Ph.D. Dissertation

International Doctoral School in Information and Communication Technology

DISI - University of Trento

Innovative Combinatorial Strategies for the Synthesis of Radar Tracking Antenna Systems

Paolo Rocca

Advisor:
Andrea Massa, Professor
Università degli Studi di Trento
Co-Advisor:
Francesco G. B. De Natale, Professor
Università degli Studi di Trento

November 2008

Acknowledgements

I would like to thank Prof. A. Massa and the whole Eledia group for this unforgettable experience of work and life.

I thank all the friends, teachers and colleagues I met during my PhD.

I would also thank Prof. T. Isernia and Prof. F. Ares for their suggestions and indications which significantly improved this work of thesis. Besides his precious teaching, a particular thank you goes to Prof. R. L. Haupt for his kindness and hospitality during my staying at the Penn State University.

I also thank Prof. D A. McNamara for inspiring this work of thesis and for the kind discussions. A sincere thank you goes to Dr. A. G. Derneryd for providing some interesting contributes on the topic.

The greatest thank you is for my family and for Alessia.

Abstract

In the framework of the synthesis of monopulse array antennas for search-and-track applications, the thesis focuses on the development and the analysis of a method based on the sub-arraying technique aimed at generating an optimal sum and compromise difference patterns through an excitation matching procedure. By exploiting some properties of the solution space, the synthesis problem is reformulated as a combinatorial one to allow a considerable saving of computational resources. Thanks to a graph-based representation of the solution space, the use of an efficient path-searching algorithm to speed-up the convergence of the procedure for the synthesis of large array antennas as well as the use of the Ant Colony Optimizer (ACO) to benefit of its hill-climbing properties in dealing with the non-convexity of the sub-arraying problem are considered. Moreover, a hybrid approach is developed to individually control the level of the secondary lobes. In particular, the sub-array configuration is determined at the first step by exploiting the knowledge of the optimum difference mode coefficients and in the second step, the sub-array weights are computed by means of a quadratic programming procedure. In the numerical validation, a set of representative examples concerned with both pattern matching problems and pattern-feature optimization are reported in order to assess the effectiveness and flexibility of the proposed approach. Comparisons with previously published results are reported and discussed, as well.

Keywords
Monopulse array antennas, sum and difference patterns, excitation matching, contiguous partition, hybrid optimization.

Published Journals Papers

[R1] P. Rocca, L. Manica, and A. Massa, "Synthesis of monopulse antennas through the iterative contiguous partition method," *Electronics Letters*, vol. 43, no. 16, pp. 854-856, August 2007.

[R2] P. Rocca, L. Manica, A. Martini, and A. Massa, "Synthesis of large monopulse linear arrays through a tree-based optimal excitations matching," *IEEE Antennas and Wireless Propagation Letters*, vol. 7, pp. 436-439, 2007.

[R3] L. Manica, P. Rocca, A. Martini, and A. Massa, "An innovative approach based on a tree-searching algorithm for the optimal matching of independently optimum sum and difference excitations," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 1, pp. 58-66, January 2008.

[R4] P. Rocca, L. Manica, and A. Massa, "Hybrid approach for sub-arrayed monopulse antenna synthesis," *Electronics Letters*, vol. 44, no. 2, pp. 75-76, January 2008.

[R5] P. Rocca, L. Manica, and A. Massa, "An effective excitation matching method for the synthesis of optimal compromises between sum and difference patterns in planar arrays," *Progress in Electromagnetic Research B*, vol. 3, pp. 115-130, 2008.

[R6] P. Rocca, L. Manica, and A. Massa, "Directivity optimization in planar sub-arrayed monopulse antenna," *Progress in Electromagnetic Research L*, vol. 4, pp. 1-7, 2008.

[R7] P. Rocca, L. Manica, F. Stringari, and A. Massa, "Ant colony optimization for tree-searching based synthesis of monopulse array antenna," *Electronics Letters*, vol. 44, no. 13, pp. 783-785, June 19, 2008.

[R8] L. Manica, P. Rocca, and A. Massa, "On the synthesis of sub-arrayed planar array antennas for tracking radar applications," *IEEE Antennas and Wireless Propagation Letters*, (accepted 25/04/08).

[R9] L. Manica, P. Rocca, and A. Massa, "An excitation matching procedure for sub-arrayed monopulse arrays with maximum directivity," *IET Radar, Sonar & Navigation*, (accepted 05/08/08).

[R10] P. Rocca, L. Manica, M. Pastorino, and A. Massa, "Boresight slope optimization of sub-arrayed linear arrays through the contiguous partition method," *IEEE Antennas and Wireless Propagation Letters*, (accepted 18/07/08).

[R11] P. Rocca, L. Manica, A. Martini, and A. Massa, "Compromise sum-difference optimization through the iterative contiguous partition method," *IET Microwaves, Antennas & Propagation*, (accepted 29/08/08).

[R12] P. Rocca, L. Manica, R. Azaro, and A. Massa, "Hybrid approach for the synthesis of sub-arrayed monopulse linear arrays," *IEEE Transactions on Antennas and Propagation*, (accepted 23/09/08).

[R13] L. Manica, P. Rocca, M. Benedetti, and A. Massa, "A fast graph-searching algorithm enabling the efficient synthesis of sub-arrayed planar monopulse antennas," *IEEE Transactions on Antennas and Propagation*, (accepted 08/10/08).

Contents

List of Tables

List of Figures

Structure of the Thesis

The thesis is structured in chapters according to the organization detailed in the following.

The first chapter deals with an introduction to the thesis, focusing on the main motivations and on the subject of this work as well as a presentation of the state-of-the-art techniques dealing with the same antenna synthesis problem.

Chapter 2 presents the proposed excitation matching method for the synthesis of monopulse linear array antennas, focusing on the definition of the solution space as a non-complete binary tree as well as on the deterministic searching algorithm.

In Chapter 3 the proposed method is integrated in an iterative matching approach integrated in an iterative procedure ensuring, at the same time, the optimization of the sidelobe level (or other beam pattern features). The flexibility and effectiveness of such an approach are pointed out in the numerical validation through an extensive set of comparative examples.

The extension of the approach from linear to planar arrays is described and assessed in Chapter 4. A more compact graph structure is considered starting from the observation that some parts of the non-complete binary tree are recursively shared in it, enabling the synthesis of arrays with a large number of elements. Accordingly, the searching strategy is customized to look for the best compromise solution within the graph.

Chapter 5 deals with the presentation of an ant colony metaheuristic used to benefit of its hill-climbing properties in dealing with the non-convexity of the sub-arraying as well as in managing graph searches.

A hybrid approach for the synthesis of linear and planar monopulse array antennas is presented in Chapter 6. At the first step, the sub-array configuration is determined by means of the proposed excitation matching method. In the second step, the sub-array weights are computed through the solution of a convex programming problem for a fixed clustering to obtain a direct control on the behavior of the secondary lobes.

Conclusions and further developments are presented in Chapter 7. Finally, two appendices give more details on the definition of contiguous partition and on the dimension of the solution space.

Chapter 1

Introduction and State-of-the-Art

In the introduction, the motivation of the thesis is pointed out starting from a brief overview on the techniques presented in the state-of-the-art an regarding the solution of the optimal compromise problem between sum and difference patterns for the synthesis of sub-arrayed monopulse array antennas.

A monopulse tracker [1][2][3] is a device aimed at detecting the position of a target by using the information collected from an antenna that generates *sum* and *difference* patterns. These beams can be synthesized by means of a reflector antenna with two (tracking on a plane) or three (3D tracking) feeds, or by using linear or planar array antennas, respectively. The latter solution is usually preferred since array antennas are easy to built and they do not require mechanical positioning systems to steer the beam pattern. Moreover, array structures can also be easily installed on mobile vehicles (e.g., aircrafts). Unlike linear structures, a planar array allows the generation of a sum and two spatially-orthogonal difference patterns [4] [i.e., the *azimuth difference mode* ($H-mode$) and the *elevation difference mode* ($E-mode$)] useful to give a complete description of the trajectory of a target in terms of range, azimuth, and elevation. These patterns are required to satisfy some constraints as narrow beamwidth, low side lobe level (SLL) and high directivity. In particular, as far as the sum pattern is concerned, there is the need of maximizing the gain. On the other hand, the more critical issue to be addressed dealing with difference patterns is concerned with the normalized difference slope on boresight direction, since it is strongly related to the sensitivity of the radar (i.e., to the angular error).

In order to synthesize independent optimal sum and difference patterns, Taylor [5][6] and Bayliss [7] developed analytical techniques to compute the corresponding excitation coefficients by sampling suitable continuous distributions. However, these optimal solutions require three independent feeding networks. Hence, high manufacturing costs usually arise and electromagnetic interferences unavoidably take place because of the large number of elements in planar monopulse arrays. In order to overcome these drawbacks, the sub-arraying technique [8] is a suitable compromise solution aimed at optimizing pre-specified sub-array layouts.

As far as linear arrays are concerned, McNamara proposed in [8] an excitation matching method aimed at determining a *best compromise* difference pattern close as much as possible to the optimum in the Dolph-Chebyshev sense [9] (i.e., narrowest first null beamwidth and largest normalized difference slope on the boresight for a specified sidelobe level). Towards this end, for each possible grouping, the corresponding sub-arrays coefficients are iteratively computed through pseudo-inversion of an overdetermined system of linear equations. It is evident that since the best sub-array configuration is not *a-priori* known, the whole process is extremely time-expensive due to the exhaustive evaluations. Moreover, because of the ill-conditioning of the matrix system, large arrays cannot be easily managed. In order to overcome the ill-conditioning and related issues, optimization approaches have been widely used [10][11][12][13][14][15]. Although such techniques allows a significant advancement in the framework of sum-difference pattern synthesis, they are still time-consuming when dealing with large arrays. As a matter of fact, even though the solution space is sampled with efficient searching criteria, the dimension of the solution space is very large.

To overcome such drawbacks allowing an effective choice of the array elements grouping as well as a fast and simple solution procedure, an innovative approach is proposed in this chapter that, likewise [8] and unlike [10][11][12][13][14][15], is aimed at obtaining a compromise difference pattern optimum in the Dolph-Chebyshev sense [9] starting from the observation that the sub-arraying is not *blind*. As a matter of fact, it can be guided by considering similarity properties among the array elements, thus significantly reducing the dimension of the solution space. Starting from such an idea and by representing each solution by means of a path in a graph structure, the synthesis problem is then recast as the searching of the minimal-cost path within the graph.

Chapter 2

The Excitation Matching Approach - Linear Arrays

The approach presented in this chapter regards a strategy for the synthesis of sub-arrayed monopulse linear arrays based on the optimal matching of independently optimum sum and difference excitations. By exploiting the relationship between the independently optimal sum and difference excitations, the set of possible solutions is considerably reduced and the synthesis problem is recast as the search of the best solution in a non-complete binary tree. Towards this end, a fast resolution algorithm that exploits the presence of elements more suitable to change sub-array membership is presented. The results of a set of numerical experiments are reported in order to validate the proposed approach pointing out its effectiveness also in comparison with state-of-the-art optimal matching techniques.

2.1 Introduction

In this chapter, starting from the general idea pointed out in Section 1, it is demonstrated how the solution space can be represented by means of a non-complete binary tree, and consequently the synthesis problem is recast as the searching of the minimal-cost path from the root to the leafs of the solution tree.

Generally speaking, in graph theory, a tree is a graph defined as a non-empty finite set of vertexes or nodes in which any two nodes are connected by exactly one path. The nodes are labeled such that there is only one node called the *root* of the tree, and the remaining nodes are partitioned in subtrees. In our case, since the tree is either empty or each node has not more than two subtrees, it is a *binary tree*. Accordingly, each node of a binary tree has either (*i*) no children, or (*ii*) one left/right child (i.e., non-complete binary tree), or (*iii*) a left child and a right child (i.e., complete binary tree), each child being the root of a binary tree called a subtree [16]. In order to solve the problem at hand, thus efficiently exploring the solution tree, a suitable cost function or *metric* is defined and an innovative algorithm for the exploration of the solution space is defined by exploiting the *closeness* (to a sub-array) property of some elements, called *border elements*, of the array.

The chapter is organized as follows. In Section 2.2, the problem is mathematically formulated defining a set of metrics aimed at quantifying the closeness of each solution to the optimal one (Sect. 2.2.1) as well as the tree structure (Sect. 2.2.2) and the algorithm for effectively exploring the solution space (Sect. 2.2.3). In Section 2.3, the results of selected numerical experiments are reported and compared with those from state-of-the-art optimal matching solutions.

2.2 Mathematical Formulation

Let us consider a linear uniform array of $N = 2M$ elements $\{\xi_m;\ m = \pm 1, ..., \pm M\}$. Following a sub-optimal strategy, the sum pattern is generated by means of the symmetric set of the real optimal[1] excitations $\underline{A}^{opt} = \{\alpha_m;\ m = 1, ..., M\}$ [5][17], while the difference pattern is defined through an anti-symmetric real excitation set $\underline{B} = \{b_m = -b_{-m};\ m = 1, ..., M\}$ [7][9]. Thanks to such symmetry properties, one half of the elements of the array $\underline{S} = \{\xi_m;\ m = 1, ..., M\}$ is descriptive of the whole array.

Grouping operation yields to a sub-array configuration mathematically described in terms of the grouping vector $\underline{C} = \{c_m;\ m = 1, \ldots, M\}$, $c_m \in [1, Q]$ being the sub-array index of the m-th element of the array [11]. Successively, a weight coefficient w_q is associated to each sub-array, $q = 1, ..., Q$, and, as a consequence, the sub-optimal difference excitation set is given by (Fig. 2.1)

[1] In the Dolph-Chebyshev sense [9], unless mentioned elsewhere.

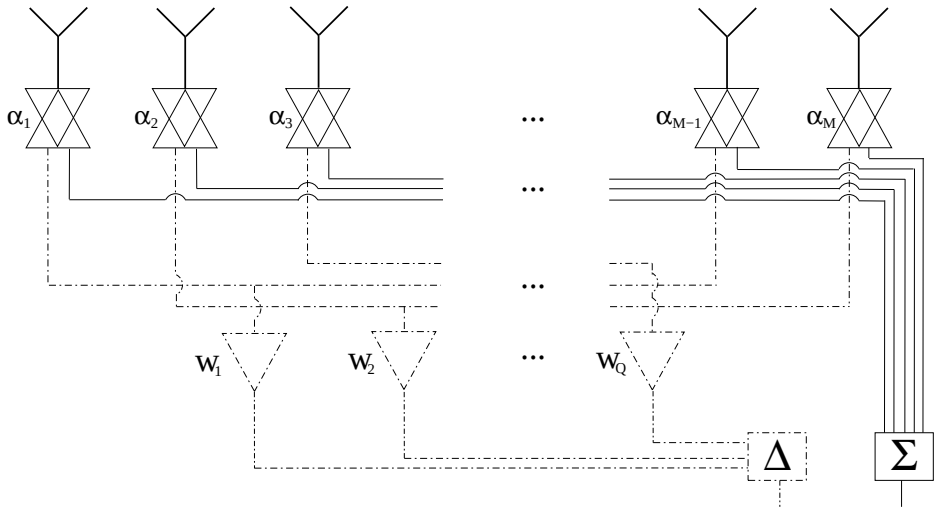


Figure 2.1: Sketch of the sub-arrayed monopulse linear array antenna.

$$\underline{B} = \{b_m = w_{mq}\alpha_m;\ m = 1, ..., M;\ q = 1, ..., Q\} \tag{2.1}$$

where $w_{mq} = \delta_{c_m q} w_q$ ($\delta_{c_m q} = 1$ if $c_m = q$, $\delta_{c_m q} = 0$ otherwise) is the weight associated to the m-th array element belonging to the q-th sub-array.

Accordingly, the original problem is recast as the definition of a sub-array configuration $\underline{C}$ and the corresponding set of weights $\underline{W} = \{w_q;\ q = 1, ..., Q\}$ such that the sub-optimal difference pattern $\underline{B}$ is as close as possible to the optimal one, $\underline{B}^{opt} = \{\beta_m;\ m = 1, ..., M\}$. Towards this end the problem metric is firstly defined in order to quantify the closeness of the sub-optimal solution to the optimal one. Then, exploiting some properties of the sub-array configurations, a non-complete binary tree, where each path codes a possible elements grouping, is built. Finally, a simple algorithm for a fast search of the lowest cost path in the binary tree is presented for defining the best sub-optimal solution of the problem in hand.

2.2.1 Definition of the Solution-Metric

In order to find the optimal solution, let us define a suitable cost function or *metric* that quantifies the closeness of every candidate/trial solution $\underline{C}_t$ to the optimal one,

$$\Psi\left\{\underline{C}_t\right\} = \frac{1}{M}\sum_{m=1}^{M} \alpha_m^2 \left[v_m - d_m\left\{\underline{C}_t\right\}\right]^2, \tag{2.2}$$

where v_m and d_m are reference and estimated parameters, respectively. The estimated parameters $d_m\left\{\underline{C}_t\right\}$ are defined as the weighted arithmetic mean of

the reference parameters v_m related to the array elements belonging to the same sub-array:

$$d_m\left\{\underline{C}_t\right\}\big|_q = \frac{\sum_{m=1}^{M}\alpha_m^2\delta_{c_m q}v_m}{\sum_{m=1}^{M}\alpha_m^2}\,. \tag{2.3}$$

As far as the reference parameters $\underline{V} = \{v_m;\ m = 1, ..., M\}$ and the sub-arrays weights $\underline{W} = \{w_q;\ q = 1, ..., Q\}$ are concerned, they are defined according to the *Gain Sorting* (GS) algorithm.

Concerning the GS technique, the reference parameters v_m are set to the *optimal gains*

$$v_m = \frac{\beta_m}{\alpha_m}, \qquad m = 1, \ldots, M, \tag{2.4}$$

while the sub-array weights are assumed to be equal to the *computed gains* d_m

$$w_q = \delta_{c_m q} d_m\left\{\underline{C}_t\right\}, \qquad q = 1, ..., Q,\ m = 1, \ldots, M. \tag{2.5}$$

2.2.2 Definition of the Solution-Tree

In general, the total number of sub-array configurations is equal to $T = Q^M$ since each of them might be expressed as a sequence of M digits in a Q-based notation system. Without any loss of information, such a number can be reduced by considering only the *admissible* (or *reliable*) solutions, i.e., grouping where there are no empty sub-arrays. Moreover, let us observe that if an *equivalence* relationship[2] among sub-array configurations holds true, it is convenient to consider just one sub-array configuration for each set (instead of the whole set), therefore obtaining a set of *non-redundant* solutions.

Now, let us sort the known reference parameters $\{v_m;\ m = 1, ..., M\}$ [computed according to either the GS (2.4)] for obtaining a ordered list $\underline{L} = \{l_m;\ m = 1, ..., M\}$, where $l_i \leq l_{i+1},\ i = 1, ..., M-1$, $l_1 = min_m\{v_m\}$, and $l_M = max_m\{v_m\}$. Since the cost function is minimized provided that elements belonging to each sub-array are consecutive elements of the ordered list $\underline{L}$ (see ***Appendix A*** for a detailed proof), the solution space can be further reduced to the so-called *essential* solution space $\Re^{(ess)}$ composed by *allowed* solutions. Consequently, the dimension T of the solution space turns out to be reduced from $T = Q^M$ up to $T^{(ess)} = \begin{pmatrix} M-1 \\ Q-1 \end{pmatrix}$ (see ***Appendix B*** for a detailed proof) and the essential solution space $\Re^{(ess)}$ can be formally represented by means of the non-complete binary tree depicted in Figure 2.2. In particular, each complete path in the tree codes an *allowed* sub-array configuration $\underline{C}_t^{(ess)} \in \Re^{(ess)}$ and the positive integer

[2] A sub-array configuration $\underline{C}_i$ is equivalent to the configuration $\underline{C}_j$ when it is possible to obtain the one from the other just using a different numbering for the same c_m coefficients. As an example, the sub-array configuration $\underline{C}_i = \{1, 2, 3, 3, 2, 3, 2, 1\}$ is equivalent to $\underline{C}_j = \{2, 3, 1, 1, 3, 1, 3, 2\}$.

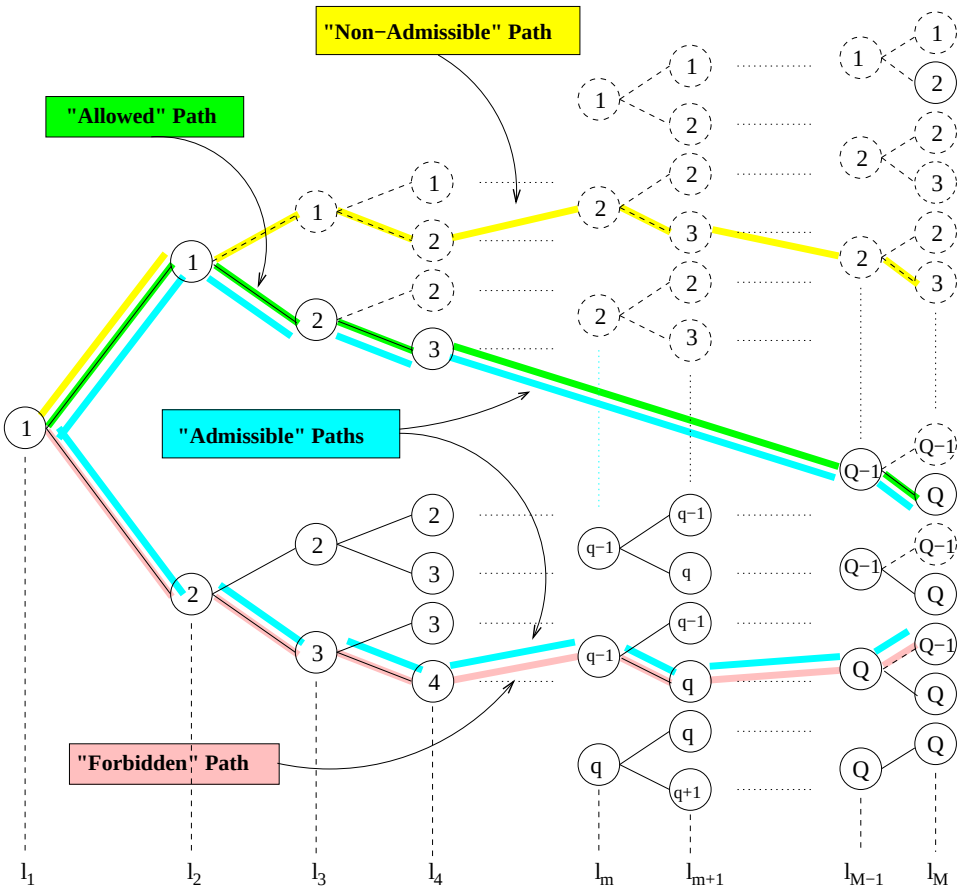


Figure 2.2: *Solution-Tree* structure representing the essential solution space $\Re^{(ess)}$.

q inside each node at the l_m-th level indicates that the array element identified by l_m is a member of the q-th sub-array. Thanks to this formulation, the original minimization problem (i.e., $\underline{C}_{opt} = arg\left\{min_{t=1,...,T}\left[\Psi\left(\underline{C}_t\right)\right]\right\}$) is recast as that of finding the optimal path in the solution tree.

2.2.3 Tree-Searching Procedure

Although the set of candidate solutions has been considerably reduced by limiting the solution space to the *essential* space, its dimension $T^{(ess)}$ becomes very large when $M \gg Q$ and an exhaustive searching would be computationally expensive. In order to overcome such a drawback, let us observe that only some elements of the list L are candidate to change their sub-array membership without violating the sorting condition of the *allowed* sub-array configurations, $\left\{\underline{C}_t^{(ess)};\, t = 1, ..., T^{(ess)}\right\}$ [see Eq. (B.1) - ***Appendix B***]. These elements, referred to as *border elements*, satisfy the following property: an array element related

to l_m is a *border element* if one of the elements whose list value is l_{m-1} or/and l_{m+1} belongs to a different sub-array. Therefore, the aggregation $\underline{C}_{opt} \in \Re^{(ess)}$ minimizing the cost function Ψ is found starting from an initial path randomly chosen among the set of paths in the solution tree and iteratively updating the candidate solution just modifying the membership of the border elements. More in detail, the iterative procedure (k being the iteration index) consists of the following steps.

- *Step* 0 - ***Initialization***. Initialize the iteration counter ($k = 0$) and the sequence index ($m = 0$). Randomly generate a trial path in the solution tree corresponding to a candidate sub-arrays configuration $\underline{C}^{(0)} \in \Re^{(ess)}$. Set the optimal path to $\underline{C}_{opt}^{(k)}\Big\rfloor_{k=0} = \underline{C}^{(0)}$.

- *Step* 1 - ***Cost Function Evaluation***. Compute the cost function value of the current candidate path $\underline{C}^{(k)}$ by means of (2.2), $\Psi^{(k)} = \Psi\left\{\underline{C}^{(k)}\right\}$. Compare the cost of the aggregation $\underline{C}^{(k)}$ to the best cost function value attained at any iteration up to the current one, $\Psi_{opt}^{(k-1)} = min_{h=1,\ldots,k-1}\left(\Psi\left\{\underline{C}^{(h)}\right\}\right)$ and update the optimal trial solution $\underline{C}_{opt}^{(k)} = \underline{C}^{(k)}$ if $\Psi\left\{\underline{C}^{(k)}\right\} < \Psi\left\{\underline{C}_{opt}^{(k-1)}\right\}$.

- *Step* 2 - ***Convergence Check***. If the *termination criterion*, based on a maximum number of iterations K or on a stationary condition for the fitness value (i.e., $\frac{\left|K_{window}\Psi_{opt}^{(k-1)} - \sum_{j=1}^{K_{window}}\Psi_{opt}^{(j)}\right|}{\Psi_{opt}^{(k)}} \leq \eta$, K_{window} and η being a fixed number of iterations and a fixed numerical threshold, respectively), is satisfied then set $\underline{C}_{opt} = \underline{C}_{opt}^{(k)}$ and stop the minimization process. Otherwise, go to Step 3.

- *Step* 3 - ***Iteration Updating***. Update the iteration index ($k \leftarrow k+1$) and reset the sequence index ($m = 0$).

- *Step* 4 - ***Sequence Updating***. Update the sequence index ($m \leftarrow m+1$). If $m > M$ then go to Step 3 else go to Step 5.

- *Step* 5 - ***Aggregation Updating***. If the array element related to $l_m^{(k)}$ is a border element belonging to the q-th sub-array then define a new grouping $\underline{C}^{(k,m)}$ by aggregating such an element to the $(q-1)$-th sub-array [if the array element corresponding to $l_{m-1}^{(k)}$ is a member of the $(q-1)$-th sub-array] or to the $(q+1)$-th sub-array [if the array element corresponding to $l_{m+1}^{(k)}$ is a member of the $(q+1)$-th sub-array]. If $\Psi^{(k,m)} = \Psi\left\{\underline{C}^{(k,m)}\right\} < \Psi\left\{\underline{C}^{(k)}\right\}$ then set $\underline{C}^{(k)} = \underline{C}^{(k,m)}$ and go to Step 1. Otherwise, go to Step 4.

2.3 Numerical Validation

In order to assess the effectiveness of the proposed method, an exhaustive set of numerical experiments has been performed and some representative results will be shown in the following.

For a quantitative evaluation, a set of beam pattern indexes has been defined and computed. More in detail, (a) the *pattern matching* Δ that quantifies the *distance* between the synthesized sub-optimal pattern and the optimal one

$$\Delta = \frac{\int_0^{\pi} \left| |AF(\psi)|_n^{opt} - |AF(\psi)|_n^{rec} \right| d\psi}{\int_0^{\pi} |AF(\psi)|_n^{opt} \, d\psi}, \tag{2.6}$$

where $\psi = (2\pi d/\lambda)\, sin\theta$, $\theta \in [0, \pi/2]$, (λ and d being the free-space wavelength and the inter-element spacing, respectively), $|AF(\psi)|_n^{opt}$ and $|AF(\psi)|_n^{rec}$ are the normalized optimal and generated array patterns, respectively; (b) the main lobes *beamwidth* B_W and (c) the *power slope* P_{slo} that give some indications on the slope on the boresight direction

$$P_{slo} = 2 \times \left[\max_{\psi} \left(|AF(\psi)|_n \right) \times \psi_{max} - \int_0^{\psi_{max}} |AF(\psi)|_n \, d\psi \right], \tag{2.7}$$

ψ_{max} being the angular position of the maximum in the array pattern; (d) the *sidelobes power* P_{sll}

$$P_{sll} = \int_{\psi_1}^{\pi} |AF(\psi)|_n \, d\psi, \tag{2.8}$$

where ψ_1 is the angular position of the *first null* in the difference beam pattern.

