennas by an hybrid real/integer-coded differential evolution method

Salvatore Caorsi, *Member, IEEE*, Andrea Massa, *Member, IEEE*, Matteo Pastorino, *Senior Member, IEEE*, and Andrea Randazzo

Abstract – The synthesis of sum and difference patterns of monopulse antennas is considered in this paper. The synthesis problem is recast as an optimization problem by defining a suitable cost function based on the constraints on the side lobe levels. A subarray configuration is considered and the excitations of the difference pattern are approximately determined. The optimization problem is efficiently solved by a differential evolution algorithm, which is able to contemporarily handle real and integer unknowns. Numerical results are reported considering classic array configurations previously assumed in the literature.

Index Terms – Array antennas, Monopulse antennas, Evolutionary algorithm, Sum and difference patterns.

I. INTRODUCTION

In designing monopulse antennas, the sum pattern and the difference pattern have to be synthesized. Recently, a number of methods have been proposed in which proper feed networks avoid the need for the completely independent implementation of the two arrays [1]-[3].

Among these methods, the one proposed by Lopez et al. [3] is based on a subarray configuration and uses a standard binary genetic algorithm to determinate the weights of the subarrays.

Essentially, the synthesis problem concerns the calculation of one of the excitation sets (for the sum or difference pattern) assuming known (and optimum) the other excitation set and using a subarray configuration to reduce the feeding complexity. However, one of the two patterns (in this paper, the difference pattern) is obtained with a certain approximation and a good compromise between the complexity of the configuration and the quality of the pattern is the goal of the synthesis.

Following the approach proposed in [3], the synthesis problem is recast as an optimization problem in which a functional is constructed and minimized in order to define, for each array element, the corresponding subarray, the weights of all the subarrays and, consequently, the excitation sets of one of the patterns (in this case, the difference pattern). The functional to be minimized is constructed on the bases of the constraints on the pattern to be generated, e.g., on the bases of the required side lobe level.

The key point is now the optimization of the constructed functional, which in the present paper is obtained by an efficient real-coded differential evolution algorithm.

The differential evolution algorithm belongs to the class of genetic and evolutionary algorithms [4], which are stochastic algorithms widely used in antenna synthesis and in other areas of the applied electromagnetics.

The use of real-code implementations is generally considered an advance over the binary coding. In particular, the genetic operator acting on a binary-coded representation of the solution (chromosome), do not assure that the chromosomes of the new population are admissible solutions. Moreover, if acceptable solutions have to belong to some domains of the solution space (e.g., in order to fulfill the synthesis requirements), monitoring this property during the action of the genetic operators can be laborious and time consuming (a decoding is required). The convergence is usually slowed.

Moreover, the differential evolution algorithm exhibits some interesting feature (discussed in Section III) making the approach to be particularly worth using in synthesis applications.

The approach has been introduced in [5]-[7] and a discussion on capabilities and limitation can be found in [6].

In the following, the used notation and the synthesis method are described in Section II, whereas in Section III the application of the differential evolution algorithm is presented in details. Finally, in Section IV, several numerical results are reported to assess the effectiveness of the proposed synthesis approach.

II. MATHEMATICAL FORMULATION

A linear array of $M=2N$ elements γ_n , $n = -N, \dots, -1, 1, \dots, N$, is considered. For this kind of structures, the array factor $F(\theta)$ is defined by [8]

$$F(\theta) = \sum_{n=-N}^{-1} a_n e^{j\left(n+\frac{1}{2}\right)kd \cos \theta} + \sum_{n=1}^N a_n e^{j\left(n-\frac{1}{2}\right)kd \cos \theta} \quad (1)$$

where a_n , $n = -N, \dots, -1, 1, \dots, N$, are the excitations of the radiating elements, k is the wavenumber of the medium in which the antenna is located, d is the distance between the elements and θ defines the angle at which F is calculated with respect to a direction orthogonal to the array.

The required sum pattern is obtained by the excitations a_n^s , $n = -N, \dots, -1, 1, \dots, N$, which are assumed to be symmetric

(i.e., $a_{-n}=a_n$, $n=1, \dots, N$) and fixed. Under these hypothesis, for the sum pattern, relation (1) reduces to

$$F_s(\theta) = \sum_{n=1}^N a_n^s \cos \left[\frac{1}{2}(2n-1)kd \cos \theta \right] \quad (2)$$

where s indicates the sum pattern mode. The difference pattern is achieved by means of a subarray structure, as shown in Figure 1.