The remaining of this section is organized as follows. Firstly, some experiments aimed at showing the asymptotic behaviour of the proposed solution are presented (Sect. 2.3.1) and a comparative study is carried out (Sect. 2.3.2). Furthermore, some experiments devoted at showing the potentialities of the proposed solution in dealing with large arrays are discussed in Sect. 2.3.3. Finally, the computational issues are analyzed (Sect. 2.3.4).

2.3.1 Asymptotic Behavior Analysis

In order to assess that increasing the number of sub-arrays Q the synthesized difference patterns get closer and closer to the optimal one, let us consider a linear array of $N = 2 \times M = 20$ elements characterized by a $d = \frac{\lambda}{2}$ inter-element spacing. The optimal sum pattern excitations, $\{\alpha_m,\, m = 1, ..., M\}$, have been fixed to that of the linear Villeneuve pattern [17] with $\overline{n} = 4$ and $25\, dB$ sidelobe ratio (Fig. 2.3 - *Villeneuve*, 1984), while the optimal difference weights $\{\beta_m,\, m = 1, ..., M\}$, have been chosen equal to those of a Zolotarev difference pattern [9] with a sidelobe level $SLL = -30\, dB$ (Fig. 2.3 - *McNamara*, 1993). Then, Q has been varied between 2 and M and the GS technique has been applied. For sake of

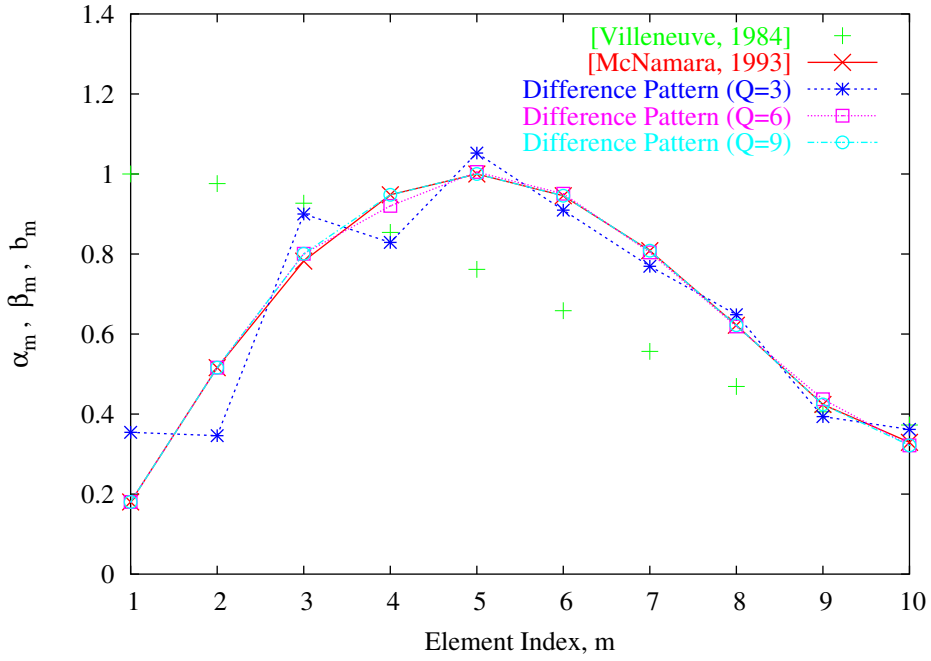


Figure 2.3: *Asymptotic Behavior* ($M = 10$, $d = \frac{\lambda}{2}$) - Sum $\{\alpha_m;\ m = 1, ..., M\}$ and difference $\{\beta_m;\ m = 1, ..., M\}$ optimal excitations. Compromise difference coefficients $\{b_m;\ m = 1, ..., M\}$ for different values of Q.

space, selected results concerned with $Q = 3$, $Q = 6$, and $Q = 9$ are reported in terms of difference excitations. As expected, the coefficients obtained with the GS converge to the optimal ones and, starting from $Q = 6$, the differences between generated and reference difference patterns turn out to be smaller and smaller.

2.3.2 Comparative Assessment

For comparison purposes and in the framework of synthesis techniques aimed at determining the *best compromise* difference pattern as close as possible to the optimal one, let us consider the EMM by *McNamara* [8] as reference[3]. As far as the test cases are concerned, the same benchmark investigated in [8] has been taken into account. The array geometry and the optimal sum excitations was as in Sect. 2.3.1, while the optimal difference excitation vector $\underline{B}^{opt}$ has been chosen for generating a modified Zolotarev difference pattern with $\overline{n} = 4$, $\varepsilon = 3$ and a sidelobe ratio of $25\,dB$ [9].

[3]No comparison with optimization-based procedures (i.e., [10][11][12][13][14]) have been reported since they are aimed at minimizing a pattern parameter (e.g., the SLL) and not at better matching an optimal difference pattern.

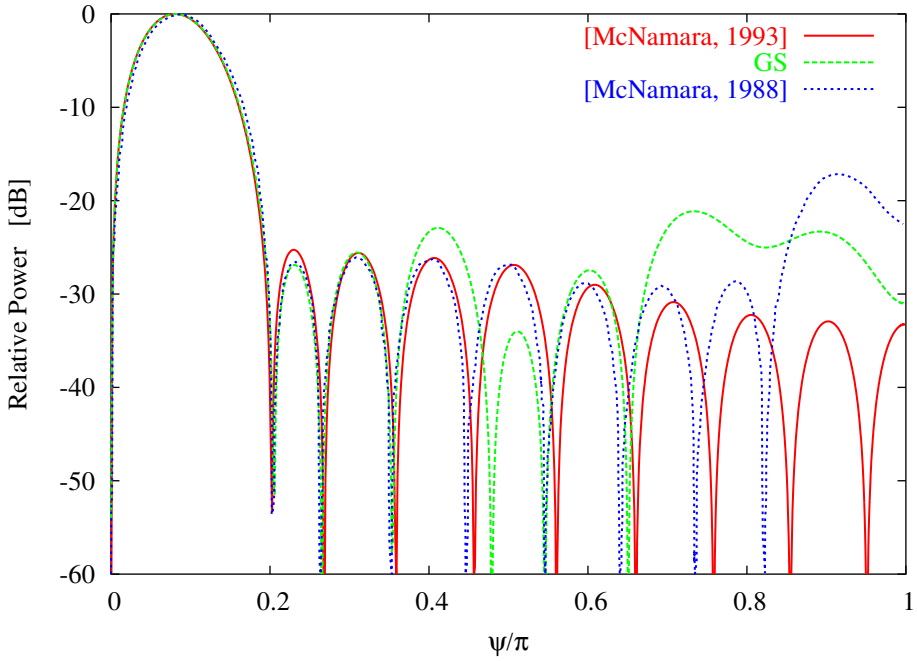


Figure 2.4: *Uniform sub-arraying* ($M = 10$, $d = \frac{\lambda}{2}$, $Q = 5$) - Reference optimum and normalized difference patterns obtained by means of the EMM and the GS.

The first test case deals with a uniform sub-arraying over the antenna with $Q = 5$. The values of the sub-arrays weights optimized with the GS are $\underline{W}^{GS} = \{0.2951,\ 0.8847,\ 1.1885,\ 1.3994,\ 1.4878\}$. Moreover, the synthesized difference pattern is shown in Figure 2.4, while the computed beam-pattern indexes are reported in Table 2.1. The advantages on the use of the tree-based approaches are evident, as confirmed by the values of both the SLL ($4\,dB$ below the level achieved by the EMM, $SLL^{EMM} = -17.00\,dB$ vs. $SLL^{GS} = -21.00$) and the *pattern matching* index ($\frac{\Delta^{EMM}}{\Delta^{GS}} \simeq 1.5$ - Tab. 2.1). Moreover, it is worth noting that, thanks to the structure of the solution tree, the dimension of the *essential* space reduces to $T^{(ess)} = 1$ (since l_1 and l_2 belong to the first sub-array, l_3 and l_4 to the second one, and so on), thus allowing a significant saving of

Approach	P_{slo}	B_W	P_{sll}	$max\{SLL\}$	Δ
EMM [8]	0.1970	0.3610	0.1038	-17.00	0.4015
GS	0.1811	0.3784	0.1082	-21.10	0.2633
Optimal [9]	0.1802	0.3735	0.0598	-25.00	–

Table 2.1: *Uniform sub-arraying* ($M = 10$, $d = \frac{\lambda}{2}$, $Q = 5$) - Beam pattern indexes.

	$Q=3$		$Q=5$		
	EMM [8]	*GS*	*EMM* [8]	*GS*	*Optimal* [9]
P_{slo}	0.2117	0.1800	0.2000	0.1806	0.1802
B_W	0.3745	0.3735	0.3854	0.3735	0.3735
P_{sll}	0.1798	0.1054	0.0950	0.0823	0.0598
$max\{SLL\}$	-14.70	-18.63	-23.40	-23.00	-25.00
Δ	0.5438	0.4073	0.2562	0.1571	$-$

Table 2.2: *Non-uniform sub-arraying* ($M=10$, $d=\frac{\lambda}{2}$, $Q=3,\,5$) - Beam pattern indexes.

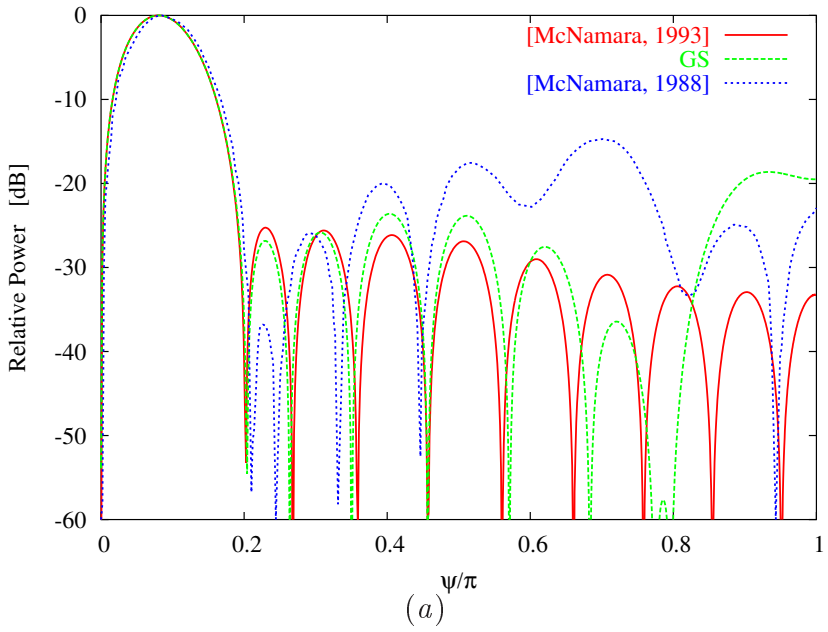
computational resources. As a matter of fact, the *EMM* requires the solution of an overdetermined system of linear equations in correspondence with any possible uniform grouping [8], i.e., a number of $T=945$ evaluations.

Second and third test cases consider non-uniform sub-arraying. The former configuration is an example of the limited number of sub-arrays ($Q=3$) that might be used with a small monopulse antenna. The latter has the same number of sub-arrays as that of the first configuration ($Q=5$). The tree-based algorithms have been applied and the following sub-array configurations have been determined. In particular, the grouping $\underline{C}_{opt}^{GS}=\{1,\,2,\,3,\,3,\,4,\,5,\,5,\,5,\,4,\,3\}$ has been synthesized when $Q=5$, while $\underline{C}_{opt}^{GS}=\{1,\,1,\,2,\,2,\,3,\,3,\,3,\,3,\,3,\,2\}$ has been obtained for $Q=3$. The obtained beam patterns are shown in Fig. 2.5 and the corresponding values of the pattern indexes are reported in Tab. 2.2. As it can be noticed, the *GS* improves the performances of the *EMM* in matching the optimal difference pattern as pointed out by the behavior of the *global* matching index Δ ($\frac{\Delta^{EMM}}{\Delta^{GS}}\Big|_{Q=3}=1.33$ and $\frac{\Delta^{EMM}}{\Delta^{GS}}\Big|_{Q=5}=1.63$). Concerning the smaller configuration, it is further confirmed (as already pointed out in Section 2.3.1) the flexibility and reliability of the *GS* algorithm in dealing also with complex cases where a limited number of sub-arrays is taken into account. As a matter of fact, for $Q=3$ the solution of the *GS* has a sidelobe ratio of $SLL=18.63\,dB$ and a main lobe very close to the optimal one, i.e., $B_w^{GS}=B_w^{opt}=0.3735$ and $P_{slo}^{GS}=0.1800$ vs. $P_{slo}^{opt}=0.1802$.

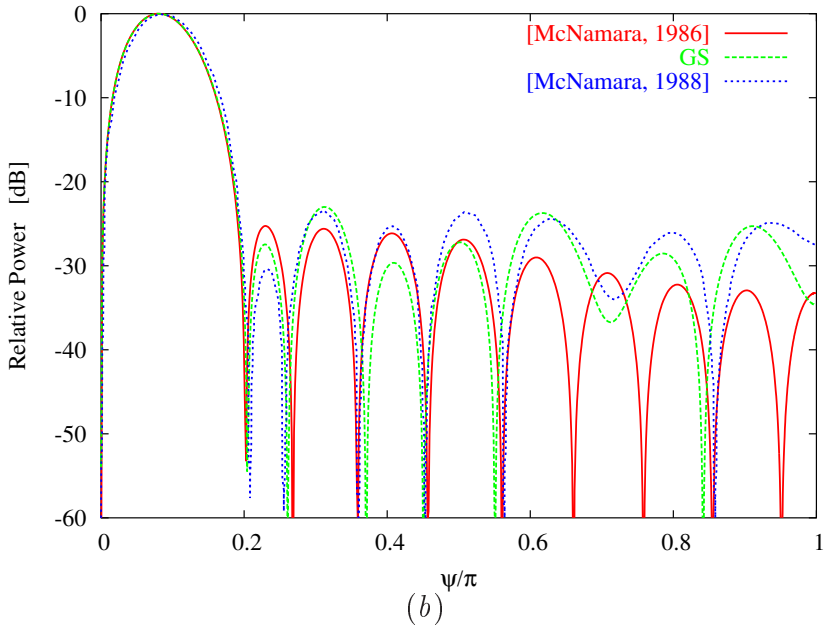
2.3.3 Large Arrays Analysis

This section is aimed at analyzing the performances of the proposed tree-based techniques when dealing with large arrays. As far as the optimal setup is concerned, sum $\{\alpha_m,\,m=1,...,M\}$ and difference $\{\beta_m,\,m=1,...,M\}$ optimal excitations have been chosen to generate a Dolph-Chebyshev pattern [19] with $SLL=-25\,dB$ and a Zolotarev pattern [9] with $SLL=-30\,dB$, respectively.

As a first experiment, a linear array of $N=200$ elements with $\lambda/2$ spacing



(a)



(b)

Figure 2.5: *Non-uniform sub-arraying* ($M = 10$, $d = \frac{\lambda}{2}$) - Reference optimum and normalized difference patterns obtained by means of the EMM, and the GS when (a) $Q = 3$ and (b) $Q = 5$.

	GS	$Optimal\ Difference$ [9]
P_{slo}	0.0066	0.0066
B_W	0.0148	0.0151
P_{sll}	0.0868	0.0824
$max\{SLL\}$	-18.00	-30.00
Δ	0.2921	$-$

Table 2.3: *Large Arrays* ($M = 250$, $d = \frac{\lambda}{2}$, $Q = 4$) - Beam pattern indexes.

has been used by considering various sub-arraying configurations. Figure 2.6 shows the optimal difference pattern (i.e., the synthesis target) and the patterns obtained when $Q = 4$ and $Q = 6$ by using the GS. For completeness, the values of the synthesized difference excitations are displayed in Figure 2.7. The GS algorithm satisfactorily approximates the optimal main lobe characteristics in terms of both B_W and P_{slo}, and the solution presents a sidelobe ratio close to the reference one ($SLL^{GS}\big\rfloor_{Q=4} = -21.90$ and $SLL^{GS}\big\rfloor_{Q=6} = -25.13$). The last test case (and second experiment dealing with large structures) is concerned with a linear array of $N = 2 \times M = 500$ elements ($d = \lambda/2$). As a representative example, the case of $Q = 4$ is reported and analyzed (Tab. 2.3). The arising beam patterns allow one to drawn similar conclusions to those from the previous scenario, since once again the effectiveness of the GS technique in dealing with a limited number of sub-arrays is pointed out. As a matter of fact, it is worth noting that unlike tree-based procedures the EMM is not reliable in dealing with large arrays since it requires the numerical processing of overdetermined linear systems, whose ill-conditioning get worse when the ratio $\frac{M}{Q}$ grows.

In order to evaluate the performance of the tree-based method versus the array dimension, N has been varied from 20 (small/medium arrays, i.e. $M < 50$) up to 500 (large arrays, i.e. $M \geq 50$) and different array partitions ($Q \in [3, 10]$) have been considered. The plot of Δ versus M for different values of Q is shown in Figure 2.8. As it can be observed, for a fixed number Q of sub-arrays, the distance between the optimal difference pattern and the compromise one does not significantly vary as the number of elements M increases ($M > 50$) ranging from $\Delta \cong 0.15$ ($Q = 10$) up to $\Delta \cong 0.36$ ($Q = 3$). Moreover, as expected, for each array aperture (i.e., $M = cost$), the synthesized difference patterns get closer and closer to the optimal one when the value of Q grows ($Q \to M$).

2.3.4 Computational Issues

Now, let us analyze the computational costs of the tree-based approaches, providing a comparison with the EMM, as well. Towards this end, let us firstly consider the dependence of the dimension of the solution space on the number of elements of the array M. As a representative case, let us analyze the behavior

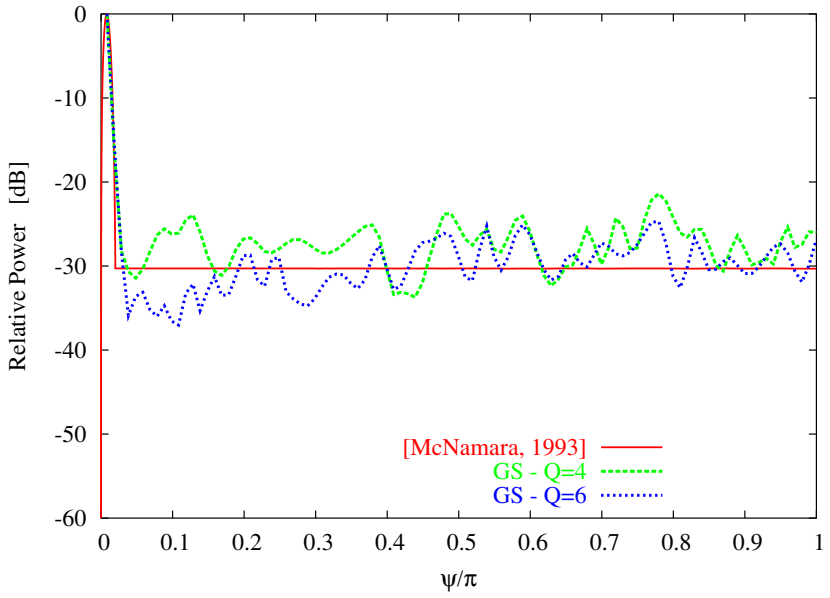


Figure 2.6: *Large Arrays* ($M = 100$, $d = \frac{\lambda}{2}$) - Reference optimum and normalized difference patterns obtained by means of the GS technique when $Q = 4$ and $Q = 6$.

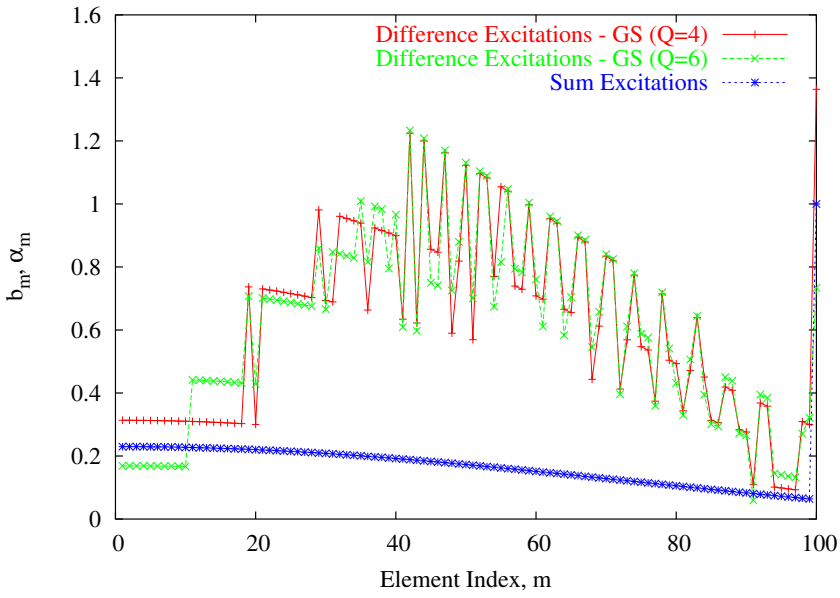


Figure 2.7: *Large Arrays* ($M = 100$, $d = \frac{\lambda}{2}$) - Difference excitations determined by the tree-based techniques when $Q = 4$ and $Q = 6$.

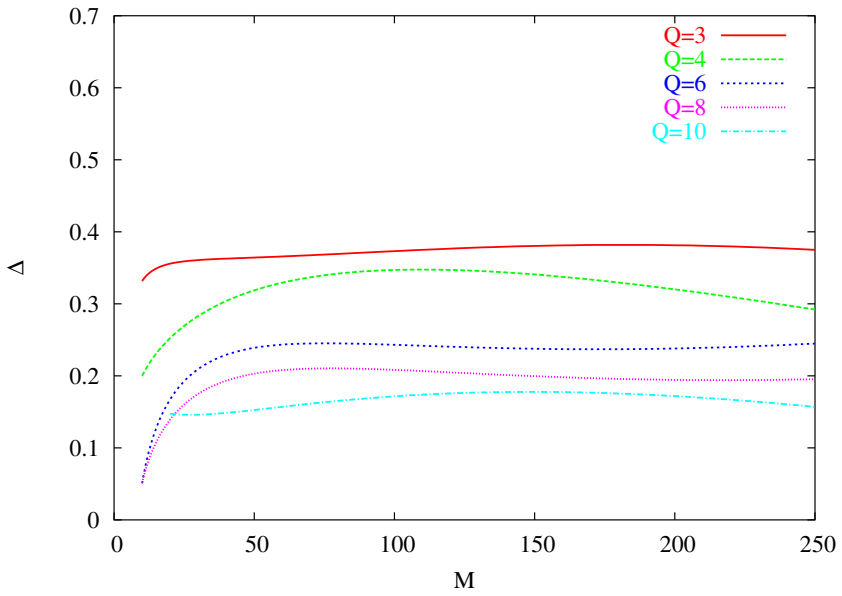


Figure 2.8: *Large Arrays* ($d = \frac{\lambda}{2}$) - Behavior of Δ versus M for various values of Q.

of T and $T^{(ess)}$ when $Q = 3$ ($K = 100$ and $\eta = 10^{-3}$) (Fig. 2.9). As it can be observed, the dimension of the solution space T of the EMM grows exponentially with M, while, as expected [see ***Appendix A***], $T^{(ess)}$ shows a polynomial behavior. Obviously, the same behavior holds true also for different values of Q (Fig. 2.9).

On the other hand, the computational effectiveness of the *Tree-Searching* procedure in sampling the solution space is further pointed out from the evaluation of the CPU-time, t (on a $3\,GHz$ Pentium 4 and $512\,MB$ of RAM), needed for reaching the convergence (Fig. 2.10). As a matter of fact, $max_Q\{t_Q\} = 70\ [sec]$ ($k_{opt} = 90$) in correspondence with the largest array ($M = 250$), while $max_Q\{t_Q\} = 12.8\ [sec]$ ($k_{opt} = 8$) and $max_Q\{t_Q\} = 2.3\ [sec]$ ($k_{opt} = 4$) when $M = 100$ and $M = 50$, respectively.

2.4 Discussions

The methodological novelties of the proposed approach lie in the appropriate definition of the solution space, the innovative formulation of the problem in terms of a search inside a non complete binary tree and the possibility of applying a fast resolution algorithm. All these improvements allow the proposed approach to deal with the synthesis of large arrays in an effective and reliable way.

As confirmed in the comparative assessment, because of the favorable trade-

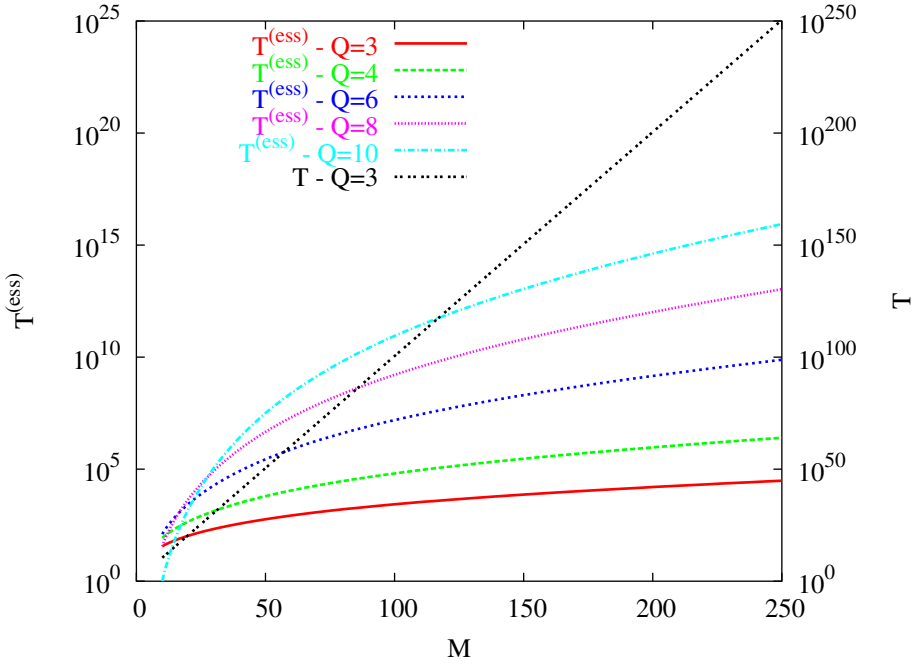


Figure 2.9: *Computational Analysis* - Behavior of T versus M when the tree-based searching is applied [$T = T^{(ess)}$].

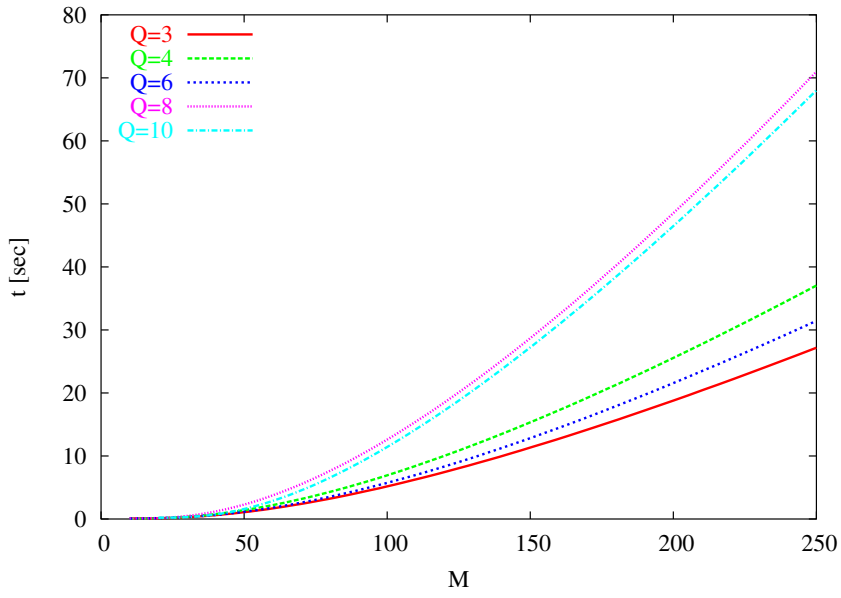


Figure 2.10: *Computational Analysis* - Behavior of t versus M for different values of Q (GS Approach).

off between complexity/costs and effectiveness, the proposed tree-based strategy seems a promising tool to be further analyzed and extended to other geometries and synthesis problems. Accordingly, some of the possible extensions will be considered in the next chapters of this work.