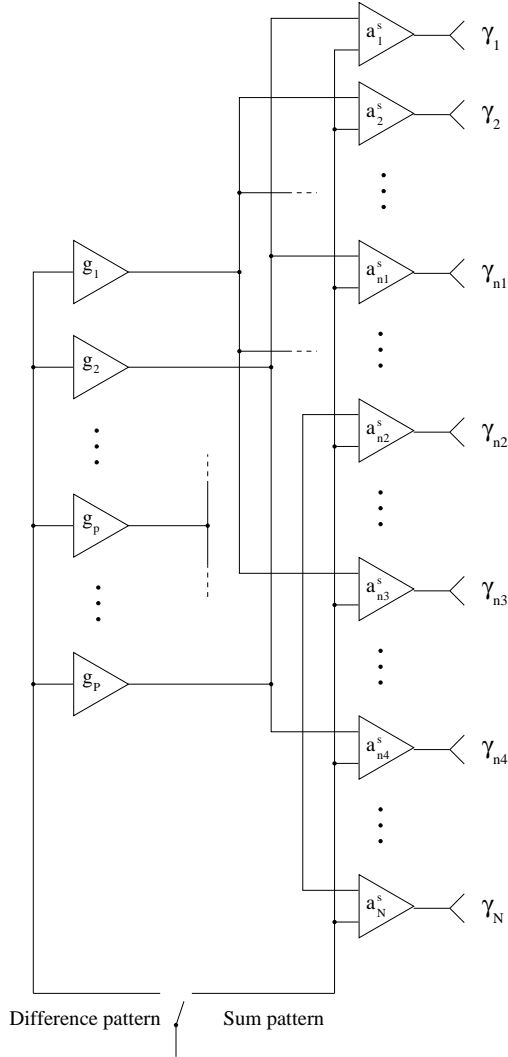


Figure 1 – Array configuration.

A positive integer c_n is associated to each element and denotes the related subarray. In particular, $0 \leq c_n \leq P$, being P the total number of subarrays. If $c_n = 0$, the element is not considered in the synthesis.

Consequently, each subarray p has associated a subset of the set of available elements Γ , in the following indicated by $\Gamma(p)$. In particular, $\Gamma(p)$ is given by $\Gamma(p) = \{ \gamma_n : c_n = p, c_n \in \mathbb{N}^+, n=1, \dots, N \}$.

The subsets are assumed to be non-overlapping, i.e. $\Gamma(p) \cap \Gamma(q) = 0$ if $p \neq q$. Furthermore, $\bigcup_{p=1}^P \Gamma(p) \subseteq \Gamma$ since the partition is not requested to cover the whole set of elements Γ .

The excitations used to generate the difference pattern are obtained by:

$$a_n^d = a_n^s \sum_{p=1}^P 1_{\Gamma(p)}(\gamma_n) g_p \quad n=1, \dots, N \quad (3)$$

where $1_{\Gamma(p)}(\cdot)$ denotes the characteristic function of the subset $\Gamma(p)$.

In order to obtain a difference pattern, the excitations must be antisymmetric, i.e. $a_{-n} = -a_n$, $n=1, \dots, N$. In this case, equation (1) reduces to

$$F_s(\theta) = \sum_{n=1}^N a_n^d \sin \left[\frac{1}{2}(2n-1)kd \cos \theta \right] \quad (4)$$

Since in both cases there is a symmetry between the excitation values, only one half of the array is considered in the following.

The weight of the p -th subarray, g_p , $p=1, \dots, P$, and the associations of the elements to the subarrays must be optimized in order to create a difference pattern that is as much closer as possible to the desired one.

The problem of the array synthesis is recast as an optimization problem by means of the definition of an appropriate cost function that relates the array parameters to the required features.

In many cases of practical interest, one of the constraints that must be satisfied is the presence of a side lobe level (SLL) with a prescribed value. To this end, according to [1], in the present paper, the following cost function is defined

$$f(\Gamma(1), \dots, \Gamma(P), a_1^d, \dots, a_N^d) = \left[SLL(\Gamma(1), \dots, \Gamma(P), a_1^d, \dots, a_N^d) - SLL_d \right] H(SLL(\Gamma(1), \dots, \Gamma(P), a_1^d, \dots, a_N^d) - SLL_d) \quad (5)$$

where SLL_d is the desired side lobe level value and $H(\cdot)$ represents the Heaviside step function.

Obviously, it is possible to introduce other type of constraints in the synthesis of the array space factor by simply adding the corresponding terms in the cost function. The cost function (5) is minimized by means of an efficient stochastic procedure, named differential evolution.

III. DIFFERENTIAL EVOLUTION ALGORITHM

The differential evolution (DE) algorithm [5]-[7] is a method that belongs to the class of evolutionary global optimization techniques. The principal difference between the DE and the others population-based methods, such as genetic algorithms, is represented by the scheme used for

the generation of the trial parameters vector. A schematic flow chart of the algorithm is reported in Figure 2.

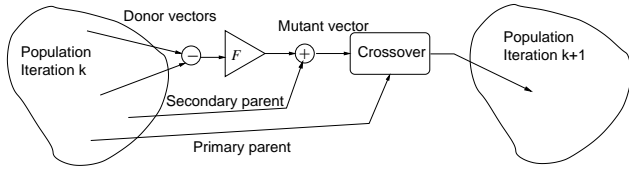


Figure 2 – Schematic representation of the DE algorithm.

In the initialization phase, a set of N_p trial solutions, called population, is randomly created.

Let $\mathbf{u}_k(i)$, $i=1, \dots, N_p$, be the i -th element of the set of trial solutions at the k -th iteration of the algorithm. In the reproduction phase, $\mathbf{u}_k(i)$ is called primary parent. Furthermore, let $\mathbf{u}_k(i_1)$, $\mathbf{u}_k(i_2)$ (called donor vectors) and $\mathbf{u}_k(i_3)$ (called secondary parent) be three randomly chosen elements of the population.

The i -th element of the population at the $k+1$ iteration is generated according to the following rule

$$\begin{aligned} \mathbf{m}_k(i) &= (\mathbf{u}_k(i_1) - \mathbf{u}_k(i_2))F + \mathbf{u}_k(i_3) \\ \tilde{\mathbf{u}}_{k+1}(i) &= \mathbf{m}_k(i) \times \mathbf{u}_k(i) \\ \text{if } (f(\tilde{\mathbf{u}}_{k+1}(i)) < f(\mathbf{u}_k(i))) &\text{ then} \\ \mathbf{u}_{k+1}(i) &= \tilde{\mathbf{u}}_{k+1}(i) \\ \text{else} \\ \mathbf{u}_{k+1}(i) &= \mathbf{u}_k(i) \end{aligned} \quad i = 1, \dots, N_p \quad (6)$$

where \times denotes the crossover operator, F is a weighting factor and the quantity $\mathbf{m}_k(i)$ is called mutant vector.

Rule (6) is executed for each element of the population at the k -th iteration. The algorithm stops when the cost function f of the best element of the population is below a threshold value f_{th} or when a maximum value of iterations k_{max} is reached.

In the problem analyzed in this paper, the array $\mathbf{u}_k(i)$, $i=1, \dots, N_p$, contains the unknown weights and the configuration of the subarray $\Gamma(p)$, $p=1, \dots, P$.

Since the subarray configuration is defined by a set of integer values c_1, \dots, c_N , each array $\mathbf{u}_k(i)$ is an hybrid real/integer chromosome of $P+N$ elements, i.e.

$$\mathbf{u}_k(i) = (\mathbf{g}_1, \dots, \mathbf{g}_P, c_1, \dots, c_N)^t \quad (7)$$

This particular structure of $\mathbf{u}_k(i)$ leads to a problem that belongs to the class of mixed integer programming [9], which are proved to be particularly hard to solve with the classic optimization methods.

One important feature of the DE algorithm is that it can easily be used to solve mixed integer programming problems by simply defining an appropriate crossover operator.

For the problem discussed in this work, the crossover operator is defined by

$$\begin{aligned} \mathbf{m}_k(i) \times \mathbf{u}_k(i) &= (t_1, \dots, t_{P+N})^t \\ t_j &= \begin{cases} \lfloor [\mathbf{m}_k(i)]_j + 0.5 \rfloor & \text{with probability } CR \\ \lfloor [\mathbf{u}_k(i)]_j + 0.5 \rfloor & \text{with probability } 1-CR \end{cases} \quad j = 1, \dots, P \\ t_j &= \begin{cases} \lfloor [\mathbf{m}_k(i)]_j \rfloor & \text{with probability } CR \\ \lfloor [\mathbf{u}_k(i)]_j \rfloor & \text{with probability } 1-CR \end{cases} \quad j = P+1, \dots, P+N \end{aligned} \quad (8)$$

where the operator $[\cdot]_j$ extract the j -th element of the vector on which it is applied and the operator $\lfloor \cdot \rfloor$ gives the greatest integer number below the argument. CR is a parameter that controls the execution of the crossover and can be tuned by the user. Operator $\lfloor \cdot \rfloor$ is used in order to ensure that all the elements corresponding to the integer-valued part of the chromosome belong to \mathfrak{X} . In this case, $\lfloor \cdot \rfloor$ returns the nearest integer value in the interval $[0, P]$.