Chapter 3

The Iterative Matching Approach

In this chapter, the optimal excitations matching method presented in Chapter 2 is integrated in an iterative procedure ensuring, at the same time, the optimization of the sidelobe level (or other beam pattern features) for the compromise difference pattern. The flexibility of such an approach allows one to synthesize various difference patterns characterized by different trade-off between angular resolution and noise/interferences rejection in order to match the user-defined requirements. On the other hand, thanks to its computational efficiency, synthesis problems concerned with large arrays are easily managed, as well. An exhaustive numerical validation assesses the reliability and accuracy of the method pointing out the improvements upon state-of-the-art sub-arraying techniques.

3.1 Introduction

The design of monopulse radar systems [2][3] requires the synthesis of both a sum pattern and a difference pattern, which satisfy some specifications such as narrow beamwidth, low side-lobe-level (SLL), and high directivity. When the sub-arraying strategy is considered, according to the guidelines of [8][10][12][11][14], the sum pattern is fixed to the optimal one, while difference excitations are obtained from the sum coefficients by properly grouping the array elements and by weighting each sub-array in order to satisfy the user-defined constraints. In such a context, two different methodological approaches might be recognized. The former (indicated in the following as "*optimal matching*") is aimed at determining the "*best compromise*" difference pattern, which is as close as possible to the optimum in the Dolph-Chebyshev sense [9] (i.e., narrowest first null beamwidth and largest normalized difference slope on the boresight for a specified sidelobe level), as considered in Chapter 2. The other, denoted as "*feature optimization*", where the beam pattern parameters (usually, the SLL [10]-[11] or the directivity [20]) are controlled by including them in a cost function to minimize according to a global optimization stochastic procedure.

Concerning the "*optimal matching*" techniques, since the the "*Excitation Matching*" method (EMM) proposed by *McNamara* in [8] does not allow the control of the beam pattern SLL, hence a constrained version of the method has been also introduced ([8], Sect. 5) in order to reduce the grating lobes effects and lead to sub-optimal difference patterns with a suitable compromise between SLL, beamwidth, and slope on boresight. Unfortunately, when the ratio between array elements and number of sub-arrays gets larger, the EMM is not always reliable/efficient because of the ill-conditioning of the matrix system as well as the large computational costs of the arising exhaustive evaluation process.

As far as the "*feature optimization*" class of sub-arraying methods is concerned, *Ares et al.* considered in [10] the application of a simulated annealing (SA) algorithm for defining the optimal sub-array weights (i.e., aimed at obtaining a difference pattern that satisfies a fixed constraint on the SLL) starting from an assigned sub-array configuration. On the other hand, taking advantage of the problem convexity with respect to the weights of the subarrays and following the same line of the reasoning as in [21], a two-step hybrid optimization strategy has been proposed in [13][14]. By optimizing at the same time both partition functions (i.e., those functions that define the membership of the array elements to each sub-array) and the sub-array coefficients, *Lopez et al.* [12] proposed a Genetic Algorithm (GA) based technique. In a similar fashion, a Differential Evolution (DE) algorithm has been used in [11].

Although the optimization of elements membership and sub-array weights significantly improved the performance of sum-difference optimization methodologies, some drawbacks still remain. As a matter of fact, such techniques are usually time-consuming especially when dealing with large arrays since the di-

mension of the solution space significantly enlarges. Moreover, "*feature optimization*" approaches are usually formulated in terms of single-objective problems and the control of multiple features of the beam pattern (e.g., SLL, beamwidth, difference slope on boresight) would require the use of customized and complex multi-objective strategies.

In the framework of optimal matching techniques, the present contribution is aimed at proposing a new approach for synthesizing best compromise patterns with SLL control. Towards this end, following the guidelines of the EMM, the proposed approach determines the difference solution close to the optimal Dolph-Chebyshev pattern through the search of the minimum cost-path in the non-complete binary tree of the possible aggregations by satisfying the SLL constraints through an iterative procedure (unlike global optimization methods that directly define a SLL penalty term in the cost function [10]-[11]).

The remaining of the chapter is organized as follows. The proposed synthesis procedure is described in detail Section 3.2. Section 3.3 deals with an exhaustive numerical validation aimed at assessing the effectiveness of the proposed technique and at providing a comparison with state-of-the-art solutions. Some final remarks are drawn in Section 3.4.

3.2 Mathematical Formulation

Let us consider a linear uniform array of $N = 2M$ elements and let us assume that the sum and difference patterns are obtained through a symmetric, $\underline{A} = \{a_m = a_{-m};\ m = 1, ..., M\}$, and an anti-symmetric, $\underline{B} = \{b_m = -b_{-m};\ m = 1, ..., M\}$, real excitations set, respectively. Thanks to these symmetry properties, only one half of the array elements is considered.

According to the guidelines of sub-arraying techniques, the sum pattern is obtained by fixing the sum excitations to the ideal ones, $\underline{A}^{ideal} = \{\alpha_m;\ m = 1, ..., M\}$ [19][5][17], while the difference excitations set is synthesized starting from the sum mode as follows

$$b_m = \sum_{q=1}^{Q} \alpha_m \left(\delta_{c_m q} w_q\right);\ m = 1, ..., M, \tag{3.1}$$

where Q is the number of sub-arrays, w_q is the weight associated to the q-th sub-array in the difference feed network, and $\delta_{c_m q}$ is the Kronecker delta whose value is determined according to the sub-array membership of each element of the array ($\delta_{c_m q} = 1$ if $c_m = q$, $\delta_{c_m q} = 0$ otherwise, $c_m \in [1, Q]$ being the sub-array index of the m-th array element).

In order to obtain the best compromise difference excitations (i.e., a set of excitations giving a pattern as close as possible to the ideal one in the Dolph-Chebyshev sense that satisfies at the same time a constraint on the SLL), an innovative adaptive searching technique, indicated as *Iterative Contiguous Partition Method* ($ICPM$), is applied. It consists of an inner loop aimed at ensuring

the closeness of the trial solution to a "*reference*" ideal pattern through the technique proposed in Chapter 2 and by an outer loop devoted at satisfying the requirements on the SLL (or another beam pattern feature).

With reference to Fig. 3.1 , the main steps of the iterative procedure are described in the following:

- *Step* 0 - ***Initialization***. The external iteration index is initialized ($e = 0$), the optimal sum excitations $\underline{A}^{ideal} = \{\alpha_m;\ m = 1, ..., M\}$ are computed [19][5][17], and the user-desired sidelobe level threshold is set, SLL_d;

- *Step* 1 - ***Reference Difference Pattern Selection***. At the first iteration ($e = 1$), an optimal - in the Dolph-Chebyshev sense - difference excitations set $\underline{B}_{ref}^{(e)} = \left\{\beta_m^{(e)};\ m = 1, ..., M\right\}$ that generates a beam pattern with a sidelobe level $SLL_{ref}^{(e)} = SLL_d$ is computed as in [9] and assumed as reference in the inner loop. Then, for each element of the array, an identification parameter is evaluated according to the *Gain Sorting* (GS) algorithm

$$v_m^{(e)} = \frac{\beta_m^{(e)}}{\alpha_m},\ m = 1, ..., M. \quad [Optimal\, Gain] \tag{3.2}$$

  The identification indexes $\left\{v_m^{(e)};\ m = 1, \ldots, M\right\}$ are ordered in a sorted list $\underline{L} = \{l_m;\ m = 1, \ldots, M\}$ (i.e., an ensemble where $l_k \leq l_{k+1}$, $k = 1, \ldots, M-1$, $l_1 = \min_m \left\{v_m^{(e)}\right\}$, and $l_M = \max_m \left\{v_m^{(e)}\right\}$);

- *Step* 2 - ***Computation of the Compromise Solution***. With reference to the e-th target pattern, the approximation algorithm based on the *Contiguous Partition* technique is run until a suitable "*termination criterion*" is satisfied. Accordingly, the following steps are performed:

  - *Step* 2.*a* - ***Solution Initialization***. The internal iteration counter is initialized $[i(e) = 0]$ and a starting trial grouping $\underline{C}^{i(e)} = \left\{c_m^{i(e)};\ m = 1, \ldots, M\right\}$, corresponding to a *Contiguous Partition*[(1)1] of $\underline{L}$ in Q subsets $\underline{P}_Q^{i(e)} = \left\{\underline{L}_q^{i(e)};\ q = 1, \ldots Q\right\}$, is randomly generated and assumed as the optimal grouping $\underline{C}_{opt}^{i(e)} = \underline{C}^{i(e)}$. Successively, the sub-array weights

[1] (1) With reference to [18], it can be easily shown that, once the parameters $v_m^{(e)}$ have been ordered in the sorted list $\underline{L} = \{l_m;\ m = 1, ..., M\}$, the grouping minimizing the cost function (3.5) corresponds to a *Contiguous Partition*. A grouping of array elements is a *Contiguous Partition* if the generic m_j-th array element belongs to the q-th sub-array only when two elements, namely the m_i-th element and the m_n-th one, belong to the same sub-array and the condition $v_i^{(e)} < v_j^{(e)} < v_n^{(e)}$ holds true.

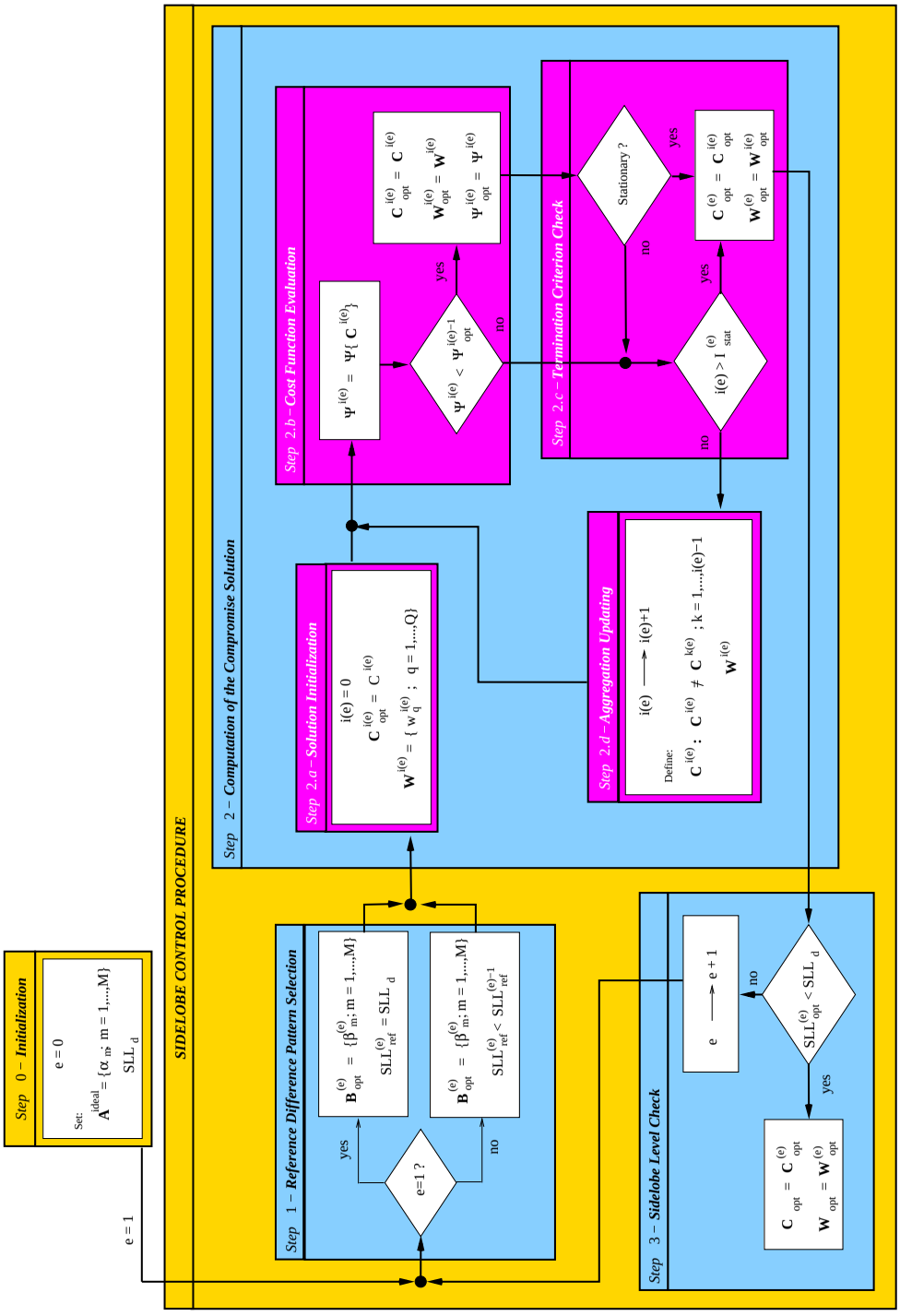


Figure 3.1: Flow chart of the *Iterative Contiguous Partition Method*.

$\underline{W}^{i(e)} = \left\{ w_q^{i(e)};\, q = 1, \ldots, Q \right\}$ are analytically computed as

$$w_q^{i(e)} = \sum_{m=1}^{M} \delta_{c_m q} d_m \left(\underline{C}^{i(e)} \right),\ q = 1, ..., Q \quad [Estimated\, Gain] \tag{3.3}$$

$d_m \left(\underline{C}^{i(e)} \right)$ being an estimate of the identification parameter $v_m^{(e)}$ given by

$$d_m \left(\underline{C}^{i(e)} \right) = \frac{\sum_{s=1}^{M} \alpha_s^2 \delta_{c_s c_m} v_s^{(e)}}{\sum_{s=1}^{M} \alpha_s^2 \delta_{c_s c_m}}, \qquad m = 1, ..., M; \tag{3.4}$$

– *Step* 2.*b* - ***Cost Function Evaluation***. The closeness to the target pattern of the current candidate solution $\underline{B}^{i(e)}$ (or in an equivalent fashion, the couple of coefficients $\underline{C}^{i(e)}$ and $\underline{W}^{i(e)}$) is quantified through the following cost function

$$\Psi \left\{ \underline{C}^{i(e)} \right\} = \frac{1}{M} \sum_{m=1}^{M} \alpha_m^2 \left[v_m^{(e)} - d_m \left(\underline{C}^{i(e)} \right) \right]^2. \tag{3.5}$$

The cost function value $\Psi^{i(e)} = \Psi \left\{ \underline{C}^{i(e)} \right\}$ is compared to the best value attained up till now, $\Psi \left\{ \underline{C}_{opt}^{i(e)-1} \right\} = \min_{h(e)=1,\ldots,i(e)-1} \left[\Psi^{h(e)} \right]$, and if $\Psi \left\{ \underline{C}^{i(e)} \right\} < \Psi \left\{ \underline{C}_{opt}^{i(e)-1} \right\}$, then the optimal trial solution is updated, $\underline{B}_{opt}^{i(e)} = \underline{B}^{i(e)}$, $\underline{C}_{opt}^{i(e)} = \underline{C}^{i(e)}$, and $\underline{W}_{opt}^{i(e)} = \underline{W}^{i(e)}$ as well as the optimal cost function value, $\Psi_{opt}^{i(e)} = \Psi^{i(e)}$;

– *Step* 2.*c* - ***Termination Criterion Check***. If a maximum number of iterations I is reached or a stationary condition $[i(e) = I_{stat}^{(e)}]$ for the cost function value,

$$\frac{\left| K_{window} \Psi_{opt}^{i(e)-1} - \sum_{t=1}^{I_{window}} \Psi_{opt}^{t(e)} \right|}{\Psi_{opt}^{i(e)}} \leq \eta\ , \tag{3.6}$$

holds true (I_{window} and η being a fixed number of iterations and a fixed numerical threshold, respectively), then the inner loop is stopped and the following setting is assumed: $\underline{C}_{opt}^{(e)} = \underline{C}_{opt}^{i(e)}$, $\underline{W}_{opt}^{(e)} = \underline{W}_{opt}^{i(e)}$ (i.e., $\underline{B}_{opt}^{(e)} = \underline{B}^{i(e)}$), and $\Psi_{opt}^{(e)} = \Psi_{opt}^{i(e)}$. The procedure goes to Step 3. Otherwise, the Step 2.*d* is performed;

– *Step* 2.*d* - ***Aggregation Updating***. The inner index is updated $[i(e) \leftarrow i(e)+1]$ and a new grouping vector $\underline{C}^{i(e)}$ is defined. More in detail, a new contiguous partition $\underline{P}_Q^{i(e)}$ is derived from the previous one

$\underline{P}_Q^{i(e)-1}$ just modifying the sub-array memberships of the *"Border Elements"* defined as follows $l_m \in \underline{L}_t^{i(e)} \wedge \left\{ \left(l_{m-1} \in \underline{L}_{t-1}^{i(e)} \right) \vee \left(l_{m+1} \in \underline{L}_{t+1}^{i(e)} \right) \right\}$, $t \in [1; Q]$. The corresponding sub-array weights $\underline{W}^{i(e)}$ are then analytically computed as in (3.3). The procedure goes to Step 2.*b*;

- *Step* 3 - ***Side-Lobe-Level Check***. The descriptive parameters of the beam pattern generated by the coefficients $\underline{B}_{opt}^{(e)}$ are computed as well as the SLL, $SLL_{opt}^{(e)} = SLL\left\{\underline{B}_{opt}^{(e)}\right\}$. If $SLL_{opt}^{(e)} \leq SLL_d$ and the "*degree of closeness*" to the reference pattern is satisfactory (e.g., some constraints on the beamwidth/directivity are satisfied), then the whole process ends and the final solution is: $\underline{C}_{opt} = \underline{C}_{opt}^{(e)}$, $\underline{W}_{opt} = \underline{W}_{opt}^{(e)}$ (i.e., $\underline{B}_{opt} = \underline{B}_{opt}^{(e)}$), $\Psi_{opt} = \Psi_{opt}^{(e)}$. Otherwise, the outer iteration index is updated ($e \leftarrow e + 1$) and another reference pattern that satisfies the condition $SLL_{ref}^{(e)} < SLL_{ref}^{(e-1)}$ is chosen. Then, the procedure restarts from Step 1 until $e = E$, E being a fixed number of outer-loop iterations.

It is worth noting that the *Contiguous Partition* technique applied in the inner loop allows a non-negligible saving of computational resources as pointed out in Section 3.3 by means of some numerical experiments as well as in Section 2.3.4.

3.3 Numerical Results

In this section, representative results from selected test cases are reported for assessing the effectiveness of the $ICPM$ in providing a suitable trade-off between desired SLL, directivity, and beamwidth (Sect. 3.3.1) as well as in dealing with smaller (Sect. 3.3.2) and larger arrays (Sect. 3.3.3). Comparisons with state-of-the-art synthesis techniques are presented (Sects. 3.3.2-3.3.3), as well.

In order to quantify the optimality and accuracy of the obtained solutions, the quantitative indexes introduced in Section 2.3 are considered. Moreover, concerning the computational costs, the total number of inner iterations, $I_{tot} = \sum_{e=1}^{E} I_{stat}^{(e)}$, the CPU-time needed for reaching the final solution, T, and the total number of possible sub-array configurations, U, are analyzed.

3.3.1 $ICPM$ Performance Analysis

This section is aimed at analyzing the behavior of the iterative SLL control procedure in providing a suitable trade-off between SLL, directivity, and beamwidth. Towards this end, a linear configuration of $N = 2 \times M = 20$ elements with $\lambda/2$ inter-element spacing is chosen and the sum excitations $\underline{A}^{ideal}$ have been set to those of the linear Villeneuve pattern [17] with $\overline{n} = 4$ and $25\,dB$ sidelobe ratio. Then, for fixed values of Q ($Q = 2, 4, 7$), the $ICPM$ has been applied by setting the sidelobe threshold to $SLL_d = -25\,dB$ and requiring a main lobe

width smaller than $Bw^{ref} = 6.0^o$. The adaptive searching procedure has been carried out by considering a succession of different reference excitation sets $\underline{B}_{ref}^{(e)}$, $e = 1, ., 3$, [9] with $SLL_{ref}^{(1)} = -25\,dB$, $SLL_{ref}^{(2)} = -30\,dB$, and $SLL_{ref}^{(3)} = -40\,dB$, respectively.

Figure 3.2 shows the results obtained by applying the sidelobe control procedure. As can be observed, the beam patterns synthesized by applying at each e-th iteration the *Contiguous Partition* technique show a trade-off between the angular resolution accuracy and noise rejection capabilities depending on the reference excitations $\underline{B}_{ref}^{(e)}$. As a matter of fact, when the difference main lobes get narrower, more power is wasted in the side lobes, and vice versa as confirmed by the values of the indexes reported in Tab. 3.1 . On the other hand, as expected, the SLL of the synthesized patterns get closer and closer to the reference one $SLL_{ref}^{(e)}$ when Q grows (e.g., $SLL_{opt}^{(3)}\Big\rfloor_{Q=2} = -16.20\,dB$ vs. $SLL_{opt}^{(3)}\Big\rfloor_{Q=7} = -31.30\,dB$ when $SLL_{ref}^{(3)} = -40\,dB$). Consequently, it turns out that the $ICPM$ more successfully applies (i.e., satisfying the SLL and bandwidth requirements) when Q is not very small ($Q > 2$). As a matter of fact, the iterative ($e = 1, ..., E$) procedure yields a satisfactory solution at $e = 2$ when $Q = 4$ (being $SLL_{opt}^{(2)}\Big\rfloor_{Q=4} = -22.30\,dB$ and $Bw^{(2)}\rfloor_{Q=4} = 5.1622^o$) and $Q = 7$ (being $SLL_{opt}^{(2)}\Big\rfloor_{Q=7} = -28.80\,dB$ and $Bw^{(2)}\rfloor_{Q=7} = 5.1555^o$), while for $Q = 2$, whatever the iteration ($e = 1, 2, 3$), the fulfillment of the SLL criterion is not met.

As far as the computational issues are concerned, it is worth noting that the $ICPM$ allows a significant reduction of the dimension of the solution space ($U^{(ess)}$ vs. U - Tab. 3.1). Moreover, although the number of possible aggregations changes ($U^{(ess)}\rfloor_{Q=2} = 9$, $U^{(ess)}\rfloor_{Q=4} = 84$, and $U^{(ess)}\rfloor_{Q=7} = 84$) for different values of Q, the computational cost for reaching the termination criterion of the inner loop remains almost the same. In fact, $I_{stat}^{(e)} = 2$ inner iterations are usually enough for determining $\underline{B}_{opt}^{(e)}$, except for the case of $Q = 7$ when $I_{stat}^{(1)} = 3$.

Another interesting observation is concerned with the value of the cost function at the inner loop convergence [i.e., when $i(e) = I_{stat}^{(e)}$]. For a fixed reference pattern, it monotonically decreases as the number of sub-arrays Q tends to M (e.g., $\Psi_{opt}^{(1)}\Big\rfloor_{Q=2} = 3.81 \times 10^{-1}$, $\Psi_{opt}^{(1)}\Big\rfloor_{Q=4} = 9.53 \times 10^{-2}$, and $\Psi_{opt}^{(1)}\Big\rfloor_{Q=7} = 2.29 \times 10^{-3}$) pointing out asymptotically a more accurate matching between the sub-optimal difference mode and the reference one.

3.3.2 Comparative Assessment

In this section, a comparative analysis between the proposed approach and state-of-the-art techniques, based on the optimization of a suitable cost function constructed with reference to a SLL with a prescribed value, is carried out. Both

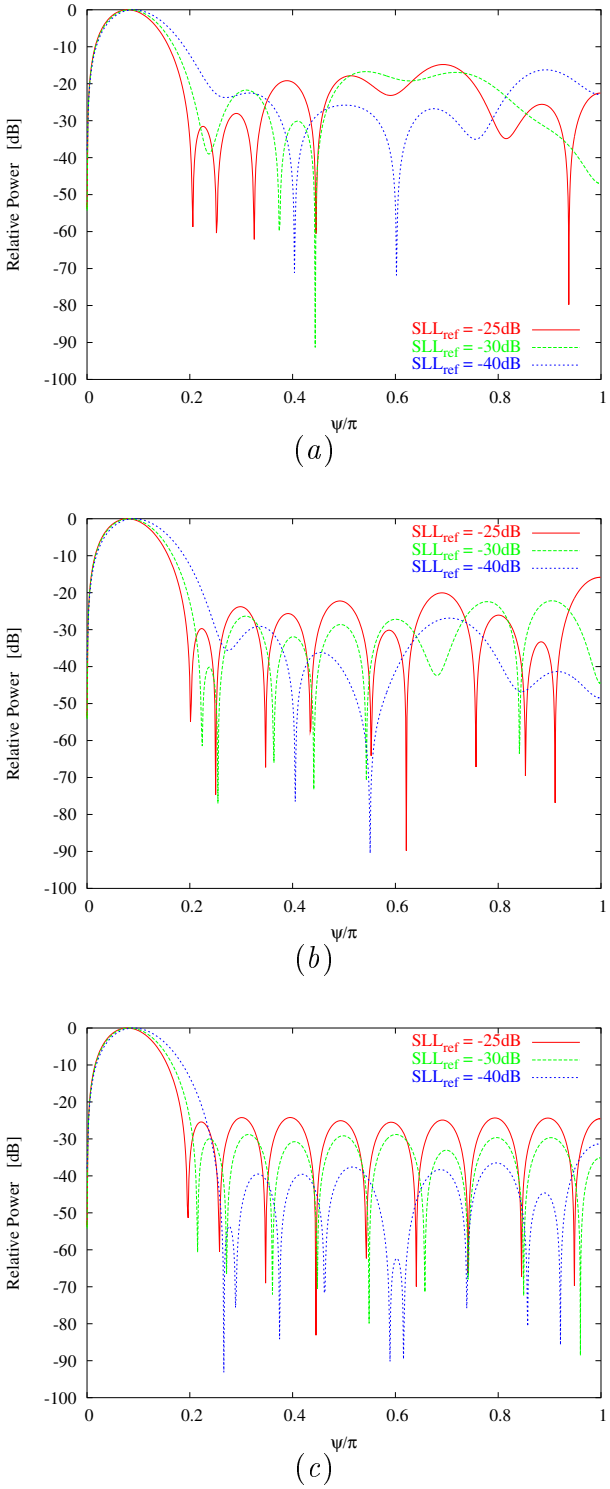


Figure 3.2: *ICPM Performance Analysis* ($M = 10$, $d = \frac{\lambda}{2}$) - Normalized difference patterns when (a) $Q = 2$, (b) $Q = 4$, and (c) $Q = 7$.

	$Q=2$			$Q=4$			$Q=7$		
e	1	2	3	1	2	3	1	2	3
$SLL_{ref}^{(e)}$	$-25\,dB$	$-30\,dB$	$-40\,dB$	$-25\,dB$	$-30\,dB$	$-40\,dB$	$-25\,dB$	$-30\,dB$	$-40\,dB$
A_{slo}	0.1773	0.1865	0.1953	0.1759	0.1840	0.1981	0.1753	0.1844	0.1955
Bw $[deg]$	4.9239	5.2356	5.7661	4.8910	5.1622	5.7976	4.8547	5.1555	5.7217
ψ_1	0.6458	0.7474	0.8463	0.6226	0.7043	0.8653	0.6197	0.6753	0.8368
A_{sll}	0.1761	0.1722	0.1333	0.1112	0.0780	0.0375	0.0938	0.0495	0.0179
$SLL_{opt}^{(e)}$	-14.80	-16.70	-16.20	-15.80	-22.30	-26.90	-24.35	-28.80	-31.30
$I_{stat}^{(e)}$	2	2	2	2	2	2	3	2	2
$\Psi_{opt}^{(e)}$	3.81×10^{-1}	4.62×10^{-1}	2.76×10^{-1}	9.53×10^{-2}	1.10×10^{-1}	3.89×10^{-2}	2.29×10^{-3}	9.93×10^{-4}	5.45×10^{-3}
$U^{(ess)}$	9			84			84		
U	1024			1048580			2.8247×10^8		

Table 3.1: *ICPM Performance Analysis* ($M = 10$, $d = \frac{\lambda}{2}$) - Difference pattern quantitative indexes and computational indicators for different values of Q.

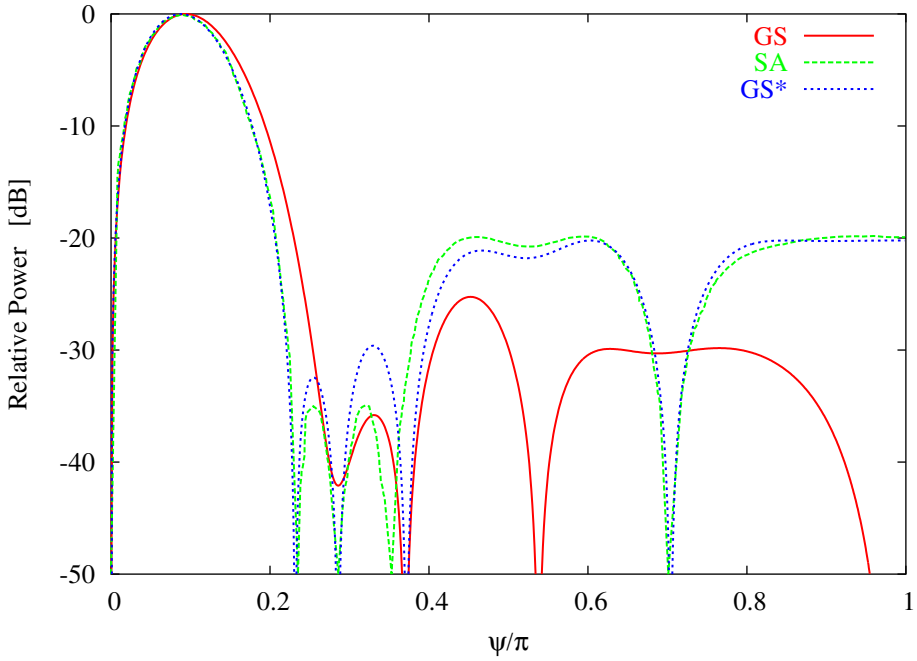


Figure 3.3: *Comparative Assessment* ($M = 10$, $d = \frac{\lambda}{2}$, $Q = 3$) - Normalized difference patterns synthesized with the $ICPM - GS$ and the SA algorithm [10].

fixed-partition (**Test Case 1**) and global-synthesis (**Test Case 2**) problems have been considered.

Test Case 1. Fixed-Partition Synthesis

The first test case deals with the synthesis of a fixed sub-array configuration. With reference to the same benchmark in [10] and addressed by *Ares et al.* with a SA-based technique [10], a linear array of $N = 2 \times M = 20$ equally-spaced ($d = \lambda/2$) elements and $Q = 3$ is considered. The optimal sum excitations have been fixed to that of a Dolph-Chebyshev pattern with $SLL = -35\,dB$ and a Zolotarev difference pattern with $SLL_{ref} = -35\,dB$ has been chosen as reference.

In Figure 3.3 , the difference patterns synthesized with the GS algorithm are compared with that shown in [10]. Moreover, the corresponding sub-array grouping and weights are given in Tab. 3.2 . The GS technique outperforms the SA-based solution in terms of the maximum value the sidelobe level ($SLL_{opt}^{(SA)} = -19.74\,dB$ [10] vs. $SLL_{opt}^{(GS)} = -25.25\,dB$) and allows a three fold reduction of the side lobe power (i.e., $\left\lfloor \frac{A_{sll}^{(SA)}}{A_{sll}^{(GS)}} \right\rfloor \simeq 3$). Moreover, by imposing the compro-

$M = 10$	$\mathbf{C}_{opt}^{(GS)}$	1 1 2 2 2 3 3 3 3 0		
$Q = 3$	$\mathbf{W}_{opt}^{(GS)}$	0.2804	0.5839	1.3971
	$\mathbf{W}_{opt}^{(GS^*)}$	0.4618	2.1607	2.9448

Table 3.2: *Comparative Assessment* ($M = 10$, $d = \frac{\lambda}{2}$, $Q = 3$, $SLL_{ref} = -35\,dB$) - Sub-array configuration and weights synthesized with the $ICPM - GS$.

mise patterns having a maximum Bw close to that of the SA-based technique ($Bw^{(SA)} = 5.5528^o$), the solution of the GS algorithm is shown in Fig. 3.3 (i.e., GS^* - $SLL_{ref} = -33.75\,dB$), while the corresponding sub-array configurations and weights are summarized in Tab. 3.2. In such a situation, the GS is still able to find a better compromise pattern with a SLL below that in [10] of about $0.5\,dB$ ($SLL_{opt}^{(GS^*)} = -20.21\,dB$ - $Bw^{(GS^*)} = 5.4947^o$).

Test Case 2. Simultaneous Global-Synthesis

The second test case is devoted to the comparative assessment when dealing with the simultaneous optimization of the sub-array membership and sub-array weights. Towards this purpose, the proposed method is compared with the GA-based method [12] and the DE algorithm [11].

The first comparison is concerned with the SLL minimization of the difference pattern in a linear array of $N = 2 \times M = 20$ elements with $d = \lambda/2$ inter-element spacing. The optimal sum excitations have been fixed to generate a linear Villeneuve pattern [17] with $\overline{n} = 4$ and sidelobe ratio of $25\,dB$. Moreover, the number of sub-arrays has been set to $Q = 3$ for considering the same example dealt with in [12]. Concerning the $ICPM$, the reference difference pattern has been chosen to be equal to a Zolotarev pattern [9] with $SLL_{ref} = -35\,dB$.

The results of the synthesis process are shown in Figure 3.4 where the reference difference pattern and those obtained with the GA [12] and the constrained EMM [8] are displayed, as well. Concerning the comparison with the GA-based method, the GS outperforms the result in [12] ($SLL_{opt}^{(GA)} = -26.18\,dB$) with a maximum side-lobe level equal to $SLL_{opt}^{(GS)} = -28.60\,dB$ [Tab. 3.3] , and similar bandwidths ($B_w^{(GA)} = 5.7934^o$ and $B_w^{(GS)} = 5.8004^o$). It is interesting to observe that the sub-array configuration determined by the GS algorithm (i.e., $\underline{C} = \{1, 2, 0, 3, 3, 3, 3, 0, 2, 1\}$) is the same obtained in [12], but the sub-array weights are different ($\underline{W}^{(GA)} = \{0.3260,\ 0.6510,\ 1.2990\}$, $\underline{W}^{(GS)} = \{0.2456,\ 0.6018,\ 1.2580\}$). Such an event is due to the fact that in [12] the sub-array gains are part of the optimization process, while in the $ICPM$-based method they are analytically computed once the sub-array configuration has been found. This allows a reduction of the number of unknowns (i.e., only the aggregations instead of weights and aggregations) and, indirectly, of the possibility

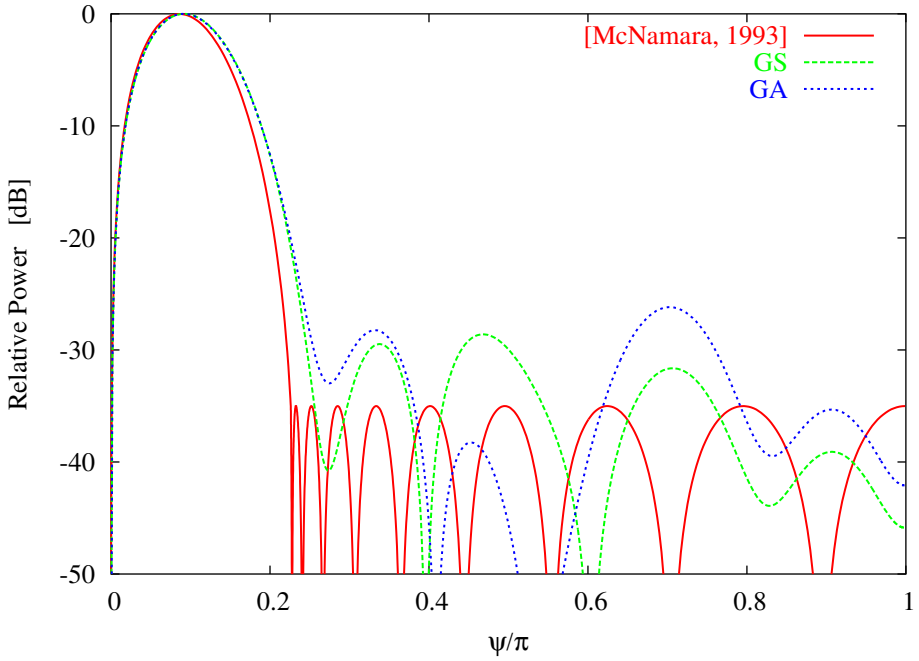


Figure 3.4: *Comparative Assessment* ($M = 10$, $d = \frac{\lambda}{2}$, $Q = 3$) - Reference pattern ($SLL_{ref} = -35\,dB$) and normalized difference patterns synthesized with the $ICPM - GS$, the GA-based method [12], and the constrained EMM [8].

	A_{slo}	$Bw\ [deg]$	A_{sll}	SLL
Reference Difference [9]	0.1933	5.7668	0.0273	−35.00
GS	0.2046	5.8004	0.0382	−28.60
*Reference Difference** [9]	0.1645	4.4747	0.1526	−18.87
*GS**	0.1690	4.5961	0.1453	−17.25
GA Optimization [12]	0.2038	5.7934	0.0440	−26.18
Constrained EMM [8]	0.1715	4.6090	0.2223	−16.50

Table 3.3: *Comparative Assessment* ($M = 10$, $d = \frac{\lambda}{2}$, $Q = 3$) - Quantitative indexes of the reference pattern ($SLL_{ref} = -35\,dB$) and of the difference patterns synthesized with the $ICPM - GS$, the GA-based method [12], and the constrained EMM [8].

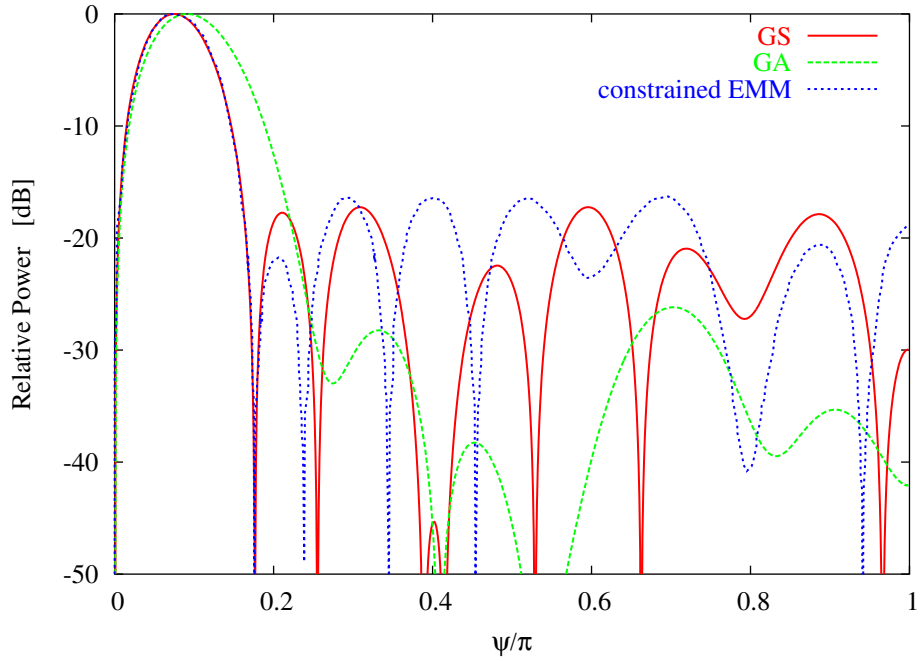


Figure 3.5: *Comparative Assessment* ($M = 10$, $d = \frac{\lambda}{2}$, $Q = 3$) - Normalized difference patterns synthesized with the $ICPM - GS$, the GA-based method [12], and the constrained EMM [8].

the solution being trapped in local minima of the cost function.

As far as the computational costs are concerned, thanks to the reduction of the number of possible aggregations ($U^{(GA)} = 3^{10}$ vs. $U^{(ess)} = 36$) and the searching limited to the sub-array membership, the number of iterations needed for reaching the final solution turns out to be significantly lowered ($I_{stat}^{(GS)} = 3$ vs. $I_{stat}^{(GA)} = 500$ [12]) with a huge computational saving ($T^{(ICPM)} < 0.085$ $[sec]$).

In order to obtain a different trade-off between sidelobe level and beamwidth, exploiting the flexibility of the proposed method, a different reference pattern could be chosen (as highlighted through the analysis in Sect. 3.3.1). As an example and for a further comparison now with another "*optimal matching*" technique instead of the GA, let us relax the requirement on the SLL and request the BW of the compromise patterns being as close as possible to that of the constrained EMM [8]. Towards this aim, a Zolotarev pattern [9] with $SLL_{ref} = -19\,dB$ has been used as reference difference pattern. The synthesized beam patterns are shown in Figure 3.5 . As far as the main lobe is concerned, the beamwidth of the GS^* pattern is narrower ($Bw^{(GS^*)} = 4.5961^o$) than that of the unconstrained GS and very close to that by *McNamara* [8] ($Bw^{(Const-EMM)} = 4.6090^o$). On the other hand, as expected, the performances in terms of SLL get worse ($-17.25\,dB$ vs. $-28.60\,dB$), but they are still better than that of the

$M = 10$	$\mathbf{C}_{opt}^{(GS)}$	1 2 3 4 4 4 4 4 3 1					
	$\mathbf{C}_{opt}^{(GS^*)}$	1 2 3 3 4 4 4 4 3 1					
$Q = 4$	$\mathbf{W}_{opt}^{(GS)}$	0.2201	0.4601		0.6932	0.9568	
	$\mathbf{W}_{opt}^{(GS^*)}$	0.3593	0.7882		1.5351	2.0122	
$M = 10$	$\mathbf{C}_{opt}^{(GS)}$	1 2 3 4 5 6 4 3 2 1					
	$\mathbf{C}_{opt}^{(GS^*)}$	1 2 3 5 6 6 6 4 3 1					
$Q = 6$	$\mathbf{W}_{opt}^{(GS)}$	0.1714	0.5075	0.7332	0.9083	0.9901	0.9926
	$\mathbf{W}_{opt}^{(GS^*)}$	0.1876	0.4765	0.6894	0.8189	0.8914	0.9857

Table 3.4: *Comparative Assessment* ($M = 10$, $d = \frac{\lambda}{2}$) - Sub-array configuration and weights synthesized with the $ICPM - GS$, when $Q = 4$ and $Q = 6$.

SLL-constrained EMM (Tab. 3.3).

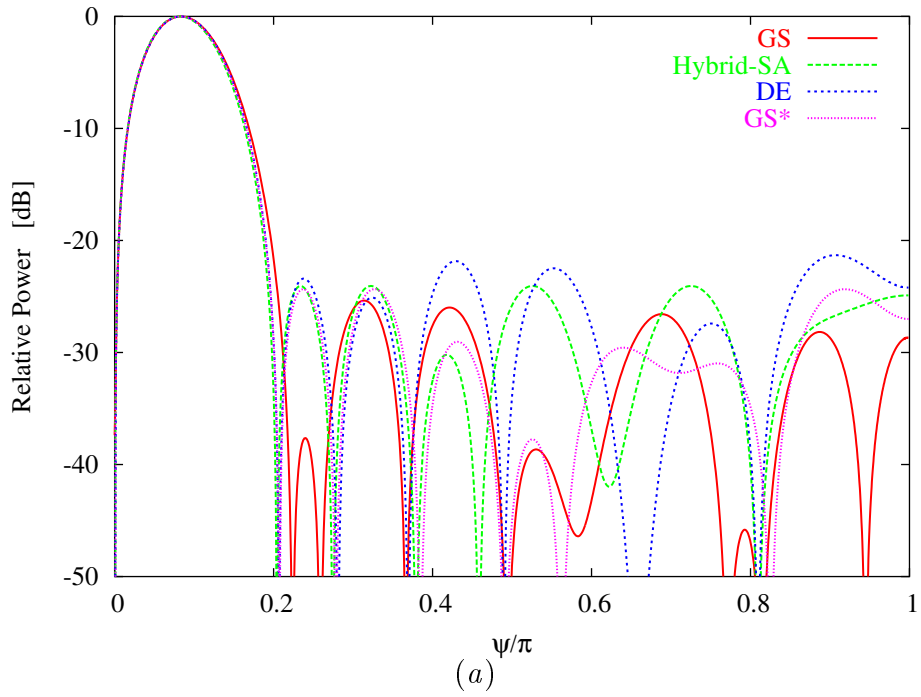
The second example addresses the same problem considered in [11][14] concerned with a 20-elements linear array with $Q = 4$ and $Q = 6$, where the sum pattern is of Dolph-Chebyshev type and characterized by $SLL = -20\,dB$. By assuming reference Zolotarev patterns with $SLL_{ref} = -30\,dB$ ($Q = 4$) and $SLL_{ref} = -35\,dB$ ($Q = 6$), the optimized difference patterns are shown in Fig. 3.6 , while the final sub-array configurations and weights are summarized in Tab. 3.4 .

The *contiguous partition* method is more effective than both the DE-based approach [11] and the two-step procedure proposed in [14] (indicated in figures and tables as $Hybrid - SA$ approach) in minimizing the level of the sidelobes as graphically shown in Fig. 3.6 and quantitatively confirmed by the behavior of the beam pattern indexes in Tab. 3.5 . Similar conclusions hold true in dealing with the required computational burden (Tab. 3.5) and CPU-time ($T^{(GS)} < 0.2$ $[sec]$).

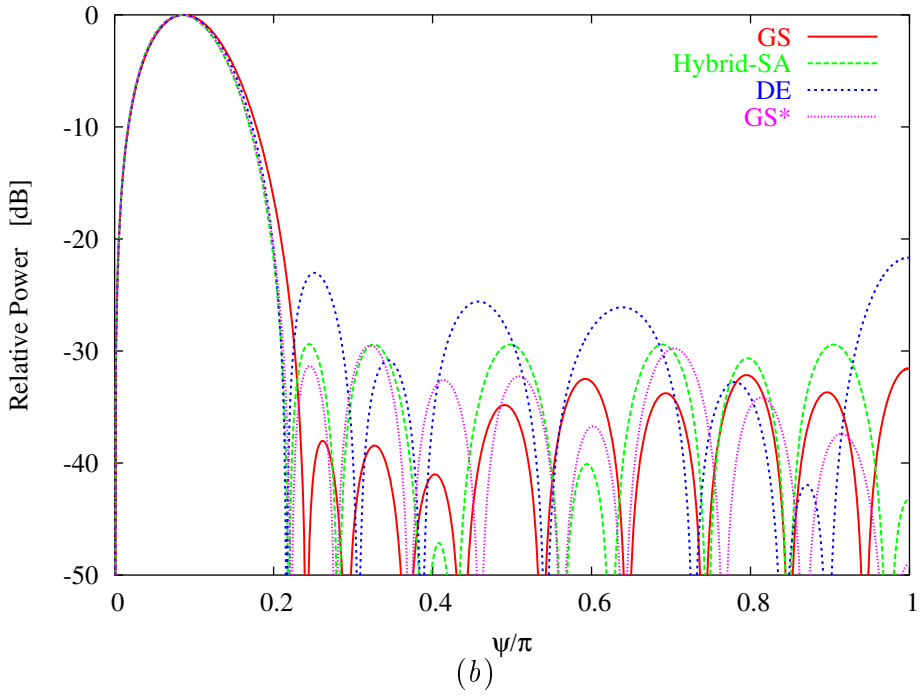
For completeness, the Bw-constrained problem has been also addressed. Accordingly, the SLL minimization has been performed by requiring a beamwidth value close to that in [11] and [14] (Tab. 3.5). The patterns computed with the sub-array configurations and weights given in Tab. 3.4 and synthesized by means of the GS^* algorithm ($Q = 4$ - $SLL_{ref}^{Zolotarev} = -27.50\,dB$, $Q = 6$ - $SLL_{ref}^{Zolotarev} = -33.00\,dB$) are shown in Fig. 3.6. Moreover, the corresponding pattern indexes are summarized in Tab. 3.5.

3.3.3 Extension to Large Arrays

The numerical study ends with analysis of the synthesis of large array patterns ($M \geq 50$) where usually local minima problems, unmanageable (or very difficult) increasing computational costs, and ill-conditioning issues unavoidably arise. In such a framework, the first experiment is concerned with a $N = 2 \times M = 100$ elements array ($d = \lambda/2$) with sum pattern fixed to the Taylor distribution [5]



(a)



(b)

Figure 3.6: *Comparative Assessment* ($M = 10$, $d = \frac{\lambda}{2}$) - Normalized difference patterns synthesized with the $ICPM-GS$, the $Hybrid-SA$ approach [14], and the DE algorithm [11] when (a) $Q = 4$ and (b) $Q = 6$.

	A_{slo}	B_w $[deg]$	A_{sll}	SLL	U	I_{stat}
$Q = 4$						
Reference Difference [9]	0.1786	5.1496	0.0510	-30.00	–	–
GS	0.1809	5.2247	0.0564	-25.40	84	2
Reference Difference* [9]	0.1803	5.0000	0.0694	-27.50	–	–
GS*	0.1863	5.1449	0.0748	-24.30	84	2
$Hybrid - SA$ [14]	0.1844	5.1442	0.0919	-24.10	$\mathcal{O}\left(10^3\right)$	25
DE Algorithm [11]	0.1878	5.1834	0.1107	-21.30	$\mathcal{O}\left(10^3\right)$	9
$Q = 6$						
Reference Difference [9]	0.1929	5.4188	0.0281	-35.00	–	–
GS	0.1948	5.4928	0.0291	-31.56	126	2
Reference Difference* [9]	0.1897	5.3138	0.0355	-33.00	–	–
GS*	0.1893	5.2694	0.0356	-29.52	126	2
$Hybrid - SA$ [14]	0.1884	5.2615	0.0439	-29.50	$\mathcal{O}\left(10^5\right)$	25
DE Algorithm [11]	0.1942	5.3872	0.0727	-21.66	$\mathcal{O}\left(10^5\right)$	7

Table 3.5: *Comparative Assessment* ($M = 10$, $d = \frac{\lambda}{2}$) - Quantitative indexes and computational indicators for the solutions obtained with the $ICPM - GS$, the $Hybrid - SA$ [$I_{stat} = 25$ indicates the number of SA iterations (i.e., first step), no indications on the convex programming procedure (i.e., second step) are available] approach [14], and the DE algorithm [11] when $Q = 4$ and $Q = 6$.

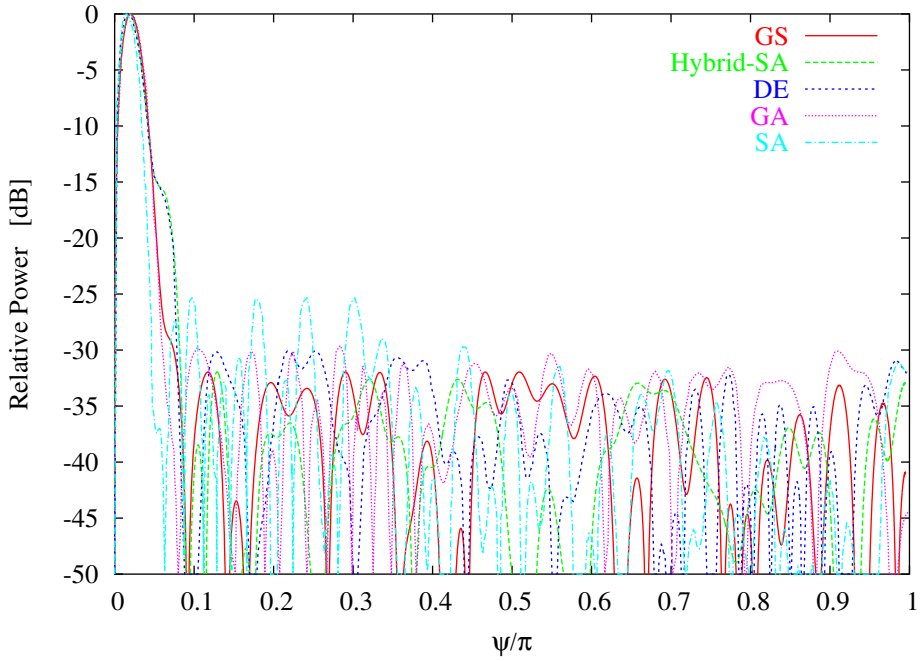


Figure 3.7: *Extension to Large Arrays* ($M = 50$, $d = \frac{\lambda}{2}$, $Q = 4$) - Normalized difference patterns synthesized with the $ICPM - GS$ ($SLL_{ref} = -40\,dB$), the SA algorithm [10], the $Hybrid - SA$ approach [14], the GA-based method [12], and the DE algorithm [11].

with $\overline{n} = 12$ and $SLL = -35\,dB$. For comparison purposes, the case of $Q = 4$ sub-arrays [10][12][11][14] is dealt with. Dealing with such a scenario, the $ICPM$ has been applied by considering a reference Zolotarev pattern [9] with sidelobe level equal to $SLL_{ref} = -40\,dB$.

The synthesized difference pattern is shown in Fig. 3.7 , while the sub-array grouping and weights are given in Tab. 3.6 . By observing both Fig. 3.7 and Tab. 3.7 , it turns out that the GS approach outperforms other single-step techniques and, unlike the case $M = 10$, its performances are quite similar (in terms of sidelobe level) to that of the two-step method even though it is much more computationally effective. Moreover, although it achieves the minimum value of SLL, the corresponding main lobe beamwidth does not significantly

$M = 50$	$\mathbf{C}_{opt}^{(GS)}$	1111222333330444444444444444444430333323222221111			
$Q = 4$	$\mathbf{W}_{opt}^{(GS)}$	0.1624	0.5162	0.8579	1.1736

Table 3.6: *Extension to Large Arrays* ($M = 50$, $d = \frac{\lambda}{2}$, $Q = 4$) - Sub-array configuration and weights synthesized with the $ICPM - GS$.

Synthesis Approach	SLL_{opt}	A_{sll}	B_w $[deg]$	A_{slo}	U	I_{stat}	T $[sec]$
SA Optimization [10]	-25.56	0.0432	1.0745	0.0329	$\mathcal{O}\left(10^{30}\right)$	–	–
GA Optimization [12]	-31.00	0.0504	1.3585	0.0529	$\mathcal{O}\left(10^{30}\right)$	500	~ 15
DE Algorithm [11]	-30.00	0.0361	1.3256	0.0361	$\mathcal{O}\left(10^{30}\right)$	804	~ 20
$Hybrid - SA$ *Method* [14]	-32.00	0.0305	1.2776	0.0401	$\mathcal{O}\left(10^{30}\right)$	25	–
GS	-32.10	0.0363	1.2952	0.0444	18424	5	1.0785

Table 3.7: *Extension to Large Arrays* ($M = 50$, $d = \frac{\lambda}{2}$, $Q = 4$) - Quantitative indexes and computational indicators for the solutions obtained with the $ICPM - GS$ ($SLL_{ref} = -40\,dB$), the $Hybrid - SA$ [$I_{stat} = 25$ indicates the number of SA iterations (i.e., first step), no indications on the convex programming procedure (i.e., second step) are available], the SA algorithm [10], the GA-based method [12], and the DE algorithm [11].