Furthermore, in order to assess the feasibility of the test vector $\tilde{\mathbf{u}}_{k+1}(i)$ obtained by the reproduction procedure (6) with respect to the constraints (that in this case are only bounds on the variables, i.e. $l_j \leq [\tilde{\mathbf{u}}_{k+1}(i)]_j \leq u_j$, $j=1, \dots, P+N$), a bound check, defined by the following relation, is performed.

$$[\tilde{\mathbf{u}}_{k+1}(i)]_j = \begin{cases} [\tilde{\mathbf{u}}_{k+1}(i)]_j & \text{if } l_j \leq [\tilde{\mathbf{u}}_{k+1}(i)]_j \leq u_j \\ l_j & \text{if } [\tilde{\mathbf{u}}_{k+1}(i)]_j < l_j \\ u_j & \text{if } [\tilde{\mathbf{u}}_{k+1}(i)]_j > u_j \end{cases} \quad (9)$$

$$j = 1, \dots, P+N$$

This version of the DE algorithm is also called DE/rand/1/bin [5]; this name summarizes that the secondary parent is chosen in a random way, the mutation vector is generated by means of one difference and the crossover is binary (single point crossover).

The parameters of the DE algorithm have been chosen in accordance with the guidelines given in [6]. In fact, as described in [6], the choice of the two parameters affects the behavior of the method, leading, if not properly tuned, to a premature convergence to a local minima or to a slow convergence rate.

DE algorithms exhibit several advantages with respect to other evolutionary methods such as genetic algorithms. First of all, they do not require a selection operator, i.e., a function, generally random, that chooses the elements that must generate the next individuals, but uses all the trial solution vectors as primary parents once and in a deterministic way. Furthermore, the mutation vector is not obtained by the summation of a random value, as in most genetic algorithms, but is generated by using only randomly chosen elements of the population. By this way, the mutation operator is not a “destructive” function (meaning that it changes the test vector in a blind way), but modify the trial solutions by taking into account the population

diversity. This kind of mutation permits a speed up in the convergence process, leading, however, to the possibility of a premature convergence to a local minima. In order to avoid this possibility, it is necessary to choose accurately the parameters CR and F .

Furthermore, DE algorithms contains a kind of implicit elitism. In fact, since the fitness of the trial vector $\tilde{\mathbf{u}}_{k+1}(i)$ is compared only with one of the primary parents $\mathbf{u}_k(i)$ from which it is generated, the corresponding element in the next generation has a fitness that can only be better or at least equal to the one of $\mathbf{u}_k(i)$.

IV. NUMERICAL SIMULATIONS

In order to validate the proposed approach, some numerical simulations have been performed.

As a first case, a linear array of $M=100$ elements with spacing $\lambda/2$ has been used. The sum pattern excitations $a_n^s, n=-N, \dots, -1, 1, \dots, N$, correspond to a Taylor pattern [8] with $\bar{n}=12$ and $SSL=-35$ dB. The same synthesis problem has been considered in [3][10]. Figure 3 reports the array space factor corresponding to the considered configuration.

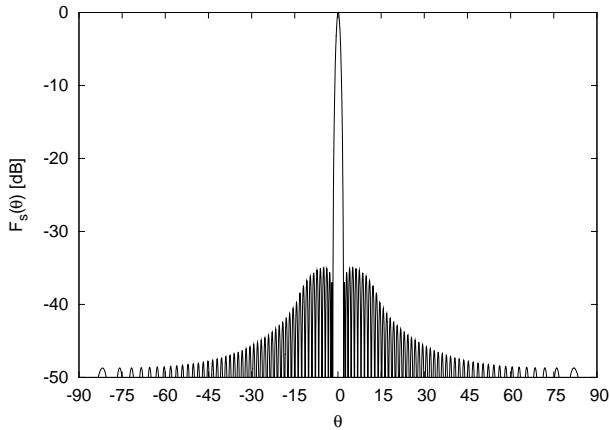


Figure 3 – Array space factor for a Taylor pattern with $\bar{n}=12$ and $SSL=-35$ dB.

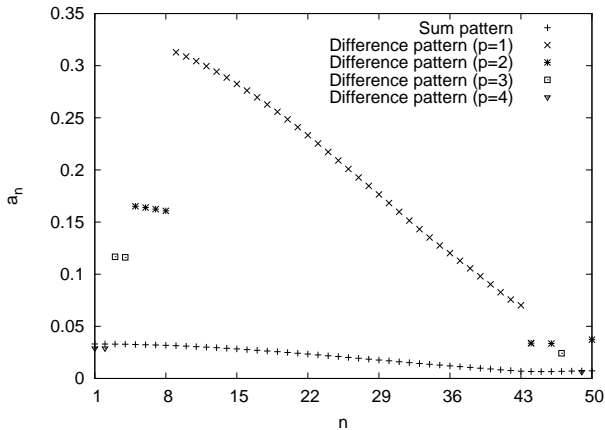


Figure 4 – Excitations of the elements of the array for both sum (Figure 3) and difference patterns (Figure 5).

The parameters of the DE algorithm used for this simulations are $CR = 0.7$, $F = 0.5$, $f_{th} = 10^{-4}$ and $k_{max} = 1000$. Furthermore, the subarray number is set equal to $P=4$.

Figure 4 shows the values of the excitations of the difference pattern $a_n^d, n = 1, \dots, N$, obtained by means of the proposed approach, together with the sum pattern excitations for $SLL_d = -30$ dB. Furthermore, Figure 4 contains the information about the distribution of the array elements γ_n in each subarray. In particular, each symbol corresponds to the specific subarray at which the antenna is associated.

Figure 5 reports the array space factor that corresponds to the values of excitations shown in Figure 4. As can be seen, the resulting array factor has a side lobe level that is lower than the desired value, confirming the ability of the optimization process, which is mainly due to the use of a real coding of the values g_p .

Finally, Figure 6 reports the behavior of the cost function during the minimization process.

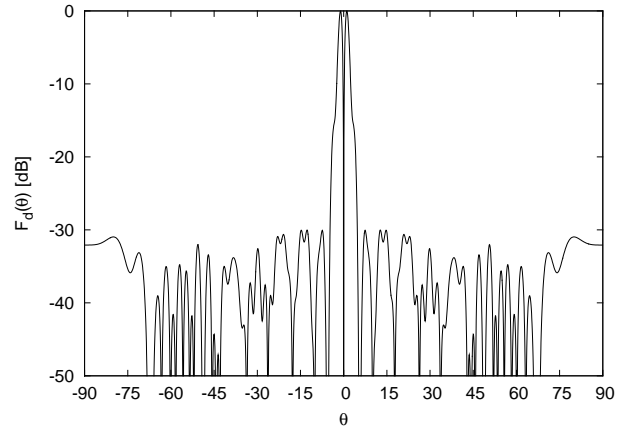


Figure 5 – Difference pattern obtained by the DE algorithm for a Taylor sum pattern. $SLL_d = -30$ dB.

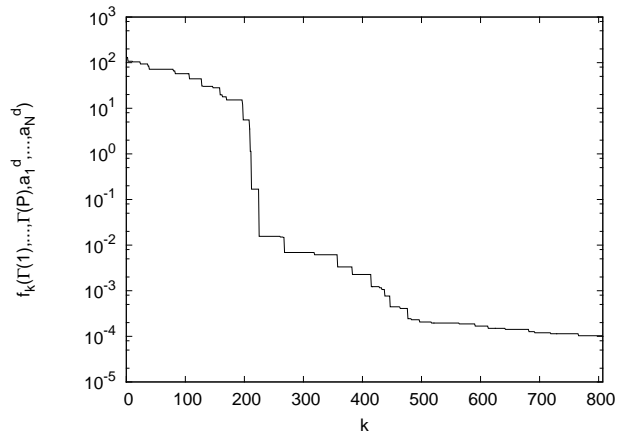


Figure 6 – Behavior of the cost function (equation (5)) versus the iteration number k .

A second simulation has been performed using a sum pattern corresponding to a Dolph-Tschebyscheff array with $M = 20$ elements, $d = \lambda/2$ and $SLL = -35$ dB, whose array space factor is shown in Figure 7.

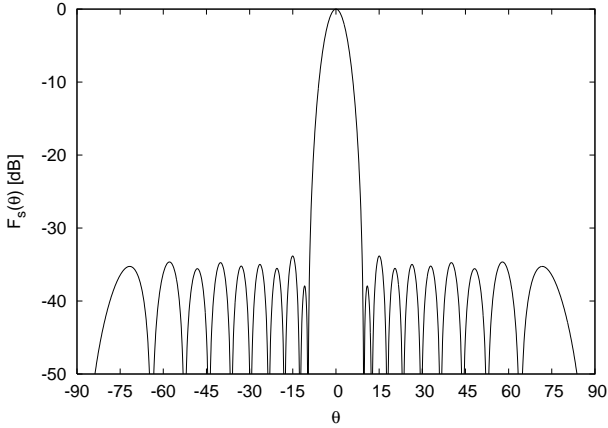


Figure 7 – Space factor for a Dolph-Tschebyscheff array with $M = 20$ element and $SLL = -35$ dB.

The number of subarrays has been fixed equal to 4. As for the previous example, the parameters of the DE algorithms are $CR = 0.7$, $F = 0.5$, $f_{th} = 10^{-4}$ and $k_{max} = 1000$. The value of the desired side lobe level is set to $SLL_d = -30$ dB.

The optimization procedure reaches the convergence after $K_{opt} = 12$ iterations, and the difference array space factor corresponding to that solution is shown in Figure 8.