$M = 50$	$\mathbf{C}_{opt}^{(GS)}$	11111222202000333333333333303300002222222211111111		
$Q = 3$	$\mathbf{W}_{opt}^{(GS)}$	0.2437	0.7079	1.0976

Table 3.8: *Extension to Large Arrays* ($M = 50$, $d = \frac{\lambda}{2}$, $Q = 3$) - Sub-array configuration and weights synthesized with the $ICPM - GS$.

differ from that of the other methods (Tab. 3.7).

In the second experiment, the same array geometry of the previous case is analyzed, but with $Q = 3$ sub-arrays analogous to [12]. The sub-array configuration and weights obtained with the GS-based strategy are reported in Tab. 3.8 . Also in this case, the GS difference pattern presents a SLL lower than that shown in [12] ($SLL_{opt}^{(GS)} = -30.25$ vs. $SLL_{opt}^{(GA)} = -29.50$) and confirms its effectiveness in terms of computational resource since $\frac{I_{stat}^{(GA)}}{I_{stat}^{(GS)}} = 250$.

3.4 Discussions

The proposed method consists of an adaptive searching procedure whose result is a compromise solution as close as possible to an optimal one in the Dolph-Chebyshev sense, which allows a satisfactory trade-off between angular resolution and reduction of noise and interferences effects. In particular, the narrowest beamwidth and the largest slope around the boresight direction are looked for by applying the optimal excitation matching method based on the contiguous partition technique, while the fulfillment of the requirements on the SLL (or other beam pattern features) is ensured by an outer iterative loop.

The obtained results have proved the effectiveness of the proposed approach in providing difference patterns with a satisfactory trade-off among beam pattern features dealing with large arrays, as well. Although the iterative contiguous

partition method is aimed at synthesizing the "best compromise" matching an optimal (in the Dolph-Chebyshev sense) reference pattern, the obtained solutions positively compare with state-of-the-art approaches in the related literature in a number of measures where only the SLL minimization is required, thus showing how the proposed approach, which is numerically efficient, works sufficiently well. As a matter of fact, the proposed technique allows one to overcome some drawbacks of both the EMM approach proposed by *McNamara* (i.e., ill-conditioning and the exhaustive evaluation of the whole set of aggregations) and the synthesis techniques based on stochastic optimization algorithms (i.e., single-objective optimization and low convergence rate when dealing with very large arrays).

On the other hand, definite conclusions about the relative performance of the $ICPM$ cannot be drawn from the presented comparisons, since the various examples deal with different synthesis problems and/or optimization criteria. This means that, depending on the selected feature, the $ICPM$ performs differently even though keeping a great computational efficiency. Moreover, since the proposed procedure is an adaptive searching technique, it does not guarantee to always obtain better solutions than those from global optimization techniques. As a matter of fact, these latter should outperform any other approach when optimizing a given functional, unless the optimum is not actually achieved, which is likely to happen when exploiting global optimization algorithms in large size problems.

Chapter 4

Monopulse Planar Array Synthesis

In this chapter, the extension of the *Contiguous Partition Method* (CPM) from linear to planar arrays is described and assessed. By exploiting some properties of the solution-tree the solution space is represented in terms of a more compact graph. The generation of compromise sum-difference patterns is thus obtained through an optimal excitation matching procedure based on a combinatorial method. A set of representative results are reported for the assessment as well as for comparison purposes.

4.1 Introduction

Exact methods of synthesizing independently optimum sum and difference arrays exist for both linear [19][17][9][22] and planar architectures [6][7]. Whether the complexity and cost of the arising feed networks are affordable, then the above methods can be directly used. However, since the implementation of two (or three) totally independent signal feeds is generally expensive and complex, a number of alternative solutions have been proposed to generate the two or three required patterns via shared feed networks at the cost of a reduction in the quality of one or more patterns [2][1].

In order to avoid the need of a completely different feeding (receiving) network for each operation mode, several researches [8][10][12][11][14] proposed to partition the original array in sub-arrays. In such a scheme, the feeding network is usually devoted to the optimization of the sum channel, so that the excitations of the arrays elements for such a mode correspond to the optimal one (e.g., Taylor [6]). Then, the difference mode is obtained thanks to a suitable choice of the weight of each sub-array. Consequently, a large part of the whole architecture is common to both modes with a non negligible saving of costs. On the other hand, a compromise difference pattern is obtained. The degree of optimality of the compromise solution is related to the number of sub-arrays, which establishes a trade-off between costs and performances. As a matter of fact, a large number of sub-arrays allows better performances, but also implies higher costs. Otherwise, few sub-arrays may imply unacceptable difference patterns. For a fixed number of sub-arrays, once the excitations of the sum pattern have been fixed, the problem is concerned with the grouping of the array elements into sub-arrays and the computation of their weights to determine the best compromise difference pattern. As far as the number of unknowns is concerned, it grows proportionally to the dimension of the array and, usually, it turns out to be very large when real applications of planar arrays are considered. Consequently, a standard use of global optimization techniques is not convenient since a suboptimal solution is generally obtained in the limited time one has at his disposal. As a matter of fact, the arising computational burden raises very rapidly with the dimension of the solution space. Although this circumstance is quite underestimated in antenna design since synthesis problems may have many different satisfactory suboptimal solutions, nevertheless they can be significantly worse than the global ones.

In order to overcome such drawbacks, in Ares et *al.* [10] the antenna aperture has been divided into four quadrants and the monopulse function has been obtained by combining the outputs in a monopulse comparator. The sum pattern and the difference one have been generated with all quadrants added in phase and with pairs of quadrants added in phase reversal, respectively. Moreover, in order to reduce the number of unknowns, each antenna quadrant has been *a-priori* divided into sub-arrays (i.e., the sectors) and only the sub-array weights

have been calculated by minimizing a suitable cost function again according to a Simulated Annealing (SA) algorithm. In an alternative fashion, D'Urso et *al.* [13] formulated the problem in such a way that global optimization tools have to deal with a reduced number of unknowns. By exploiting the convexity of the cost functional to be minimized with respect to a part of the unknowns (i.e., the sub-array gains), an hybrid two-step optimization strategy has been applied instead of simultaneously optimizing (in the same way) both the involved variables. As a matter of fact, once the clustering into sub-arrays has been determined by using a SA technique, the problem at hand gives rise to a Convex Programming (CP) problem with a single minimum that can be retrieved with a local optimization technique. Unfortunately, although unlike [10] no *a-priori* informations are necessary, the evaluation of the auxiliary CP objective function is usually more cumbersome than the original cost function. Such an event could result in an excessively large computational burden that would prevent the retrieval of the global optimum in the available amount of time or to efficiently deal with large planar arrays.

In the following, the method proposed in Chapter 3 is considered for the synthesis of planar monopulse array antenna. Towards this end, a suitable implementation is mandatory to keep also in the planar case the best features of the linear approach both in term of reliability and computational efficiency. As a matter of fact, unlike the linear case, the planar structure requires two difference patterns (i.e., the difference $E-mode$ and the $H-mode$). Moreover, the dimensionality of the problem at hand significantly grows with respect to the linear situation, thus enhancing the computational problems in applying global optimization methodologies and thus preventing their use also in hybrid modalities.

Therefore this paper is aimed at describing and assessing the planar extension of the CPM (in the following $PCPM$) according to the following outline. The mathematical formulation is presented in Sect. 4.2 pointing out the main differences compared to the linear array case. Section 4.3 is devoted to the numerical assessment. Both a consistency check, carried out through an asymptotic study, and a comparative analysis (unfortunately, just only a test case is available in the recent literature) are considered.

4.2 Mathematical Formulation

Let us consider a planar array lying on the $xy-plane$ whose array factor is given by

$$AF(\theta,\phi)=\sum_{r=-R}^{R}\sum_{s=-S(r)}^{S(r)}\xi_{rs}e^{j(k_x x_r+k_y y_s)},\quad r,s\neq 0 \tag{4.1}$$

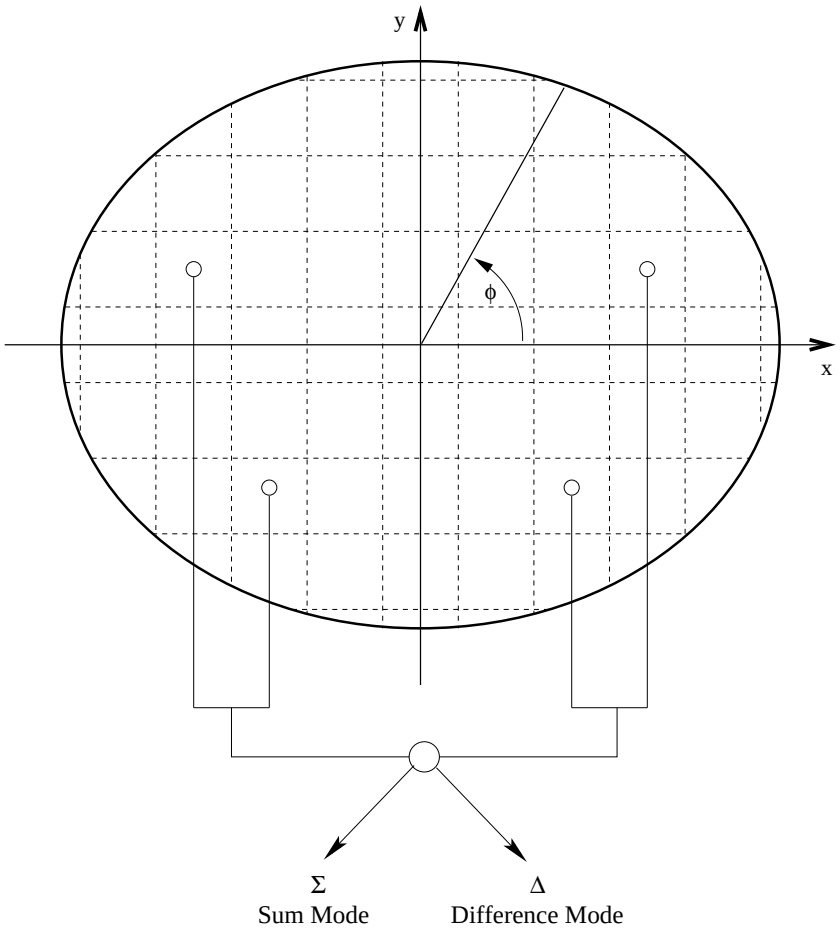


Figure 4.1: Sketch of the antenna feed network.

where $x_r = \left[r - \frac{sgn(r)}{2}\right] \times d_x$ and $y_s = \left[s - \frac{sgn(s)}{2}\right] \times d_y$, d_x and d_y being the inter-element distance along the x and y direction, respectively. Moreover, $k_x = \frac{2\pi}{\lambda} sin\theta cos\phi$ and $k_y = \frac{2\pi}{\lambda} sin\theta sin\phi$. Concerning independently optimum sum and difference patterns, they are generated by using three independent feeding networks and setting the excitation vector $\underline{\xi} = \{\xi_{rs};\ r = \pm 1, ..., \pm R;\ s = \pm 1, ..., \pm S(r)\}$ to $\underline{\zeta} = \left\{\zeta_{rs} = \zeta_{(-r)s} = \zeta_{r(-s)} = \zeta_{(-r)(-s)};\ r = 1, ..., R;\ s = 1, ..., S(r)\right\}$ and to $\underline{\varsigma}^{\triangle} = \left\{\varsigma^{\triangle}_{rs} = \varsigma^{\triangle}_{(-r)s} = -\varsigma^{\triangle}_{r(-s)} = -\varsigma^{\triangle}_{(-r)(-s)};\ r = 1, ..., R;\ s = 1, ..., S(r)\right\}$, $\triangle = E$, H, respectively. Otherwise, when sub-arraying strategies are considered [10] (Fig. 4.1), the sum beam is generated in an optimal fashion by fixing $\underline{\xi} = \underline{\zeta}$, while the compromise $\triangle-$modes are obtained through a grouping operation described by the aggregation vectors $\underline{c}^{\triangle}$

$$\underline{c}^{\triangle} = \left\{c^{\triangle}_{rs};\ r = 1, ..., R;\ s = 1, ..., S(r)\right\} \tag{4.2}$$

where $c^{\triangle}_{rs} \in [1, Q]$ is the sub-array index of the element located at the r-th row and s-th column within the array architecture. Accordingly, the compromise

difference excitations are given by

$$\underline{\gamma}^{\triangle} = \left\{\gamma_{rs}^{\triangle} = \zeta_{rs} O\left(c_{rs}^{\triangle},\, q\right) g_q^{\triangle};\, r = 1, ..., R;\, s = 1, ..., S(r);\, q = 1, ..., Q\right\} \quad (4.3)$$

where $g_q^{\triangle}$ is the gain coefficient of the q-th sub-array and $O\left(c_{rs}^{\triangle},\, q\right) = 1$ if $c_{rs}^{\triangle} = q$ and $O\left(c_{rs}^{\triangle},\, q\right) = 0$, otherwise. Summarizing, the problem of defining the best compromise between sum and difference patterns is recast as the definition of the configuration $\underline{c}_{opt}^{\triangle}$ and the corresponding set of weights $\underline{g}_{opt}^{\triangle}$ so that $\underline{\gamma}_{opt}^{\triangle}$ is as close as much as possible to $\underline{\varsigma}^{\triangle}$.

Towards this end, the CPM is applied. Similarly to the linear array case, the following cost function is defined

$$\Psi\left(\underline{c}^{\triangle}\right) = \frac{1}{N} \sum_{r=1}^{R} \sum_{s=1}^{S(r)} \zeta_{rs}^2 \left| \left[\alpha_{rs}^{\triangle} - \sum_{q=1}^{Q} w_{rsq}\left(\underline{c}^{\triangle}\right) \right] \right|^2 \quad (4.4)$$

where N is the number of elements lying on the aperture [i.e., $N = \sum_{r=1}^{R} S(r)$]. Moreover, $\alpha_{rs}^{\triangle} = \frac{\varsigma_{rs}^{\triangle}}{\zeta_{rs}}$ and $w_{rsq}^{\triangle} = w_{rsq}\left(\underline{c}^{\triangle}\right)$ is given by

$$w_{rsq}^{\triangle} = \frac{\sum_{r=1}^{R} \sum_{s=1}^{S(r)} \zeta_{rs}^2 O\left(c_{rs}^{\triangle},\, q\right) \alpha_{rs}^{\triangle}}{\sum_{r=1}^{R} \sum_{s=1}^{S(r)} \zeta_{rs}^2 O\left(c_{rs}^{\triangle},\, q\right)}, \quad r = 1, ..., R;\;\; s = 1, ..., S(r);\;\; q = 1, ..., Q. \quad (4.5)$$

As regards to the sub-array weights, they are computed once the aggregation vector $\underline{c}^{\triangle}$ has been identified by simply using the following relationship

$$g_q^{\triangle} = O\left(c_{rs}^{\triangle},\, q\right) w_{rsq}^{\triangle} \qquad r = 1, ..., R;\;\; s = 1, ..., S(r);\;\; q = 1, ..., Q. \quad (4.6)$$

In order to determine the unknown clustering that minimizes (4.4), the indication given in [18] has been exploited. More in detail, it has been proved that a *contiguous partition* of the array elements is the optimal compromise solution. Accordingly, the set of contiguous partitions (i.e., the set of admissible solutions) is defined by iteratively partitioning in Q sub-sets the list $V = \{v_n;\, n = 1, ..., N\}$ (n being the list index) of the array elements ordered according to the corresponding $\alpha_{rs}^{\triangle}$ values such that $v_n \leq v_{n+1}$ $(n = 1, ..., N-1)$, $v_1 = min_{rs}\left\{\alpha_{rs}^{\triangle}\right\}$, $v_N = max_{rs}\left\{\alpha_{rs}^{\triangle}\right\}$.

Although the dimension of the $PCPM$ solution space, $\Im^{PCPM}$, is significantly reduced compared to that of full global optimizers [$U^{(PCPM)} = \begin{pmatrix} N-1 \\ Q-1 \end{pmatrix}$ vs. $U^{(GA)} = Q\left(Q^{N-1}+1\right)$] or hybrid global-local optimization techniques [$U^{(Hybrid)} = Q^N$], non-negligible computational problems still remain since the large amount of computational resources needed to sample $\Im^{PCPM}$ especially when N enlarges as it happens in realistic planar architecture. Therefore, it is mandatory to devise an effective sampling procedure able to guarantee a good trade-off between

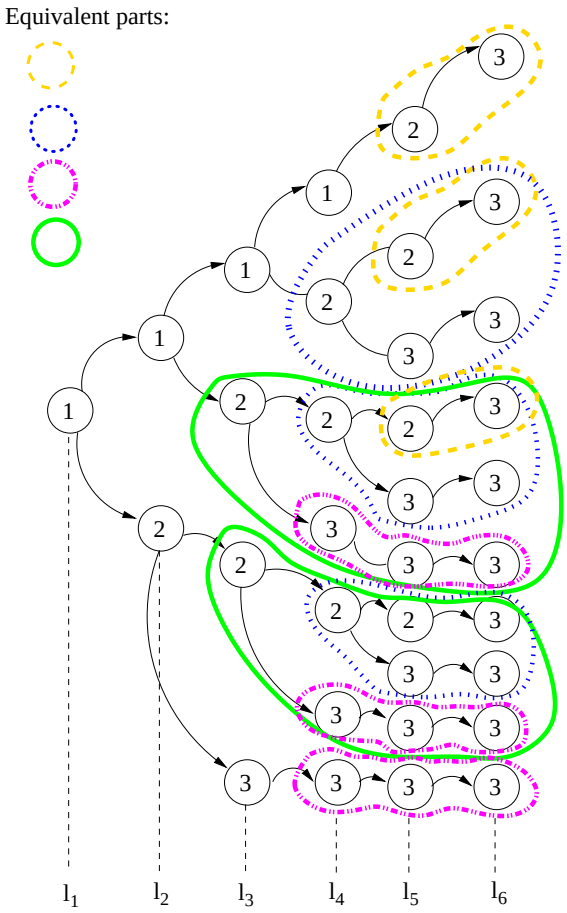


Figure 4.2: Pictorial representation of the redundant parts within the solution tree.

computational costs and optimality of the synthesized compromise solution. Towards this end, the "solution tree" of the linear case has been collapsed into a more compact structure (Fig. 4.2) , namely the *direct acyclic graph* (*DAG*) [16], to describe the whole solution space. Such a representation enables the excitation matching synthesis of planar arrays with large numbers of elements thanks to the significant reduction of both the computational time and the *CPU* memory requirements. Moreover, the *DAG* allows the implementation and an effective use of a fast graph-searching algorithm to look for the optimal planar compromise.

More in detail, the *DAG* is composed by Q rows and N columns. The q-th row is related to the q-th sub-array ($q = 1, ..., Q$), whereas the n-th column ($n = 1, ..., N$) maps the v_n-th element of the ordered list V. An admissible compromise solution is coded into a path, denoted by ψ, in the *DAG*. Each path ψ is described by a set of N vertexes, $\{t_n;\ n = 1, ..., N\}$ and through $N-1$ relations/links $\{e_n;\ n = 1, ..., N-1\}$ among the vertexes belonging to the path.

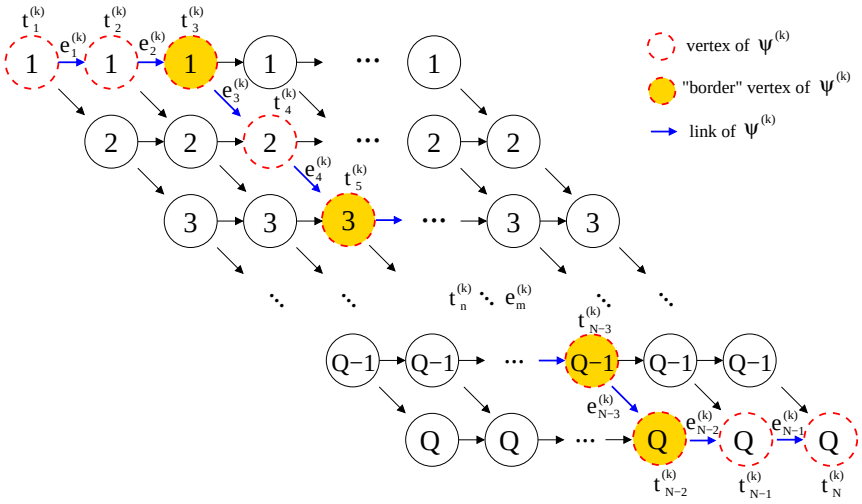


Figure 4.3: *DAG* Representation.

With reference to Fig. 4.3 , each vertex t_n is indicated by a circle and each link e_n with an arrow from a vertex t_n to another one t_{n+1} on the same row [i.e., $arg\,(t_n) = arg\,(t_{n+1}) = r_n$, being r_n the row of the n-th vertex, $r_n \in [1,\, Q]$] or down to an adjacent row [i.e., $arg\,(t_{n-1}) = r_n$ and $arg\,(t_n) = r_n + 1$].

In order to identify the optimal compromise (or, in an equivalent fashion, the optimal path ψ_{opt} in the *DAG*), let us reformulate the concept of "*border elements*" of the linear case to the planar representation in terms of *DAG*. Moreover, let us consider that analogously to the linear case, only the "*border elements*" of ψ (i.e., those vertexes t_n, $n = 2, ..., N-1$ having at most one of the adjacent vertexes, t_{n-1} or t_{n+1}, that belongs to a different row of the *DAG*) are candidate to change their sub-array membership without generating non-admissible aggregations. Accordingly, in order to determine the optimal sub-array configuration $\underline{c}^{\triangle}_{opt}$ that minimizes $\Psi\left(\underline{c}^{\triangle}\right)$ (4.4), a sequence of trial paths $\psi^{(k)} = \left\{ \left(t_n^{(k)}, e_m^{(k)} \right);\, n = 1, ..., N;\, m = 1, ..., N-1 \right\}$ (k being the iteration/trial index) is generated. Starting from an initial path $\psi^{(k)}$ ($k = 0$) defined by setting $arg\left(t_1^{(0)}\right) = 1$ and $arg\left(t_N^{(0)}\right) = Q$ and randomly choosing the other vertexes such as $arg\left(t_{n-1}^{(0)}\right) \leq arg\left(t_n^{(0)}\right) \leq arg\left(t_{n+1}^{(0)}\right)$, the path $\psi^{(k)}$ is iteratively updated ($\psi^{(k)} \leftarrow \psi^{(k+1)}$, $\underline{c}^{\triangle(k)} \leftarrow \underline{c}^{\triangle(k+1)}$) just modifying the memberships of the border elements of the *DAG*. More in detail, the "*border*" vertexes are updated as follows

$$arg\left(t_n^{(k+1)}\right) = \begin{cases} r_n^{(k)} + 1 & if \quad r_{n-1}^{(k)} = r_n^{(k)} \\ r_n^{(k)} - 1 & if \quad r_{n+1}^{(k)} = r_n^{(k)} \end{cases}, \tag{4.7}$$

while the links $e_{n-1}^{(k)} \triangleq link\left[arg\left(t_{n-1}^{(k)}\right), arg\left(t_n^{(k)}\right)\right]$ and $e_n^{(k)} \triangleq link\left[arg\left(t_n^{(k)}\right), arg\left(t_{n+1}^{(k)}\right)\right]$

connected to the "border" vertex $t_n^{(k)}$ are modified through the relationships

$$e_{n-1}^{(k+1)} = \begin{cases} link\left[r_n^{(k)},\, r_n^{(k)}+1\right] & if \quad r_{n-1}^{(k)} = r_n^{(k)} \\ link\left[r_n^{(k)}-1,\, r_n^{(k)}-1\right] & if \quad r_{n+1}^{(k)} = r_n^{(k)} \end{cases} \tag{4.8}$$

and

$$e_n^{(k+1)} = \begin{cases} link\left[r_n^{(k)}+1,\, r_n^{(k)}+1\right] & if \quad r_{n-1}^{(k)} = r_n^{(k)} \\ link\left[r_n^{(k)}-1,\, r_n^{(k)}\right] & if \quad r_{n+1}^{(k)} = r_n^{(k)} \end{cases} . \tag{4.9}$$

The iterative process stops when a maximum number of iterations K_{max} ($k > K_{max}$) or the following stationary condition holds true:

$$\frac{\left|K_w \Psi^{(k-1)} - \sum_{h=1}^{K_w} \Psi^{(h)}\right|}{\Psi^{(k)}} \leq \eta \tag{4.10}$$

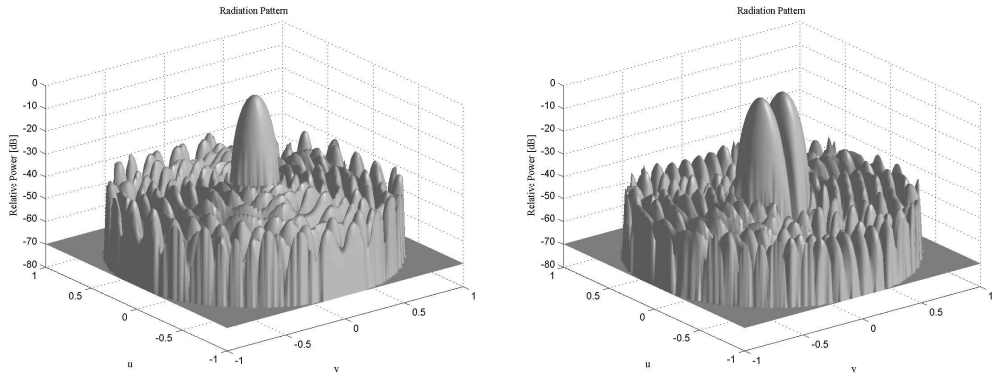
where $\Psi^{(k)} = \Psi\left(\underline{c}^{\triangle(k)}\right)$, K_w and η being a fixed number of iterations and a fixed numerical threshold, respectively. At the end of the iterative sampling of the DAG, the path ψ^{opt} is found and the corresponding aggregation vector, $\underline{c}_{opt}^{\triangle}$, is assumed as the optimal compromise solution.

4.3 Numerical Results

This section is aimed at assessing the effectiveness of the $PCPM$ through a set of representative results from several numerical simulations. The remaining of this section is organized as follows. Firstly, some experiments are presented in Sub-Sect. 4.3.1 to analyze the behavior of the proposed approach in matching a reference pattern for different numbers of sub-arrays. Successively, a comparative study is carried out (Sub-Sect. 4.3.2) by considering the available test case concerned with planar geometries and previously faced in [10].

4.3.1 Pattern Matching

In the first test case, the planar array consists of $N_{tot} = 4 \times N = 316$ elements equally-spaced ($d_x = d_y = \frac{\lambda}{2}$) elements arranged on a circular aperture $r = 5\lambda$ in radius. Because of the circular symmetry, the synthesis procedure is only concerned with $N = 79$ elements. Moreover, the sum pattern excitations $\underline{\zeta}$ have been fixed to those of a Taylor pattern [6] with $SLL = -35\,dB$ and $\overline{n} = 6$. On the other hand, the optimal difference $H-mode$ excitations $\underline{\varsigma}^H$ have been chosen to afford a Bayliss pattern [7] with $SLL = -40\,dB$ and $\overline{n} = 5$. The corresponding three-dimensional ($3D$) representations of the relative power distributions are reported in Fig. 4.4 where $u = \sin\theta\cos\phi$ and $v = \sin\theta\sin\phi$ [23], being $\theta \in$



(a) (b)

Figure 4.4: *Pattern Matching* ($N = 316$, $d = \frac{\lambda}{2}$, $r = 5\lambda$) - Relative power distribution of the reference (a) Taylor sum pattern ($SLL = -35\,dB$, $\overline{n} = 6$) [6] and of the (b) $H - mode$ Bayliss difference pattern ($SLL = -40\,dB$, $\overline{n} = 5$) [7], respectively.

$[0, 90^o]$ and $\phi \in [0, 360^o]$, respectively. As regards to the compromise synthesis, the optimization has been limited to the difference $H - mode$ by exploiting the following relationship $\underline{\gamma}^E = \left\{\gamma_{rs}^E = -\gamma_{rs}^H;\, r = 1, ..., R;\, s = 1, ..., S(r)\right\}$ that holds for the $E - mode$ excitations due to the symmetry properties.