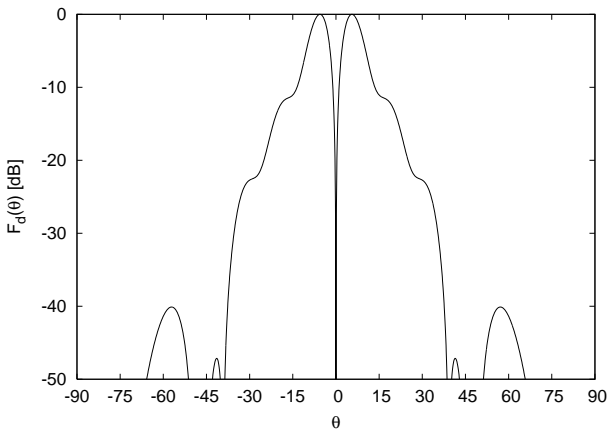


Figure 8 – Array space factor for the difference configuration.

As can be seen from this picture, the constraints on the side lobe level is satisfied. The excitation values related to the array space factor are reported in Figure 9. It is worth noting that, in this case, the synthesis procedure has led to a configuration in which one of the subarrays ($p = 2$) is not used, i.e., no elements are associated to it. The same configuration has been treated in [10].

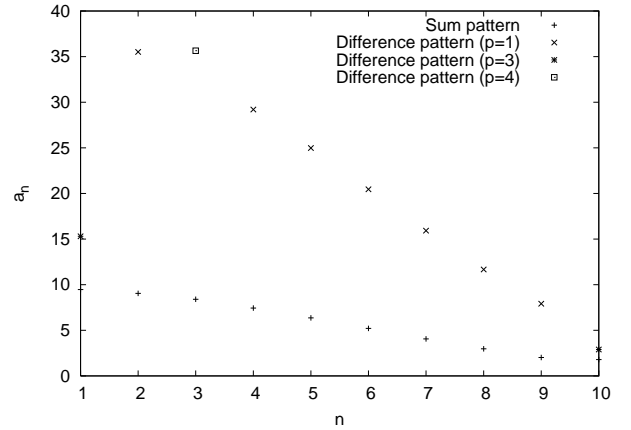


Figure 9 – Excitations for the obtained difference pattern and for the sum pattern of Figure 7.

The behavior of the cost function versus the number of iteration is reported in Figure 10. As can be seen, only 12 iterations of the DE algorithm are sufficient to reach the required accuracy.

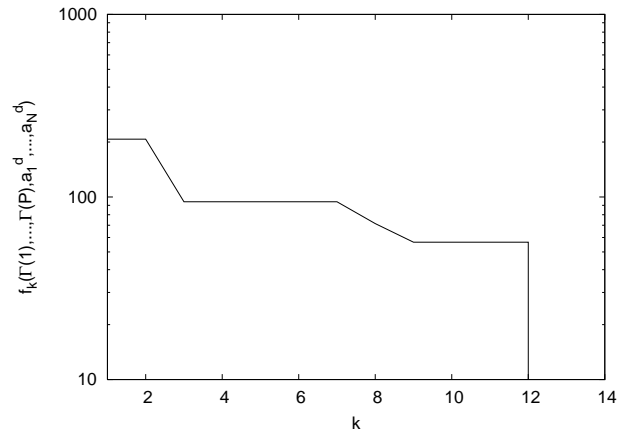


Figure 10 – Behavior of the cost function (equation (5)) versus the iteration number k .

In order to study the behavior of the synthesis procedure versus the number of subarray, a more complete simulation have been performed using the previous configuration. In particular, the number of subarrays has been set equal to $P=2$, $P=4$, $P=6$, $P=8$, and $P=10$, and the results have been compared.

Figure 11 shows the synthesized arrays and Figure 12 the corresponding excitations. As can be seen from this picture, when using only two subarrays the proposed method is not able to reach a solution that satisfy the constrain $SLL < SLL_d = -30$ dB. However, for all the other configurations, a satisfactory pattern is obtained.

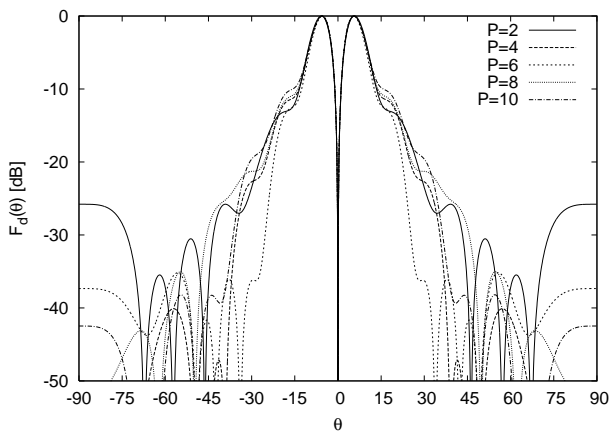


Figure 11 – Array factor for the difference pattern using different number of subarrays.

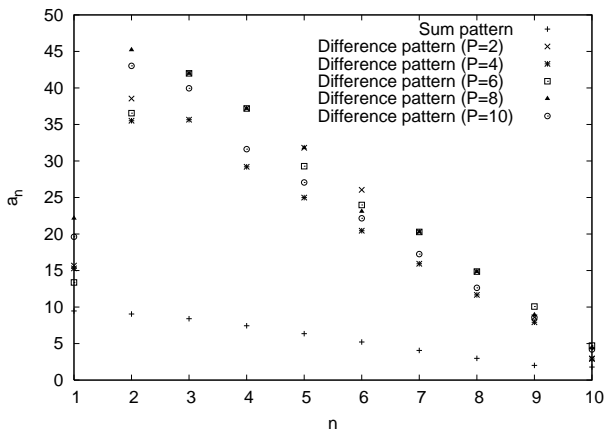


Figure 12 – Excitations for the array factor of Figure 11.

Finally, Figure 13 reports some information on the convergence process showing how the cost function decreases during the minimization procedure.

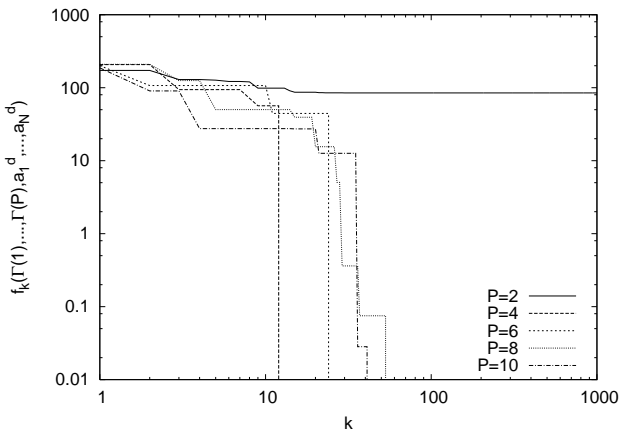


Figure 13 – Behavior of the cost function (equation (5)) versus the iteration number k .

Finally, another comparative simulation using a Dolph-Tschebyscheff array, with $M=20$ element and $SLL = -20$ dB, has been performed. The array space factor for such a configuration is shown in Figure 14

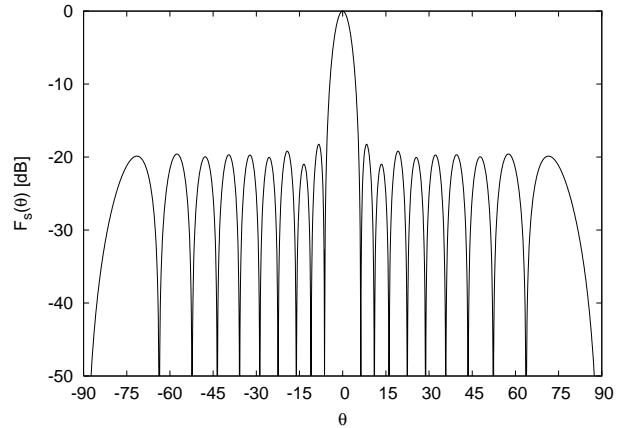


Figure 14 – Array space factor for a Dolph-Tschebyscheff array with $M = 20$ element and $SLL = -20$ dB.

The number of subarrays that has been considered is $P=2, P=4, P=6, P=8,$ and $P=10$, as in the previous case. The desired value of side lobe level has been set to $SLL_d = -20$ dB and the parameters of the DE algorithm have been kept equal to the ones used in the previous simulations. The array space factors obtained by the minimization procedure are reported in Figure 15.

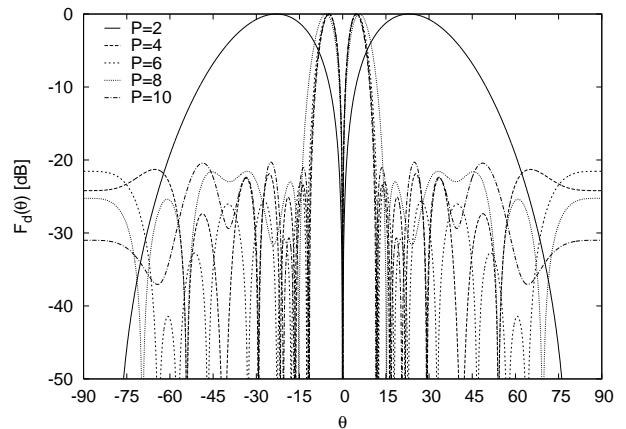


Figure 15 – Array space factor for the difference pattern for various values of P (number of subarrays).

As can be seen from this picture, in each case the method is able to reach a solution that satisfy the constrain on the side lobe level. However, when only two subarrays have been used, the resulting space factor have an aperture of the central lobes that is unacceptable in almost of the practical situations in which this kind of antennas are used. When the number of subarrays grows, this drawback is avoided, as

can be seen from Figure 15, in which all of the synthesized array have the first null at about 15° .