In the first experiment, the number of sub-arrays has been varied from $Q = 3$ up to $Q = 10$. Figure 4.5 shows the $3D$ representations of the synthesized $H - mode$ patterns. As it can be observed, the shapes of both the main lobes and the sidelobes of the compromise distributions get closer to the reference one [Fig. 4.4(b)] when the ratio $\frac{N}{Q}$ reduces. In order to better show such a trend and to efficiently represent the behavior of the side-lobes, let us analyze the *sidelobe ratio* (SLR) defined as

$$SLR\,(\phi) = \frac{SLL\,(\phi)}{max_\theta\left[AF\,(\theta, \phi)\right]},\quad 0 \leq \theta < \frac{\pi}{2} \tag{4.11}$$

where $AF\,(\theta, \phi)$ indicates the array factor. By following the same guidelines in [10], the SLR has been controlled in the range $\phi \in [0^o, 80^o]$ since the $H - mode$ pattern vanishes at $\phi = 90^o$. As expected, the behavior of the SLR approximates that of the reference pattern when Q increases (Fig. 4.6) . Such an indication is quantitatively confirmed by the statistics of the SLR values given in Tab. 4.1 as well as, pictorially, by the plots in Fig. 4.7 where the pattern values along the $\phi = 0^o$ cut are shown.

4.3.2 Comparative Assessment

To the best of the author's knowledge, the topic of planar sub-arraying has been recently addressed only by Ares *et al.* in [10]. More in detail, a Simulated

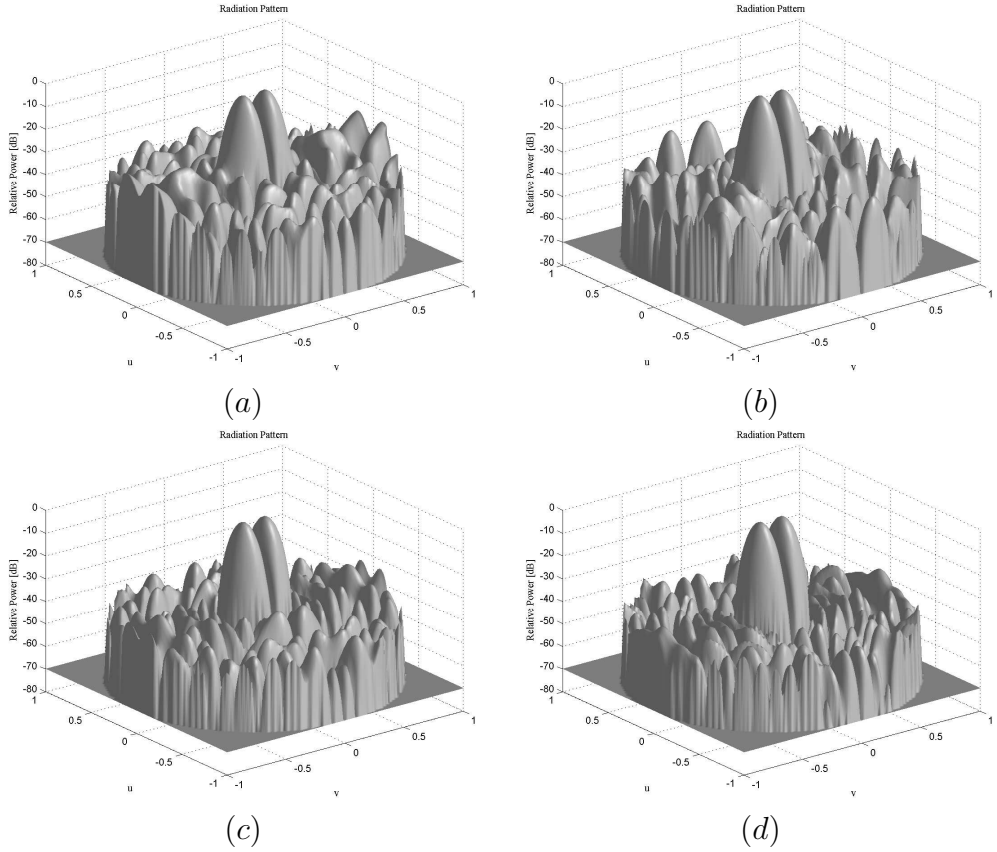


Figure 4.5: *Pattern Matching* ($N = 316$, $d = \frac{\lambda}{2}$, $r = 5\lambda$) - Relative power distribution of the synthesized $H - mode$ difference pattern when (a) $Q = 3$, (b) $Q = 4$, (c) $Q = 6$, and (d) $Q = 10$.

$[dB]$	$min\{SLR\}$	$max\{SLR\}$	$av\{SLR\}$	$var\{SLR\}$
$Reference$ [7]	-40.44	-27.29	-36.68	6.05
$Q = 3$	-33.82	-16.48	-26.74	14.26
$Q = 4$	-37.32	-15.68	-31.56	15.11
$Q = 6$	-36.67	-17.47	-31.25	26.30
$Q = 10$	-38.72	-23.75	-34.77	11.46

Table 4.1: *Pattern Matching* ($N = 316$, $d = \frac{\lambda}{2}$, $r = 5\lambda$) - Statistics of the SLR values in Fig. 3.

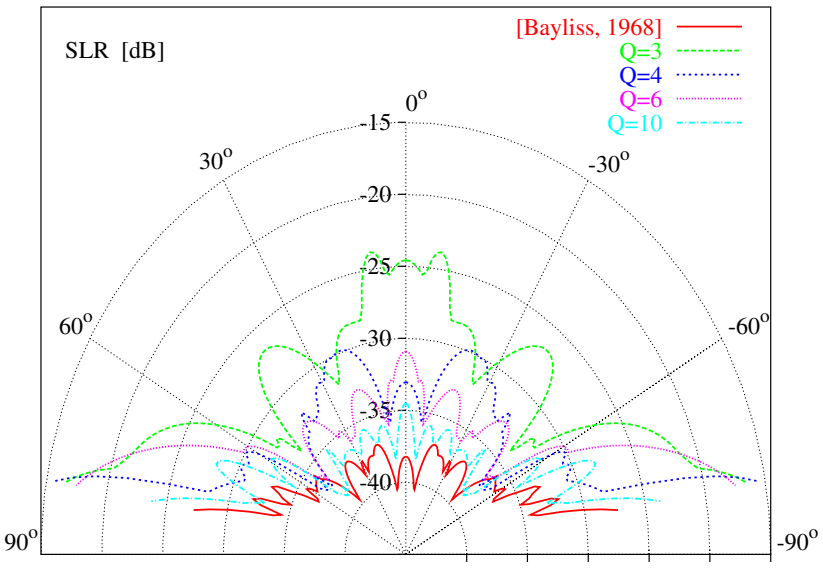


Figure 4.6: *Pattern Matching* ($N = 316$, $d = \frac{\lambda}{2}$, $r = 5\lambda$) - Plots of the SLR values of the Bayliss pattern ($SLL = -40\,dB$, $\overline{n} = 5$) [7] and of the compromise $H - mode$ difference patterns when $Q = 3$, 4, 6, 10 ($\phi \in [-80^o, 80^o]$).

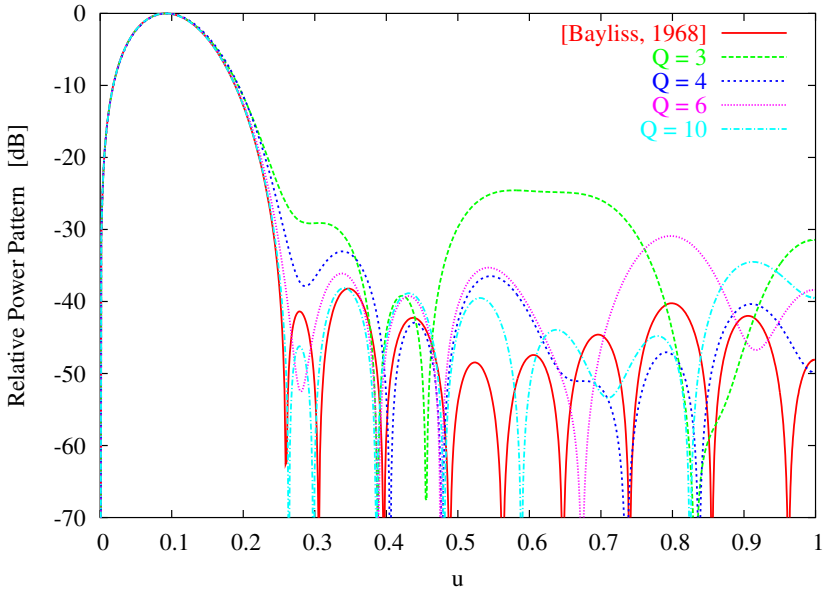


Figure 4.7: *Pattern Matching* ($N = 316$, $d = \frac{\lambda}{2}$, $r = 5\lambda$) - Azimuthal ($\phi = 0^o$) plots of the relative power of the Bayliss pattern ($SLL = -40\,dB$, $\overline{n} = 5$) [7] and of the compromise $H - mode$ patterns when $Q = 3$, 4, 6, 10.

$[dB]$	$min\{SLR\}$	$max\{SLR\}$	$av\{SLR\}$	$var\{SLR\}$
SA [10]	-27.70	-18.93	-22.52	6.41
$CPM: SLL_{ref}^{H(1)} = -25\,dB$	-23.30	-14.58	-21.48	3.93
$CPM: SLL_{ref}^{H(2)} = -30\,dB$	-28.78	-16.95	-24.08	14.15
$CPM: SLL_{ref}^{H(3)} = -35\,dB$	-29.43	-18.94	-25.87	5.74

Table 4.2: *Comparative Assessment* ($N = 300$, $d = \frac{\lambda}{2}$, $r = 4.85\lambda$, $Q = 3$) - Statistics of the SLR values of the $H-mode$ difference pattern synthesized with the SA approach [10] and with the iterative $PCPM$ (Reference Bayliss pattern $\overline{n} = 6$ [7]: $SLL_{ref}^{H(1)} = -25\,dB$, $SLL_{ref}^{H(2)} = -30\,dB$, and $SLL_{ref}^{H(3)} = -35\,dB$).

Annealing (SA) procedure has been used to determine the sub-array weights for a *pre-fixed* sub-array configuration by minimizing a suitable cost function aimed at penalizing the distance of the SLL of the compromise pattern from a prescribed value.

For comparison purposes, let us consider the same array geometry of [10]. More in detail, the elements are placed on a 20×20 regular grid ($d_x = d_y = \frac{\lambda}{2}$) lying on the xy-plane. The radius of the circular aperture of the antenna is equal to $r = 4.85\,\lambda$. The sum excitations have been fixed to those values affording a circular Taylor pattern [6] with $SLL = -35\,dB$ and $\overline{n} = 6$. Concerning the compromise solution, $Q = 3$ sub-arrays have been considered.

As far as the comparative study is concerned, the final solution of the CPM-based algorithm (i.e., definition of $\underline{c}_{opt}^{H}$ and $\underline{g}_{opt}^{H}$) has been required to present SLR values smaller than those from the SA approach [10]. Since the $PCPM$ is an excitation matching method, it has been iteratively applied by updating the reference difference pattern until the constraints on the compromise solution were satisfied. Accordingly, a succession of reference excitations $\underline{\varsigma}^{H(k)}$, $k = 1, ..., K$ have been selected. In particular, they have been fixed to those of a Bayliss difference pattern [7] with $\overline{n} = 6$ and $SLL_{ref}^{H(k)} = -25\,dB$ $(k = 1)$, $SLL_{ref}^{H(k)} = -30\,dB$ $(k = 2)$, and $SLL_{ref}^{H(k)} = -35\,dB$ $(k = 3)$. The aggregations obtained at the end of each k-th iteration by the $PCPM$ have cost function values equal to $\Psi\left(\underline{c}_{opt}^{H(1)}\right) = 0.65 \times 10^{-1}$, $\Psi\left(\underline{c}_{opt}^{H(2)}\right) = 0.31 \times 10^{-1}$, and $\Psi\left(\underline{c}_{opt}^{H(3)}\right) = 0.27 \times 10^{-1}$, respectively. Although the application of the $PCPM$ could be further iterated by defining others reference targets, the process has been stopped at $k = k_{opt} = 3$ since the requirement $[SLR^{PCPM}(\phi) < SLR^{SA}(\phi),\ 0^o \leq \phi \leq 80^o]$ has been fulfilled by the compromise solution ($\underline{c}_{opt}^{H} = \underline{c}_{opt}^{H(3)}$, $\underline{g}_{opt}^{H} = \underline{g}_{opt}^{H(3)}$). The corresponding relative power distributions are shown in Fig. 4.8 where the solution obtained by Ares *et. al* [10] is reported [Fig. 4.8(a)], as well. To better point out the capabilities of the iterative $PCPM$, also the plots of the SRL values (Fig. 4.9) and the corresponding statistics (Tab. 4.2) are given. Moreover, in order to make the $PCPM$ results reproducible, the sub-array configurations and

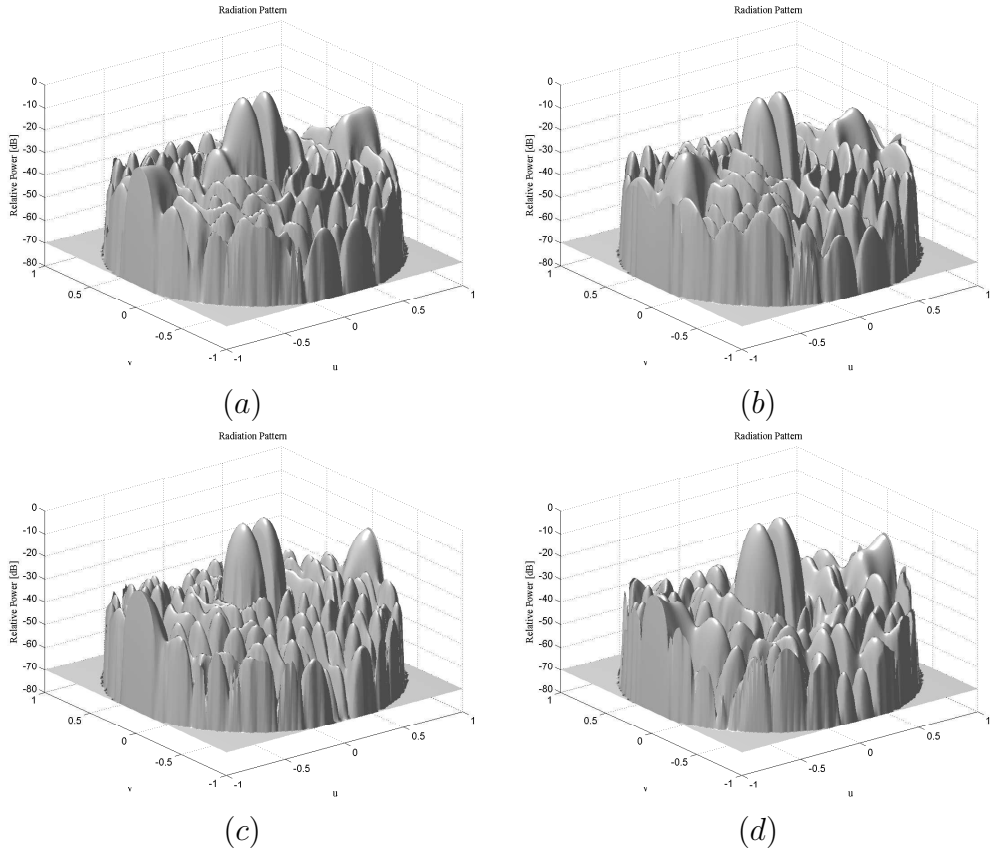


Figure 4.8: *Comparative Assessment* ($N = 300$, $d = \frac{\lambda}{2}$, $r = 4.85\lambda$, $Q = 3$) - Relative power distribution of the $H - mode$ compromise pattern synthesized with (a) the SA approach [10] and the $PCPM$ when the Reference Bayliss pattern $\overline{n} = 6$ [7] presents a sidelobe level equal to (b) $SLL_{ref}^{H(1)} = -25\,dB$, (c) $SLL_{ref}^{H(2)} = -30\,dB$, and (d) $SLL_{ref}^{H(3)} = -35\,dB$.

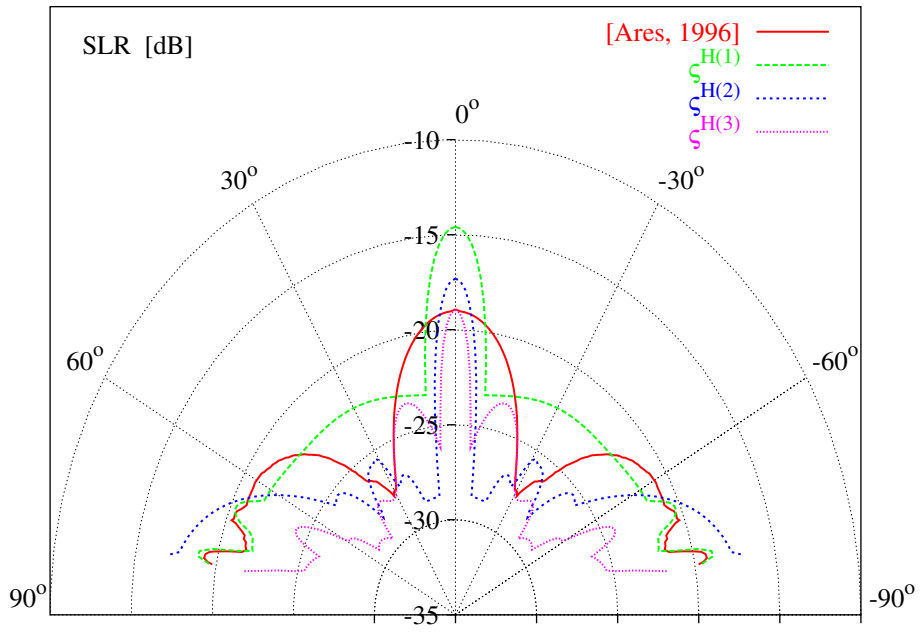


Figure 4.9: *Comparative Assessment* ($N = 300$, $d = \frac{\lambda}{2}$, $r = 4.85\lambda$, $Q = 3$) - Plots of the SLR values of the compromise $H-mode$ difference patterns synthesized by the SA approach [10] and the $PCPM$ when the Reference Bayliss pattern $\overline{n} = 6$ [7] presents a sidelobe level equal to $SLL_{ref}^{H(1)} = -25\,dB$, $SLL_{ref}^{H(2)} = -30\,dB$, and (d) $SLL_{ref}^{H(3)} = -35\,dB$ ($\phi \in [-80^o, 80^o]$).

	$SLL_{ref}^{H(1)} = -25\,dB$	$SLL_{ref}^{H(2)} = -30\,dB$	$SLL_{ref}^{H(k_{opt})} = -35\,dB$
$\underline{c}$	1 1 1 1 1 2 2 1 1 1 2 2 2 1 1 1 2 2 2 3 1 1 1 2 2 2 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 3 3 3 3 3 1 1 1 2 2 3 3 3 3 3	1 1 1 1 1 1 2 1 1 1 2 2 2 1 1 2 2 2 2 2 1 1 2 2 2 3 3 3 1 1 2 2 3 3 3 3 3 1 1 2 2 3 3 3 3 3 1 1 2 2 3 3 3 3 3 1 1 2 2 3 3 3 3 3 3 1 1 2 2 3 3 3 3 3 3	1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 2 1 1 2 2 2 2 2 2 1 1 2 2 3 3 3 2 2 1 1 2 2 3 3 3 3 2 1 1 2 2 3 3 3 3 3 1 1 2 3 3 3 3 3 3 2 1 1 2 3 3 3 3 3 3 2
g_1	0.4668	0.3337	0.3355
g_2	1.3435	0.9763	0.9381
g_3	2.1736	1.6091	1.4469

Table 4.3: *Comparative Assessment* ($N = 300$, $d = \frac{\lambda}{2}$, $r = 4.85\lambda$, $Q = 3$) - Sub-array configurations and weights obtained with the $PCPM$ (Reference Bayliss pattern $\overline{n} = 6$ [7]: $SLL_{ref}^{H(1)} = -25\,dB$, $SLL_{ref}^{H(2)} = -30\,dB$, and $SLL_{ref}^{H(3)} = -35\,dB$).

weights are given in Tab. 4.3. The lists of digits of Tab. 4.3 (second row) indicate the sub-array memberships of the $N = 75$ array elements belonging to a quadrant of the antenna aperture.

Finally, let us analyze the computational issues. The total amount of CPU-time to get the final solution (on a $3.4\,GHz$ PC with $2\,GB$ of RAM) was $T_{tot} = 2.6361\,[sec]$ (i.e., $T^{(1)} = 0.8148\,[sec]$, $T^{(2)} = 0.8302\,[sec]$, and $T^{(3)} = 0.9911\,[sec]$). Moreover, the number of iterations required at each step to synthesize an intermediate compromise solution is equal to $K_{opt}^{(1)} = 14$, $K_{opt}^{(2)} = 14$, and $K_{opt}^{(3)} = 17$, respectively.

Chapter 5

The Ant Colony Optimizer for Graph Searching

Dealing with the proposed excitation matching method, this chapter presents a global optimization strategy for the optimal clustering in sum-difference compromise linear arrays. Starting from a combinatorial formulation of the problem at hand as shown in the previous part of this thesis, the proposed technique is aimed at determining the sub-array configuration expressed as the optimal path inside a directed acyclic graph structure modelling the solution space. Towards this end, an ant colony metaheuristic is used to benefit of its hill-climbing properties in dealing with the non-convexity of the sub-arraying as well as in managing graph searches. A selected set of numerical experiments are reported to assess the efficiency and current limitations of the ant-based strategy also in comparison with the local combinatorial search method previously presented.

5.1 Introduction

In the framework of the optimal matching techniques for the solution of the optimal compromise between sum and difference patterns, besides the methodological and algorithmic novelties introduced in this work, the main result yielded is the proof that the compromise synthesis problem can be formulated as a combinatorial one where the dimension of the solution space grows as a binomial function of the number of array elements (and not exponentially as in classical optimization formulations). Moreover, only the sub-array aggregations are looked for, while the sub-array weights are obtained as a "free by-product". In order to solve the problem at hand, the solution space has been represented through a tree structure where the best compromise solution corresponds to the minimum cost path. Moreover, an *ad-hoc* local search strategy (called *BEM*) has been implemented to effectively sample the solution space. In spite of the good results obtained in pattern matching (Chapter 2), and *SLL* control (Chapter 3), the whole procedure could suffer from a misleading clustering of the array elements that would deeply influence the second step (i.e., the weight computation) since the functional to be optimized is non-convex with respect to the sub-array memberships of the array elements. To avoid this drawback, global optimization is required for solving the clustering step since local searches could get stuck into local minima. However, "standard" evolutionary techniques or general purpose optimizers cannot be adopted because of their computational costs especially when dealing with high-dimension problems and *ad-hoc* algorithms must be used. Accordingly, this paper describes and analyzes the performance of a suitable state-of-the-art evolutionary strategy, namely the Ant Colony Optimizer (*ACO*) [24], whose intrinsic structure seems to be very appropriate to fully exploit a suitable defined graph-like model of the solution space. As a matter of fact, such an approach should in principle avoid the local minima of the cost function because of its *hill climbing* behavior as a global optimizer. On the other hand, it should perform better than other 'physically inspired' optimization algorithms because its intrinsic combinatorial nature able to fully adapt to the description of the solutions as an ensemble of contiguous partitions.

The outline of the chapter is as follows. After a short review of the *BEM* (Sect. 5.2), the *ACO* for graph-searching is carefully described (Sect. 5.3). In Section 5.4, the results of a selected set of numerical experiments are reported in order to firstly describe the *ACO* behavior and then to point out its advantages and best features compared to the *BEM*. Finally, some conclusions are drawn (Sect. 5.5).

5.2 BEM for Graph-Searching

Concerning the notation adopted in the following, it is the same of Chapter (2). There, it has been shown how the solution space of the *contiguous partitions*

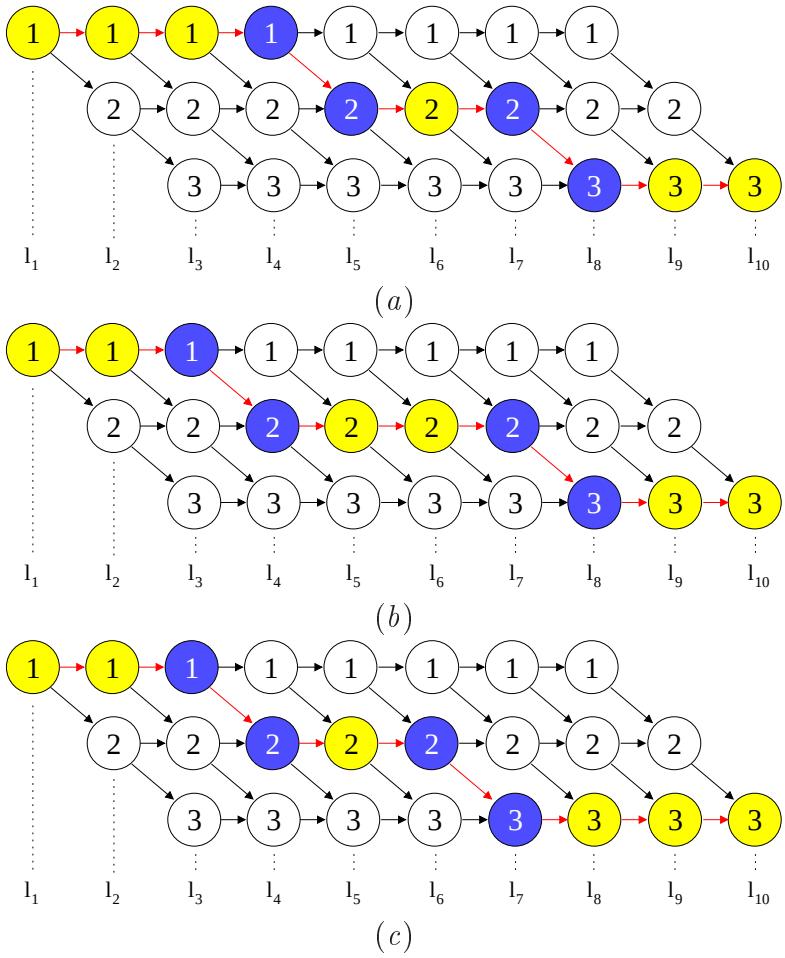


Figure 5.1: Evolution of the *BEM* solution within the *DAG*.

can be represented in an effective fashion through a non-complete binary tree of depth $M-1$, wherein each level of the tree from the root to the leaves defines the sub-array membership for an element of the array. A more compact and non-redundant structure able to give a complete representation of the whole set of admissible sub-array configurations is based on a *Directed Acyclic Graph* (*DAG*) (Chapter 4). As a matter of fact, the non-complete binary tree can be reduced to an equivalent *DAG* by simply noticing that some parts of the tree recursively repeat themselves. Generally speaking, the *DAG* is a graph $G = (\underline{V}, \underline{E})$ composed by a set of $\underline{V}$ vertexes and $\underline{E}$ edges indicated in Fig. 5.1 by circles and arrows, respectively. As regards to the compromise problem, the *DAG* is made of Q rows (i.e., the number of sub-arrays) and $M-Q+1$ vertexes within each row (i.e., the maximum number of elements that can be assigned to a single sub-array by considering non-null clusters). Moreover, the paths inside

the solution graph have the same length[1] equal to $M-1$ and each path codes a trial sub-array configuration $\underline{C}$.