Figure 16 reports the value of the excitation coefficients for all of the array of Figure 15, compared to the ones of the sum pattern.

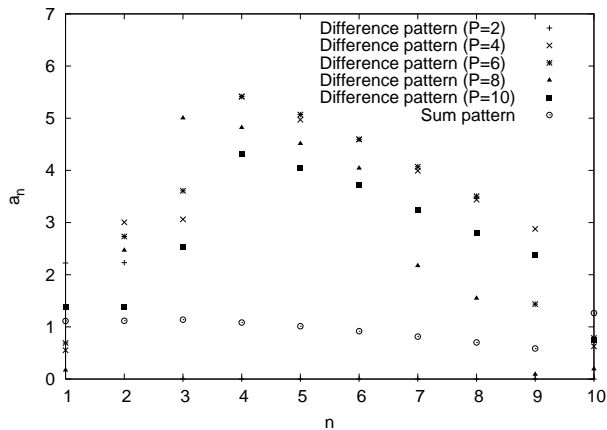


Figure 16 – Excitations for the array factor of Figure 15.

Finally, for completeness, Figure 17 reports the behavior of the cost function for these simulations.

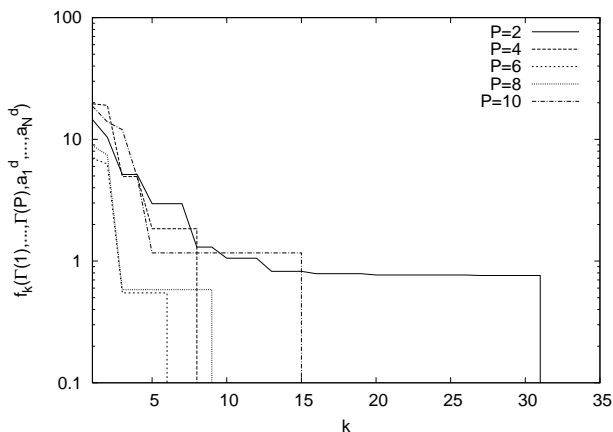


Figure 17 – Behavior of the cost function (5) versus the iteration number k .

V. CONCLUSIONS

In this paper, the synthesis of sum and difference patterns of monopulse antennas has been considered. In particular, following a subarray approach previously developed, the synthesis problem has been recast as an optimization problem solved by an efficient hybrid real/integer differential evolution algorithm, which represents an advance over the commonly used binary genetic algorithm. The method has been checked versus several array configurations. The results have shown that the requirements concerning the side lobe level have been fulfilled with good accuracy even by using a simplified subarray configuration. The method has been applied, in

particular, to Taylor and Dolph-Tschebyscheff arrays previously reported in the literature. Further development will concern the construction of different cost functions based on different constraints.

VI. REFERENCES

- [1] D. A. McNamara, "Synthesis of sum and difference patterns for two-section monopulse arrays," *Inst. Elect. Eng. Proc.*, vol. 135, pt. H, no. 6, pp. 371–374, Dec. 1988.
- [2] D. A. McNamara, "Synthesis of sub-arrayed monopulse linear arrays through matching of independently optimum sum and difference excitations," *Inst. Elect. Eng. Proc.*, vol. 135, pt. H, no. 5, pp. 293–296, Oct. 1988.
- [3] P. Lòpez, J. A. Rodriguez, F. Ares, and E. Moreno, "Subarray weighting for the difference patterns of monopulse antennas: Joint optimization of subarray configurations and weights," *IEEE Trans. Antennas Propagat.*, vol. 49, no. 11, pp. 1606–1608, Nov. 2001.
- [4] J. M. Johnson, Y. Ramat-Samii, "Genetic algorithms in engineering electromagnetics," *IEEE Antennas Propagat. Mag.*, vol. 39, no. 4, pp. 7–21, Aug. 1997.
- [5] R. Storn, and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimization*, Kluwer Academic Publishers, vol. 11, pp. 341–359, 1997.
- [6] R. Storn, "On the usage of differential evolution for function optimization," *NAFIPS 1996*, Berkeley, pp. 519 – 523, 1996.
- [7] K. A. Michalski, "Electromagnetic imaging of elliptical-cylindrical conductors and tunnel using a differential evolution algorithm," *Microwave Opt. Technol. Lett.*, vol. 28, no.3, pp. 164–169, Feb. 2001.
- [8] C. A. Balanis, *Antenna theory: analysis and design*, New York, Wiley, 1982.
- [9] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, 1988.
- [10] F. Ares, J. A. Rodriguez, and E. Moreno, "Optimal compromise among sum and difference patterns," *J. Electromagnetic Waves and Appl.*, vol. 10 pp. 1543–1555, 1996.