In order to explore the solution graph looking for the path minimizing (2.2), the *Border Element Method* (*BEM*) dealing with a tree architecture is adapted here to work with the *DAG*, as well. Accordingly, the so-called *border elements* are now those elements of the actual configuration/path whereof at least one closest element of the path belongs to a different row of the *DAG* (i.e., it is assigned to a different sub-array). For sake of clarity and with reference to Fig. 5.1, the cluster configurations are indicated by the red edges and the border elements are denoted by the blue vertexes. It is worth notice that it is possible to obtain a new admissible trial aggregation $\underline{C}'$ just changing the membership of a border element as schematically described in the following pseudo-code that summarizes the *BEM* for the sampling of the *DAG* structure:

```
Compute v_m : m = 1,...,M
Sort v_m : m = 1,...,M to obtain L: l_1 = min{v_m}, l_M = max{v_m}
Initialize C^(0) = { c_m^(0) ; m = 1,...,M : c_m^(0)(l_1) = 1, c_m^(0)(l_M) = Q }
for each iteration k : k = 1,...,K

    C^(k) = C^(k-1)
    c_m^(k)(l_1) = c_m^(0)(l_1), c_m^(k)(l_M) = c_m^(0)(l_M)
    for each element l_h : h = 2,...,M-1

        if (c_m^(k)(l_h) is a border element) AND (is not a single element sub-array) then

            Assign c_m^(k) the membership of the closer/different sub-array to obtain C'
        End if
        Calculate Fitness of C'
        if (Ψ{C'} < Ψ{C^(k)}) then

            New solution found: C^(k) = C'
        End if
    End for
    if (Ψ{C^(k)} stationary) then

        C^BEM = C^(k); k_end = k
        Stop
    End if

End for
C^BEM = C^(K); k_end = K
Stop
```

More in detail, the *BEM* is first aimed at looking for the border elements of the current path $\underline{C}^{(k)}$ belonging to the *DAG* and successively at changing their memberships (once a time), until a termination criterion based on a maximum number of iterations K ($k = 0, ..., K$; k being the iteration index) or on a stationary condition of the cost function value $\Psi\left\{\underline{C}^{(k)}\right\}$ is reached. For illustrative purposes, a pictorial representation of the *BEM*-based searching is given in Fig. 5.1. It is concerned with the test case characterized by $M = 10$ and $Q = 3$.

[1]The length of a *DAG* is equal to the number of edges of the longest directed path.

Starting from the guess solution $\underline{C}^{(0)}$ displayed in Fig. 5.1(a), the iterative process stops after two iterations determining the final aggregation $\underline{C}^{BEM} = \underline{C}^{(2)}$ shown in Fig. 5.1(c).

5.3 ACO for Graph-Searching

From the *BEM* pseudo-code, it is simple to recognize that such a method, for both tree and graph-like architectures, is a deterministic technique that suffers of the usually standard drawbacks of local search algorithms. In particular, the *BEM* solution might be trapped in a local minimum and strongly influenced by the starting guess aggregation $\underline{C}^{(0)}$ chosen at the initialization because of the non-convexity of the problem at hand.

In order to overcome the problems related to the presence of local minima in the cost function (2.3), the *Ant Colony Optimizer* (*ACO*) is adopted here to search for the optimal path $\underline{C}^{opt}$ within the solution graph that minimizes (2.3). The *ACO* is a global optimization algorithm inspired by the foraging behavior of ant colonies looking for food sources [24]. The ants look for the shortest path between the food sources and the nest. Towards this end, each ant leaves a chemical substance, called *pheromone*, while moving in the space surrounding the nest. The amount of pheromone on a path quantifies its degree of optimality, but it decays with time (*evaporation* mechanism). These mechanisms allow one to avoid poor food sources on one hand (*pheromone release*) and on the other, to efficiently sample the whole solution space (*pheromone evaporation*).

The *ACO* developed by *Dorigo* [25] has been widely applied especially in distributed and discrete problems such as routing [26][27], assignment [28][29], scheduling [30][31], subset [32], but it is relatively infrequent in electromagnetics. To the best of authors' knowledge, it has been recently applied to few electromagnetic problems (e.g., antenna synthesis considering binary [33] or real implementations [34][35][36] and microwave imaging [37]). However, because of its effectiveness in facing hard combinatorial problems and since the combinatorial formulation of the optimal compromise between sum and difference patterns requires the searching of the best path within a graph, the *ACO* seems to be a suitable metaheuristic for the problem at hand. Towards this aim, the simplest version of the *ACO*, namely *Ant System* [24], is used. The proposed *ACO* implementation is customized to the graph architecture to properly address the synthesis of small as well as large arrays. As a matter of fact, due to the high number of vertexes needed for the storage of the solution, applying the *ACO* to the search within the solution-graph presents some memory limitations when dealing with very large dimensional spaces. On the other hand, it must be pointed out that the *ACO* performances in terms of solution accuracy do not depend on the representation of the solution space, but only the feasibility and the computational indexes (i.e., the storage resources and the rate of sampling the solution space) are affected by the architecture at hand.

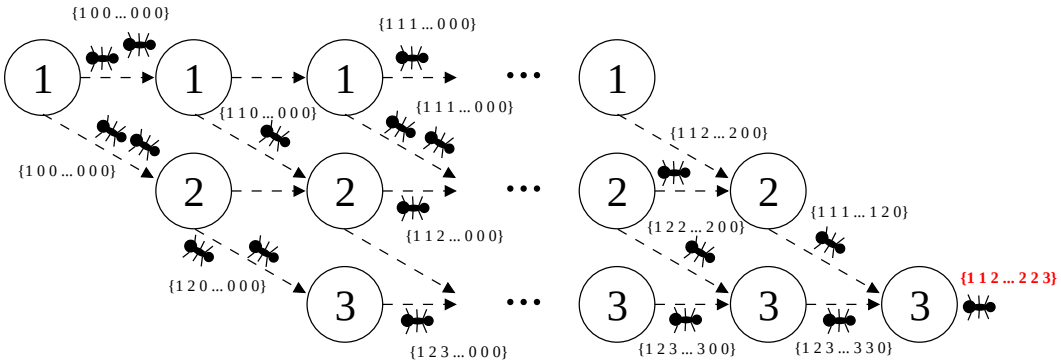


Figure 5.2: Evolution of the *ACO* solution within the *DAG*.

The proposed implementation of the *ACO*-based approach can be summarized as follows. Each i-th ($i = 1, ..., I$) ant codes a vector $\underline{a}_i$ of M integer values that models a trial sub-array configuration $\underline{C}_i$ (i.e., $\underline{a}_i = \underline{a}\{\underline{C}_i\}$). Every vector is initialized to the null one at each iteration (i.e., $\underline{a}_i^{(k)} = \{0, ..., 0\}$, $k = 1, ..., K$ and $i = 1, ..., I$) and it is filled step-by-step while the ants are moving through each level of the graph as shown in Fig. 5.2 . At the initialization ($k = 0$), the quantity of pheromone on each edge $\tau^{(0)}(e_z^r)$, $e_z^r = 1, ..., E$ is the same and each edge of the graph can be explored with a uniform probability $p^{(0)}(e_z^r) = 0.5$. As regards to the apex r, it is equal to $q \rightarrow q$ if the edge e_z^r connects two vertexes belonging to the same sub-array (i.e., the same row of the *DAG*) and to $q \rightarrow q+1$ if it connects two vertexes assigned to different sub-arrays (i.e., different rows of the *DAG*). Moreover, the pedex z, $z = z_1, ..., z_{M-1}$, identifies the level of the edge within the graph. Concerning the iterative loop ($k > 0$), the probability of choosing one of the two subsequent edges (if present) at each vertex is given by

$$p^{(k)}(e_z^r) = \frac{\tau^{(k)}(e_z^r)}{\tau^{(k)}(e_z^{q \rightarrow q}) + \tau^{(k)}\left(e_z^{q \rightarrow q+1}\right)}, \quad z = z_1, ..., z_{M-1}; \; r = q \rightarrow q + [0, 1]. \tag{5.1}$$

When the whole ant colony has completed a path within the *DAG*, the pheromone level $\tau^{(k)}(e_l^r)$ of each edge is updated as follows

$$\tau^{(k+1)}(e_z^r) \leftarrow \tau^{(k)}(e_z^r) + \sum_{i=1}^{I} \delta_{e_z^r \underline{a}_i^{(k)}} \frac{H}{\Psi\left(\underline{C}_i^{(k)}\right)}, \quad \forall \tau^{(k)}(e_z^r) \tag{5.2}$$

where $\delta_{e_z^r \underline{a}_i^{(k)}} = 1$ when $e_z^r \in \underline{a}_i^{(k)}$ $\left[\underline{a}_i^{(k)} = \underline{a}\left\{\underline{C}_i^{(k)}\right\}\right]$ and $\delta_{e_z^r \underline{a}_i^{(k)}} = 0$ otherwise, H being a positive constant. Successively, the evaporation procedure takes place in order to reduce and at most delete worse paths from the graph

$$\tau^{(k+1)}(e_z^r) \leftarrow (1 - \rho)\, \tau^{(k+1)}(e_z^r), \quad \forall \tau^{(k+1)}(e_z^r) \tag{5.3}$$

$\rho \in (0\ ,\ 1]$ being a parameter aimed at controlling the evaporation rate. Finally, the same stopping criterion ($k = k_{end}$) used for the BEM is adopted here for the ACO-based method to allow fair comparisons.

5.4 Numerical Simulations and Results

Because of the novelty of the proposed approach, the first part of this section (Sect. 5.4.1) is devoted to the calibration of the ACO algorithm [38] when dealing with the searching of the "best compromise" solution among those admissible within the solution graph. Successively, the use of the ACO is motivated (Sect. 5.4.2) showing how the BEM solution suffers from the non-convexity of the aggregation problem because of the local nature of the algorithm. Finally, a set of comparative results concerned with a wide number of compromise problems are reported (Sect. 5.4.3) to point out potentialities and current limitations of the ACO-based approach.

5.4.1 *ACO* Calibration

A key feature of the ACO algorithm is the simple implementation. As a matter fact, besides the number I of ants in the colony, it only requires the definition of two parameters to work, namely the pheromone update coefficient H and the pheromone evaporation coefficient ρ. In order to determine their optimal values for the problem at hand, an extensive set of numerical experiments has been carried out by considering an array of $N = 40$ elements and $Q = 6$ sub-arrays as reference benchmark. In this case, the number of contiguous partitions is equal to $U^{(ess)} = \begin{pmatrix} 19 \\ 5 \end{pmatrix} = 11628$. As far as the reference excitations are concerned, those affording a Dolph-Chebyshev sum pattern with $SLL = -25\,dB$ [19] and a Zolotarev difference pattern with $SLL = -30\,dB$ [9] have been chosen. Concerning the calibration study, the values of the ACO control coefficients have been varied in the range $H \in [0 : 5]$ and $\rho \in (0 : 1]$ [24], respectively. Moreover, because of the stochastic nature of the ACO algorithm, 100 different simulations have been performed for each setting of the calibration parameters. Each simulation has been run with a number of ants equal to $I = [3,\ 5,\ 8,\ 10,\ 100,\ 1000]$ for a maximum number of $K = 1000$ iterations.

As a representative result, the average performances for each parameter configuration when $I = 3$ are reported in Fig. 5.3 . As it can be observed, the convergence cost function value is more sensitive to the evaporation coefficient ρ and less to the value of the parameter H that controls the pheromone update. A similar conclusion holds true whatever the value of I. Concerning the optimal setup, the configuration $H = 1$ and $\rho = 0.05$ has been selected since the corresponding representative point in Fig. 5.3 lies in the lowest region and the value

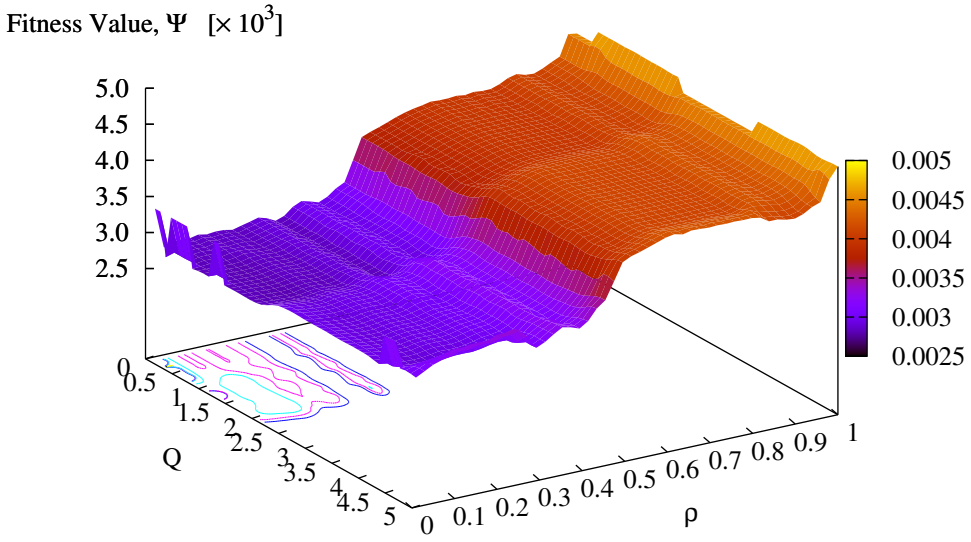


Figure 5.3: *ACO Calibration* ($N = 40$, $Q = 6$) - Behavior of the average convergence cost function value versus the pheromone update constant, H, and the pheromone evaporation parameter, ρ.

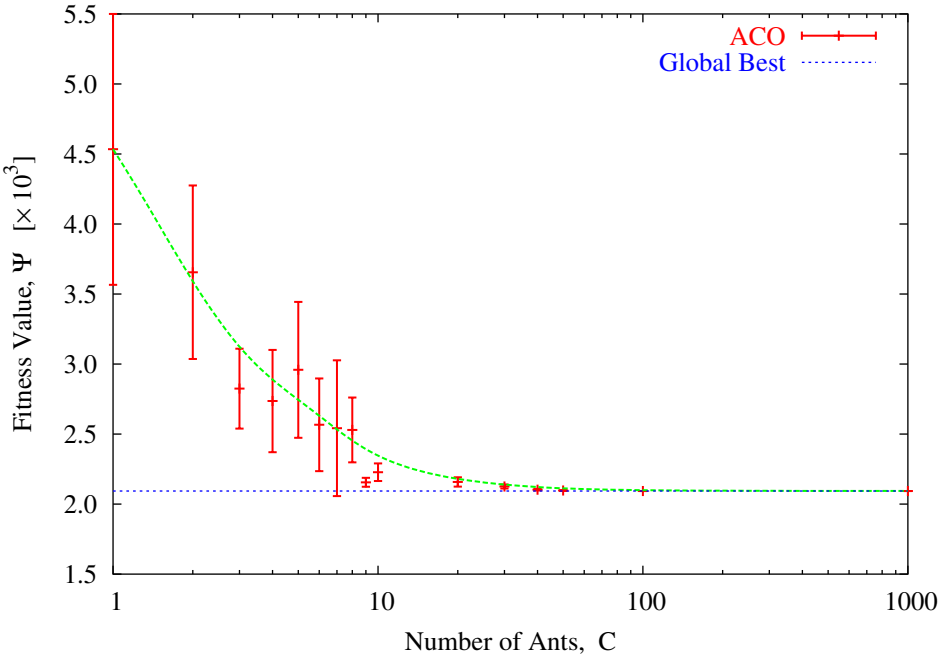


Figure 5.4: *ACO Calibration* ($N = 40$, $Q = 6$; $H = 1$, $\rho = 0.05$) - Behaviors of the statistic values of the average convergence cost function value versus the ant colony dimension, C.

$H = 1$ has already been identified as an optimal choice in other graph searching problems (e.g., TSP [26]).

As regards to the dimension of the ant colony, the analysis has been devoted to define the optimal value of I in relationship to the dimension of the solution space $U^{(ess)}$. Towards this end, I has been varied between 1 and $\frac{1}{10}U^{(ess)}$. Figure 5.4 shows the results of the statistical study, each cross being the average Ψ among the values reached at the end of each group of 100 simulations. For completeness, the standard deviation is shown, as well. From these results, it can be inferred that the choice $I \simeq \left[\frac{1}{125}U^{(ess)} : \frac{1}{100}U^{(ess)}\right]$ defines a good rule of thumb to reach the global solution with a percentage above 90% [2]. On the other hand, the minimum value of $I_{lb} = 5$ ants has been set as lower bound in order to exploit the cooperative behavior of the ACO in those problems where the previous criterion would give too small values (i.e., $I < I_{lb}$).

[2]It is worth noting that the results here reported have been obtained under the assumption of a maximum number of iterations equal to $K = 1000$. Probably, increasing the number of iterations would allow a reduction of the number of ants for obtaining the same conclusions.

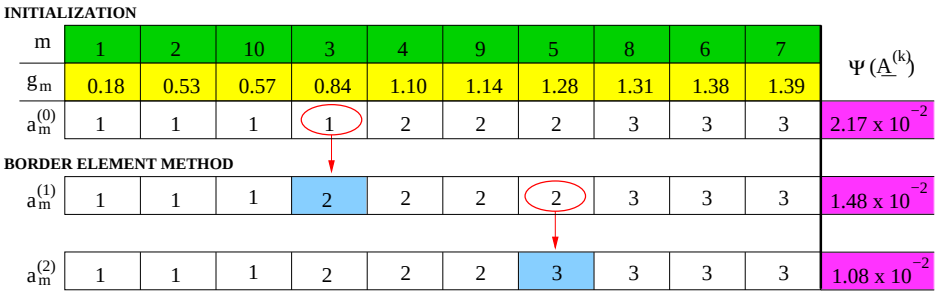


Figure 5.5: *ACO's Hill Climbing Behavior* ($N = 20$, $Q = 3$) - Iterative *BEM* procedure.

5.4.2 *ACO*'s Hill-Climbing Behavior

In order to show how the performance of the *BEM* are influenced from the choice of the initial solution, while the *ACO* is not dependent on the starting guess and therefore more robust to the local minima problem thanks to its hill-climbing properties, three samples of compromise syntheses concerned with small as well as larger arrays for different number of sub-arrays are discussed in the following.

The first experiment deals with a 20-elements array ($M = 10$) with inter-element spacing $d = \frac{\lambda}{2}$. The optimal sum and difference coefficients have been chosen to afford a Dolph-Chebyshev sum pattern with $SLL = -25\,dB$ [19] and a Zolotarev difference pattern with $SLL = -30\,dB$ [9], respectively. As regards to the compromise feed network, $Q = 3$ sub-arrays have been used.

Concerning the *Contiguous Partition Method* (*CPM*) customized in the present work to the searching within the solution graph, the optimal gains v_m, $m = 1, \ldots, M$, are first computed as described in Chapter (2) and then sorted on a line in order to obtain the list $\underline{L} = \{l_h : l_h \leq l_{h+1},\, h = 1, ..., M-1\}$, where $l_1 = min\{v_m\}$ and $l_M = max\{v_m\}$. Each element of the sorted list $\underline{L}$ is assigned to a level of the solution graph as shown in Fig. 5.1. Starting from a uniform sub-arraying (i.e., a sub-array configuration wherein the number of elements within each sub-array differs at most of one element when M is or not a multiple of Q), the initial sub-array vector turns out to be $\underline{C}^{(0)} = \{1\,1\,1\,2\,2\,3\,3\,3\,2\,1\}$ (Fig. 5.5) . Then, the iterative loop of the *BEM* takes place according to the pseudo-code of Sect. (5.2) and as detailed in Fig. 5.5. For completeness, Figure 5.1 shows the corresponding evolution of the *BEM* trial solution in the solution graph. As it can be noticed, the *BEM* gets stuck only after $k_{end}^{BEM} = 2$ iterations. The final grouping is $\underline{C}^{BEM} = \underline{C}^{(2)} = \{1\,1\,2\,2\,3\,3\,3\,3\,2\,1\}$ [Fig. 5.1(c)] with a convergence fitness value of $\Psi\left(\underline{C}^{BEM}\right) = 1.08 \times 10^{-2}$, while the intermediate solution $\underline{C}^{(1)} = \{1\,1\,2\,2\,2\,3\,3\,3\,2\,1\}$ [Fig. 5.1(b)] has a fitness equal to $\Psi\left(\underline{C}^{(1)}\right) = 1.48 \times 10^{-2}$. The radiation patterns generated at the various iterations and the reference pattern are reported in Fig. 5.6, as well.

Successively, the *ACO* has been applied to the same test case. Since the

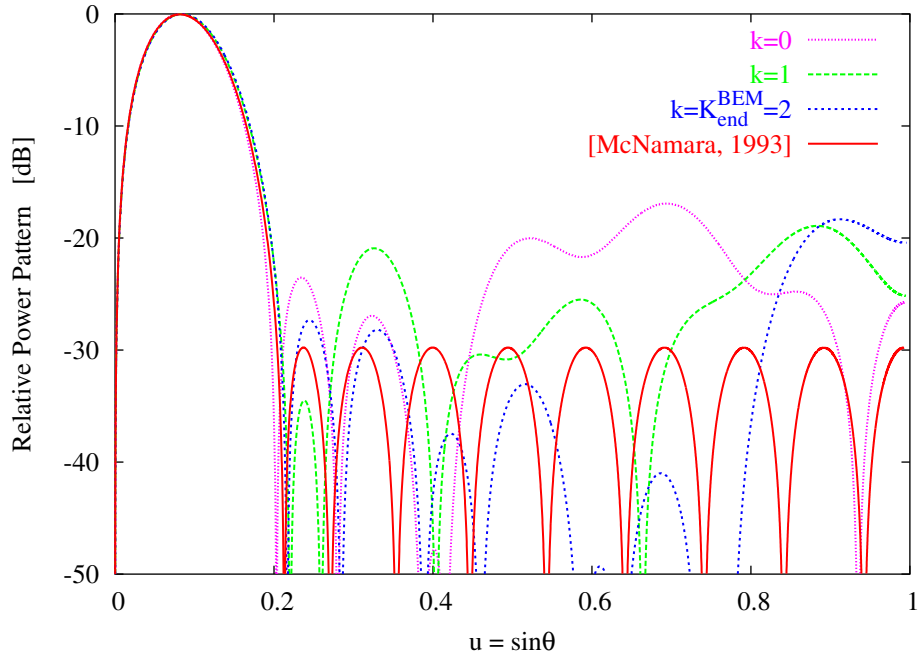


Figure 5.6: *ACO's Hill Climbing Behavior* ($N = 20$, $Q = 3$) - *BEM* power pattern at different iterations of the iterative optimization ($k = 1, ..., k_{end}$).

number of trial solutions within the graph is equal to $U^{(ess)} = \begin{pmatrix} 9 \\ 2 \end{pmatrix} = 36$ and I, according to the criterion previously defined, would result lower than one, the *ACO* population has been set to $I = I_{lb} = 5$. Moreover, the pheromone update H and the evaporation ρ have been fixed to their optimal values. As expected, the *ACO* outperforms the *BEM* since the fitness value of the synthesized solution $\underline{C}^{ACO} = \{1\,2\,2\,3\,3\,3\,3\,3\,3\,2\}$ is equal to $\Psi\left(\underline{C}^{ACO}\right) = 8.26 \times 10^{-3}$ [vs. $\Psi\left(\underline{C}^{BEM}\right) = 1.08 \times 10^{-2}$]. To further confirm the *ACO* effectiveness, it is worth noting that the clustering determined by the *ACO* is the one having the minimum fitness among the $U^{(ess)} = 36$ admissible different clustering. On the contrary, the *BEM* has been able to retrieve the second best solution coded into the solution graph as shown in Fig. 5.7 (red line) where each cross denotes the Ψ value among the $U^{(ess)} = 36$ contiguous partitions ranked according to their cost function values. More specifically, the *BEM* solution is evidenced with a circle, while the minimum fitness value or global minimum of the excitation matching cost function coincides with the *ACO* clustering [i.e., $\Psi^{opt} = \Psi\left(\underline{C}^{ACO}\right)$]. On the other hand, it is also interesting to point out that, even though the *BEM* solution is the second best compromise, it has three elements over ten whose sub-array memberships are different from those of the global optimum $\underline{C}^{opt}$ recognized by the *ACO*-based algorithm, $\underline{C}^{ACO} = \underline{C}^{opt}$.

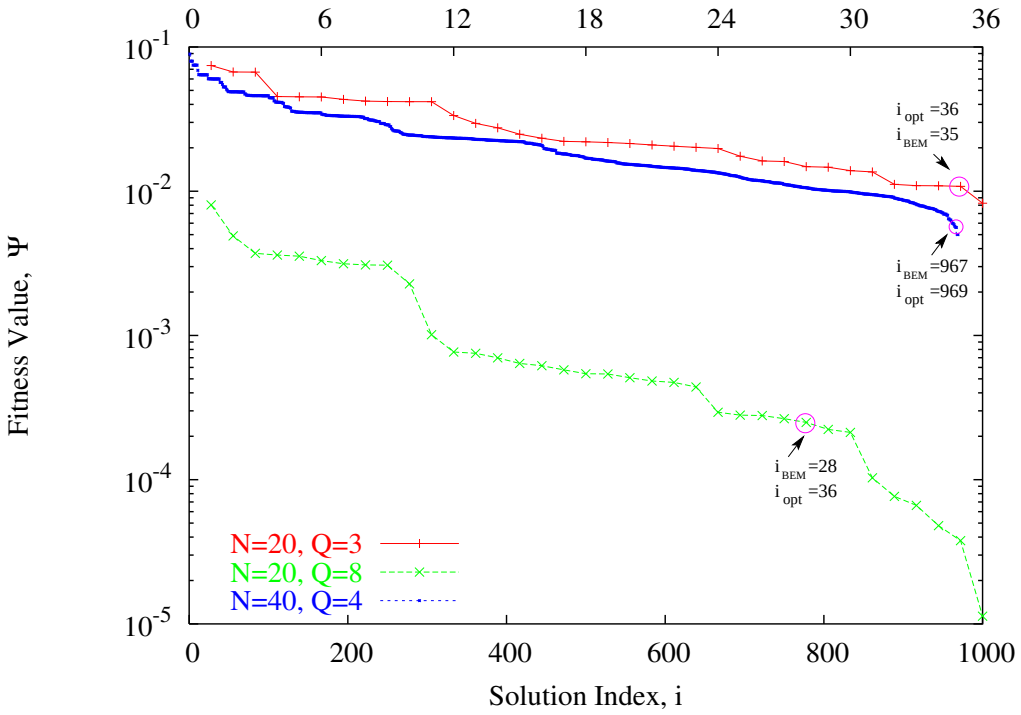


Figure 5.7: *ACO's Hill Climbing Behavior* - Cost function values of the solutions coded in the solution *DAG*.

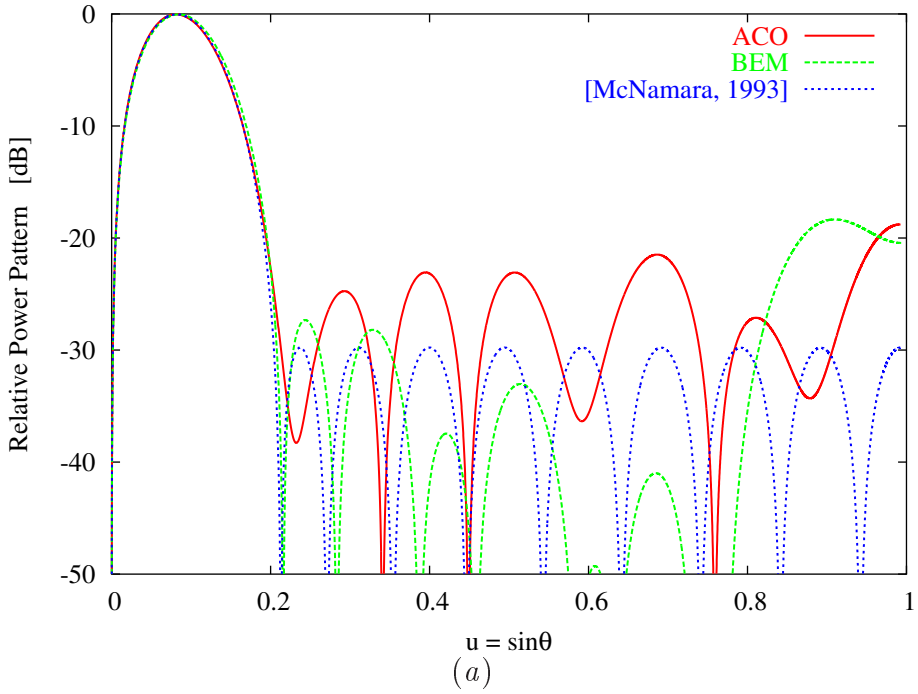
$M = 10$	a_m^{BEM}	$\{1\,1\,2\,2\,3\,3\,3\,3\,2\,1\}$		
	a_m^{ACO}	$\{1\,2\,2\,3\,3\,3\,3\,3\,3\,2\}$		
$Q = 3$	w_q^{BEM}	0.3827	0.9736	1.3363
	w_q^{ACO}	0.1798	0.6602	1.2549

Table 5.1: *ACO's Hill Climbing Behavior* ($N = 20$, $Q = 3$) - Sub-array configurations and weights determined by the BEM and the ACO.

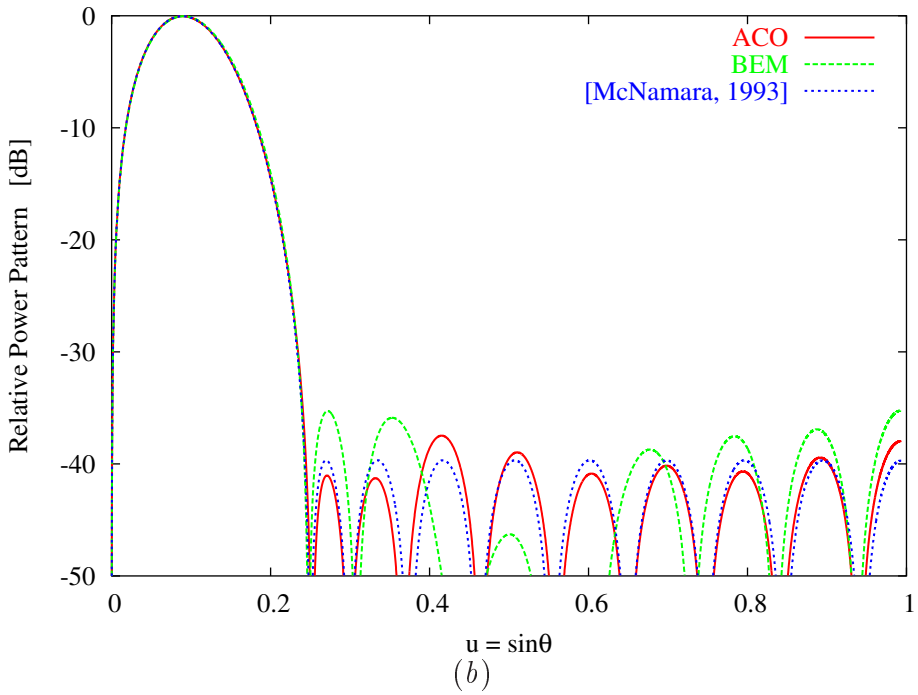
Approach	Ψ_{opt}	Δ	$SLL\,[dB]$	$BW\,[deg]$	k_{end}	F_{end}	$t\,[sec]$	$T^{(ess)}$
$N = 2M = 20\,,\, Q = 3$								
BEM	1.08×10^{-2}	0.3199	-18.25	5.28	2	3	$< 10^{-8}$	36
ACO	8.26×10^{-3}	0.2689	-18.75	5.12	2	10	$< 10^{-8}$	36
$N = 2M = 20\,,\, Q = 8$								
BEM	2.49×10^{-4}	0.0545	-35.20	5.74	2	3	$< 10^{-8}$	36
ACO	1.13×10^{-5}	0.0145	-37.50	5.68	2	10	$< 10^{-8}$	36
$N = 2M = 40\,,\, Q = 4$								
BEM	5.60×10^{-3}	0.2886	-20.10	2.50	21	22	$< 10^{-7}$	969
ACO	4.99×10^{-3}	0.2609	-22.85	2.50	34	340	4.5×10^{-3}	969

Table 5.2: *ACO's Hill Climbing Behavior* - Pattern performances and computational indexes.

For completeness, Table 5.1 details the results obtained with the BEM and the ACO by reporting the final sub-array configurations and the gain values. Moreover, the synthesized difference compromises are shown in Fig. 5.8(a) . Because of the excitation-matching nature of the proposed technique, let us quantify the closeness of the arising patterns with respect to the optimal/reference one by computing the *pattern matching* Δ (2.6). As expected and indicated by the corresponding lower fitness value, the ACO pattern is closer to the reference one. As a matter of fact, it is $\Delta^{ACO} = 0.2689$ vs. $\Delta^{BEM} = 0.3199$ (Tab. 5.2) . Table 5.2 also reports the values of other indexes in order to give a complete overview of the features of the obtained patterns (i.e., sidelobe level, SLL, and main lobe width, Bw). Moreover, the computational issues are pointed out by the following indexes: the number of convergence iterations, k_{end}, the number of function evaluations, F_{end}, and the CPU-time t necessary to find $\underline{C}^{(k_{end})}$ on a $3.4\,GHz$ PC with $2\,GB$ of RAM. As it can be noticed, both BEM and ACO are able to find a convergence solution almost in real time since $t < 10^{-8}$. Such an event points out once again the computational efficiency of the CPM approach, but also the usefulness of the graph representation that enables the use of an evolutionary algorithm without excessively increasing the computational costs and memory resources.



(a)



(b)

Figure 5.8: *ACO's Hill Climbing Behavior* - Compromise difference power patterns obtained with the *BEM* and the *ACO* when (a) $N = 20$, $Q = 3$ (Zolotarev [9], $SLL = -30\,dB$) and (b) $N = 20$, $Q = 8$ (Zolotarev [9], $SLL = -40\,dB$).

$M = 10$	a_m^{BEM}	$\{1\,2\,3\,5\,7\,8\,6\,4\,2\,1\}$							
	a_m^{ACO}	$\{1\,3\,5\,7\,8\,8\,7\,6\,4\,2\}$							
$Q = 8$	w_q^{BEM}	0.2146	0.6107	0.9221	0.9825	1.1582	1.1797	1.2818	1.2864
	w_q^{ACO}	0.2049	0.2432	0.5937	0.7250	0.9221	0.9825	1.1650	1.2838

Table 5.3: *ACO's Hill Climbing Behavior* ($N = 20$, $Q = 8$) - Sub-array configurations and weights computed with the BEM and the ACO.

In the second experiment, the same array geometry of the previous example has been considered, but the array has been partitioned into $Q = 8$ sub-arrays. Moreover, a Zolotarev difference pattern with $SLL = -40\,dB$ [9] has been adopted as reference target. It is worth observing that despite the higher number of sub-arrays, the dimension of the solution space is still equal to $U^{(ess)} = 36$ thanks to the symmetric nature of the binomial distribution [i.e., $U^{(ess)} = \begin{pmatrix} 9 \\ 7 \end{pmatrix} = \begin{pmatrix} 9 \\ 2 \end{pmatrix} = 36$]. Analogously to the previous example, the BEM stops after $k_{end}^{BEM} = 2$ iterations synthesizing the solution in Tab. 5.3 , but in this case other 8 solutions with lower fitness values are present in the solution graph (Fig. 5.7 - green line). On the other hand, the ACO has been able to reach the global optimum in Tab. 5.3 after $k_{end}^{ACO} = 2$ iterations with a total number of fitness evaluation equal to $F_{end}^{ACO} = 10$ since $I = I_{lb} = 5$. In particular, the ACO solution presents a fitness value of more than one order in magnitude below the one of the BEM [i.e., $\Psi\left(\underline{C}^{ACO}\right) = 1.13 \times 10^{-5}$ vs. $\Psi\left(\underline{C}^{BEM}\right) = 2.49 \times 10^{-4}$] and $\frac{\Delta^{BEM}}{\Delta^{ACO}} \simeq 3.76$ as it can be qualitatively observed by comparing the patterns in Fig. 5.8(b). For the sake of completeness, Table 5.2 compares the retrieved solutions in terms of performance indexes.

The last experiment of this section is concerned with a larger uniform array of 40 $\frac{\lambda}{2}$-spaced elements. A Dolph-Chebyshev sum pattern with $SLL = -25\,dB$ [19] and a Zolotarev difference pattern with $SLL = -30\,dB$ [9] have been chosen as reference patterns and the number of sub-arrays has been set to $Q = 4$. In such a case, the number of possible sub-array configuration within the solution space is equal to $U^{(ess)} = 969$. As far as the ACO is concerned, $I = 10$ ants have been used. The two approaches have found the corresponding solutions after $k_{end}^{BEM} = 21$ and $k_{end}^{ACO} = 34$ as shown in Fig. 5.9 where the behavior of the cost function during the iterative searching process for both the BEM and the ACO is described. The synthesized sub-array configurations and weights are given in Tab. 5.4 , whereas the corresponding patterns are displayed in Fig. 5.10 . As expected and likewise to the previous experiments, the BEM is still trapped into a local minimum and the retrieved solution turns out to be sub-optimal. However, it should be observed (Fig. 5.7 - blue line) that the BEM configuration is the third best contiguous partition among $U^{(ess)} = 969$ different solutions and the value of the ratio $\frac{\Delta^{BEM}}{\Delta^{ACO}} \simeq 1.11$ assesses its closeness to the optimal one. As

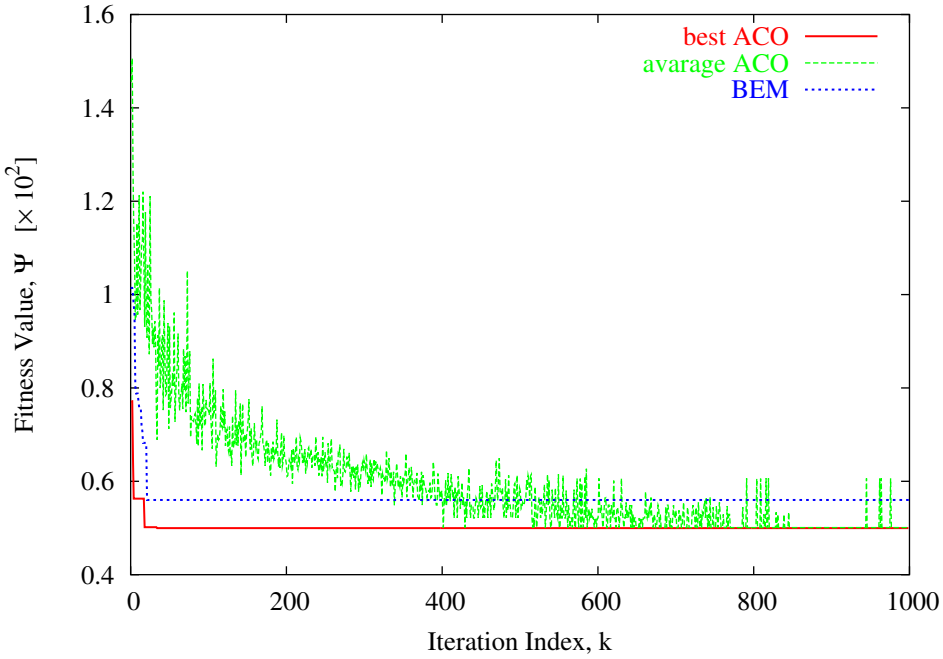


Figure 5.9: *ACO's Hill Climbing Behavior* ($N = 40$, $Q = 4$) - Behavior of the cost function value $\Psi^{(k)}$ during the iterative optimization process when applying the *BEM* and the *ACO*.

$M = 20$	a_m^{BEM}	$\{1\,1\,2\,2\,2\,3\,3\,3\,4\,4\,4\,4\,4\,4\,4\,4\,4\,3\,3\,2\}$			
	a_m^{ACO}	$\{1\,1\,2\,2\,3\,3\,3\,4\,4\,4\,4\,4\,4\,4\,4\,4\,4\,3\,2\}$			
$Q = 4$	w_q^{BEM}	0.1779	0.5658	1.0257	1.3288
	w_q^{ACO}	0.1779	0.5055	0.8989	1.2923

Table 5.4: *ACO's Hill Climbing Behavior* ($N = 40$, $Q = 4$) - Sub-array configurations and weights synthesized by means of the *BEM* and the *ACO*.

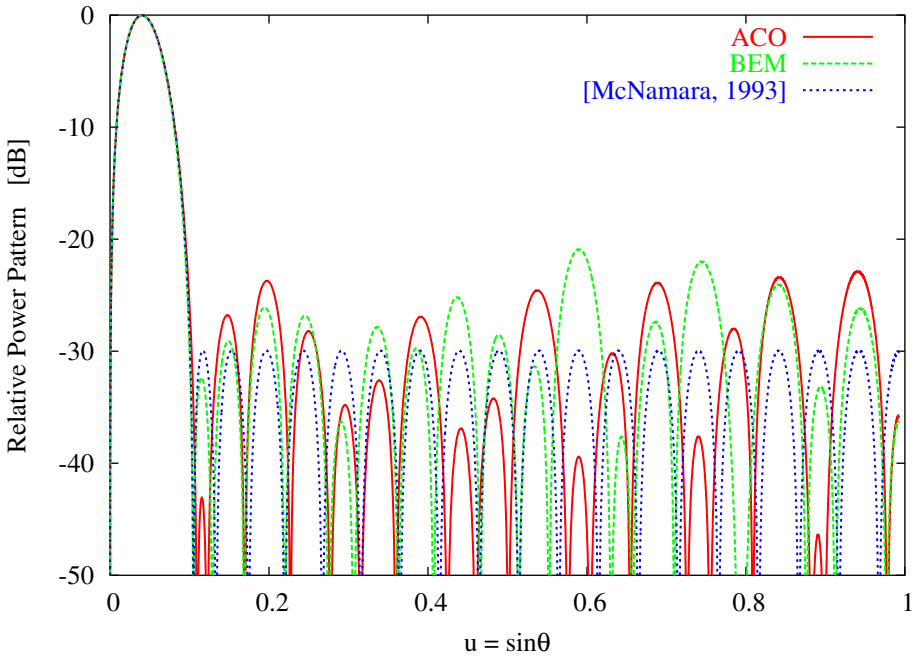


Figure 5.10: *ACO's Hill Climbing Behavior* ($N = 40$, $Q = 4$) - Reference (Zolotarev [9], $SLL = -30\,dB$) and compromise difference power patterns synthesized with the BEM and the ACO.

regards to the computational issues, such a test further confirms the efficiency of the BEM (in terms of speed) in exploring the solution space being $t^{BEM} < 10^{-7}$ while $t^{ACO} = 4.5 \times 10^{-3}$. As a matter of fact, although the CPU-time required by the ACO-based approach is certainly smaller than that of standard global optimizers, it cannot be omitted that from a computational point of view the BEM results more competitive than the ACO when the ratio $\frac{M}{Q}$ gets larger and larger. Such a statement will be further analyzed in the following section.

5.4.3 *ACO*'s Performances and Problem Dimensions

In dealing with the optimal compromise between sum and difference patterns, different global optimization techniques have been applied to determine the most suitable partition of the array elements into sub-arrays that minimizes a suitable cost function related to some pattern features. Among them, it is worth mentioning the *Genetic Algorithm* [12], the *Differential Evolution Algorithm* [11] and its enhanced version [15], and the *Simulated Annealing* [14]. Despite the different way of tackling the problem at hand (i.e., direct optimization of element memberships and weights [12][11][15] or two-step nested approach [14] exploiting functional convexity), the dimension of the solution space to be explored for retrieving the elements aggregation is equal to $U^{(tot)} = Q^M$ since each clustered configuration can be expressed as a string of M digits in a Q-based notation system. Let us now suppose to use in a standard fashion (i.e., without reformulating the problem at hand as a combinatorial one) a global optimizer and to apply the rule deduced in Sect. (5.4.1) for the population size [i.e., $I^{(tot)} \simeq 10^{-2} \times T^{(tot)}$] for running a simulation in a fixed number of iterations $\hat{K}$ looking for the optimal aggregation within the set of $I^{(tot)}$ possible solutions. The total CPU time necessary to complete such a simulation turns out be $\Delta t^{(tot)} = \delta t \times \hat{K} \times I^{(tot)}$, δt being the CPU-time for one evaluation of the cost function. Moreover, it should be pointed out that there is not guarantee that the synthesized aggregation is the global optimum of the functional at hand. Then, let us refer to the combinatorial formulation of the compromise problem and map the reduced solution space of dimension $I^{(ess)}$ into the graph representation described in Sect. 5.3. By exploiting such a structure and accordingly using the proposed implementation of the ACO, the number of ants of the colony turns out to be $I^{(ess)} \simeq 10^{-2} \times T^{(ess)}$ much smaller than $I^{(tot)}$ since $U^{(ess)}$ grows at most polynomially [i.e., $U^{(ess)} = \begin{pmatrix} M-1 \\ Q-1 \end{pmatrix}$] and not exponentially as $U^{(tot)}$ [$U^{(tot)} = Q^M$]. Therefore, the iterative optimization runs for a time $\Delta t^{(ess)} = \delta t \times \hat{K} \times I^{(ess)}$, which satisfies the following condition $\Delta t^{(ess)} \ll \Delta t^{tot}$[3] since $I^{(ess)} \ll I^{(tot)}$. Such

[3]For the sake of simplicity, δt has been assumed to be equivalent for both standard and combinatorial optimizations. However, please also consider that $\delta t^{(ess)} < \delta t^{(tot)}$ since usually $\delta t^{(tot)}$ requires the computation of a pattern feature, while $\delta t^{(ess)}$ is related to a matching operation.

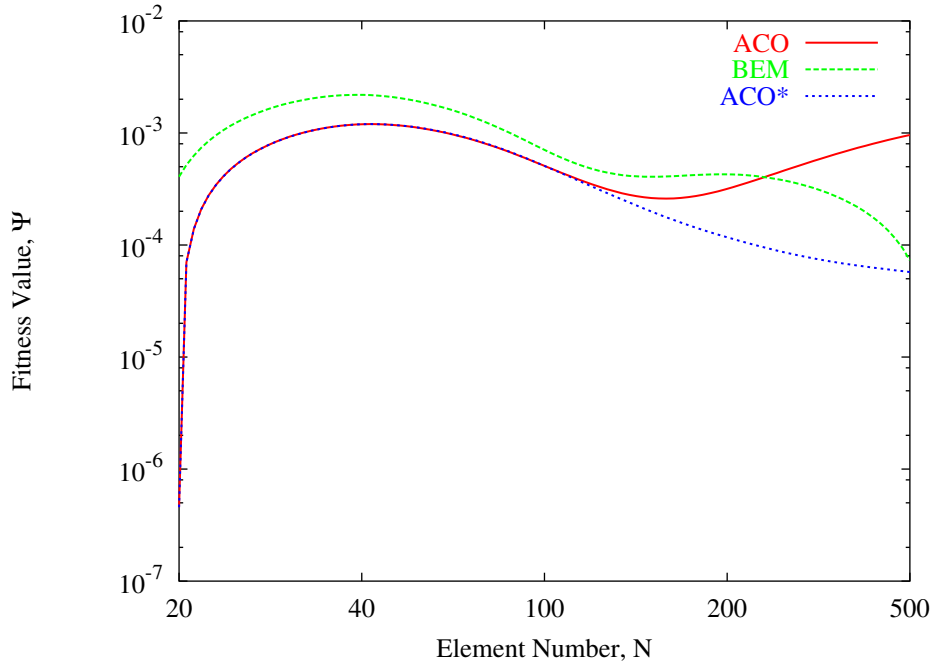
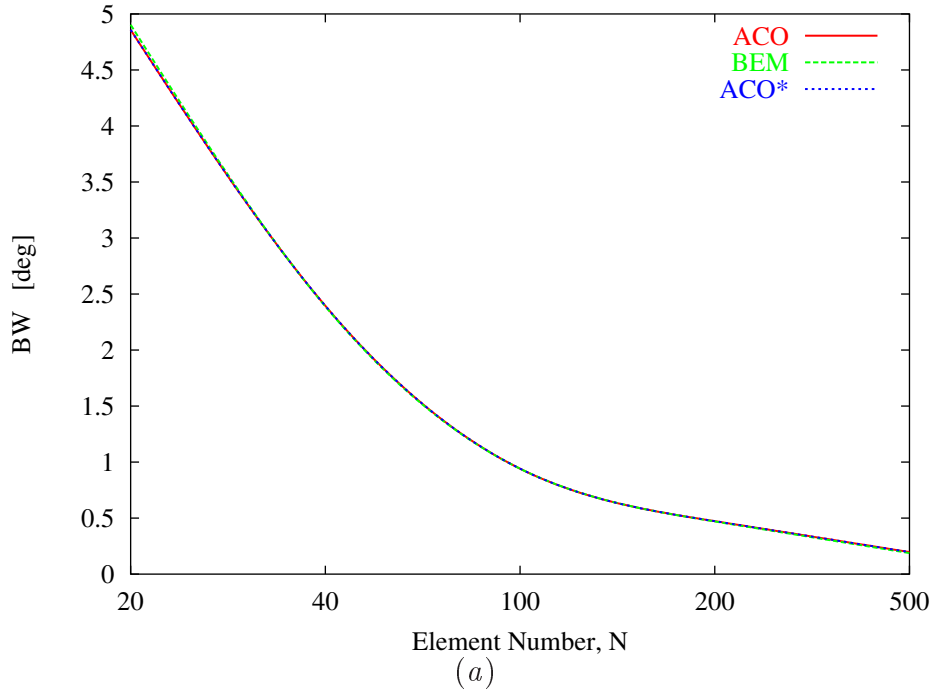


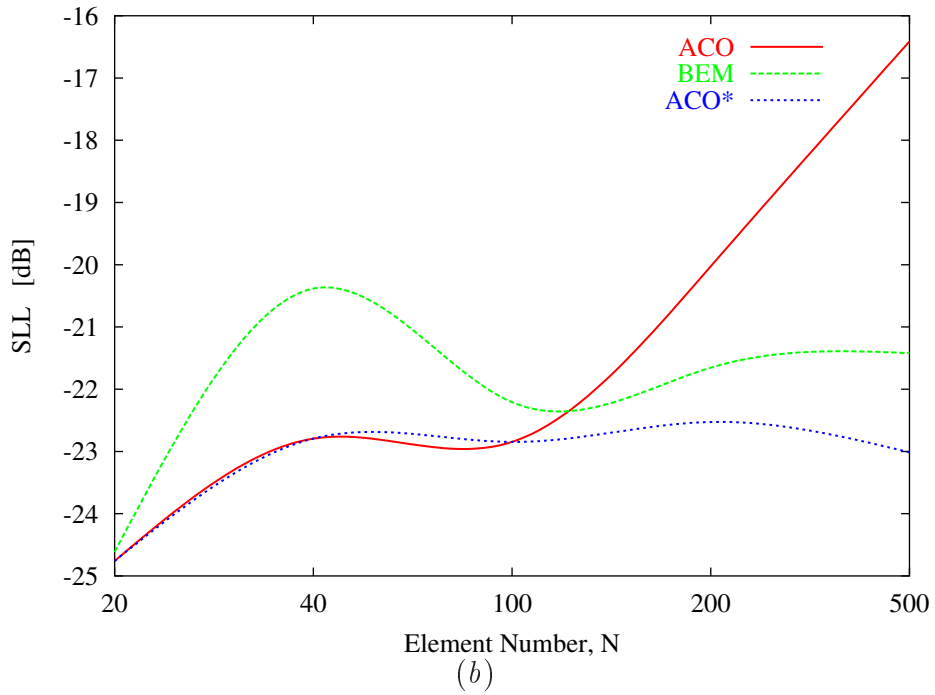
Figure 5.11: *Comparative Assessment* (Zolotarev [9], $SLL = -25\,dB$, $Q = 8$) - Behavior of the average convergence cost function value versus the number of array elements, N.

a conclusion clearly evidences the significant reduction of the computational burden as well as the more profitable and proper use of a suitable global optimization technique within the combinatorial framework. As a matter of fact, although also in this case the convergence to the global optimum solution is not guaranteed, the probability of reaching it significantly grows compared to the standard use of global optimizers. In order to detail such an argumentation, let us assume one has at disposal a limited amount of time $\Delta t^{(tot)}$ for defining the best aggregation for the compromise problem at hand. On one hand, the *ACO*-based approach would have $\Delta K = \hat{K}' - \hat{K}$ more iterations for exploring the solution space, being $\hat{K}' = \frac{\Delta t^{(tot)}}{\delta t \times I^{(ess)}}$. On the other hand, it would be possible to use a larger colony of $I_1^{(ess)} = \frac{\Delta t^{(tot)}}{\delta t \times \hat{K}}$ ants for the same number of iterations $\hat{K}$ and the following conditions would hold true: $I_1^{(ess)} \gg I^{(ess)}$ and $I_1^{(ess)} \simeq U^{(ess)}$. In this latter case, the convergence of the *ACO*-based procedure to the optimum clustering would be assured since each ant could be assigned to explore a single and different path of the solution graph thus covering/sampling the whole solution space.

In order to assess and confirm these indications, Figures 5.11 and 5.12 summarize the performance achieved with the *BEM* and *ACO* methods. The plots refer to a representative set of simulations performed by varying the number of elements of the array aperture between $N = 20$ and $N = 500$, but maintaining a uniform



(a)



(b)

Figure 5.12: *Comparative Assessment* (Zolotarev [9], $SLL = -25\,dB$, $Q = 8$) - Behaviors of (a) the SLL and (b) the BW values of the synthesized compromise patterns versus the number of array elements, N.

inter-element distance ($d = \frac{\lambda}{2}$). In all the experiments, the sets of reference excitations have been chosen to generate a Dolph-Chebyshev sum pattern with $SLL = -25\,dB$ [19] and a Zolotarev difference pattern with $SLL = -25\,dB$ [9]. Moreover, the number of sub-arrays has been fixed to $Q = 8$. As regards to the ACO values, they are related to the average performance over a statistical set of 50 independent executions of the same simulation (i.e., with the same parametric configuration, but varying the randomness in the ACO). In particular, the plots denoted by ACO and ACO^* indicate the values obtained when the ACO algorithm has been run for $\hat{K} = 1000$ iterations with a colony of $I^{(ess)}$ and $I_1^{(ess)}$ ants, respectively. As expected, the ACO-based approach with $I_1^{(ess)}$ trial solutions for each iteration always outperforms the BEM. Unfortunately, when $U^{(ess)}$ turns out to be too large, both the computational load and the storage requirements of the ACO result quite cumbersome and once again, although with larger dimensions, verify the same drawbacks usually encountered by standard global optimizers when dealing with non-small array geometries. In such a situation, the BEM seems to be more attractive even though less robust against local minima problems.

5.5 Conclusions

In Chapter 2, it has been shown how the excitation matching formulation of the optimal compromise problem can be recast as a combinatorial one by exploiting the knowledge of independently optimal sum and difference modes. Thanks to a tree representation of the set of admissible solutions, a local search strategy, called border element method (BEM), has been implemented to efficiently explore the reduced solution space with a large saving of computational resources. Instead, an ACO-based technique has been here considered in order to avoid the occurrence of sub-optimal aggregations caused by the presence of local minima in the non-convex excitation matching functional where the solution space has been described through a directed acyclic graph.

From the analysis carried out within this research work and summarized in this chapter, the following conclusions can be drawn:

- unlike ACO-based approach, both the dimension of the solution space and computational burden rise much more rapidly when standard global optimizers are used. In practice, these standard stochastic algorithms work effectively only with small arrays thus synthesizing array solutions having a limited angular resolution;

- being a local search technique, the BEM depends on the initial solution, but it is an excellent computational saving technique suitable for synthesizing very large arrays ($N \geq 200$) although without any guarantee of avoiding local minima solutions;

- the *ACO* takes on one side the advantages of global optimization approaches in facing non-convexity, while on the other and to the best of the authors' knowledge, it is the most suitable algorithm among state-of-the-art metaheuristics for path-searching in a graph-represented solution space.

Chapter 6

The Hybrid Approach

A hybrid approach for the synthesis of the "optimal" compromise between sum and difference patterns for sub-arrayed monopulse antennas is presented. Firstly, the sub-array configuration is determined by exploiting the knowledge of the optimum difference mode coefficients to reduce the dimension of the searching space. In the second step, the sub-array weights are computed by means of a convex programming procedure, which takes advantages from the convexity, for a fixed clustering, of the problem at hand. A set of representative results are reported to assess the effectiveness of the proposed approach. Comparisons with state-of-the-art techniques are also presented.

6.1 Introduction

In the recent literature, the use of a hybrid approach, namely, the *Simulated Annealing Convex Programming* ($Hybrid - SA$) method [13], for the synthesis of sub-arrayed monopulse linear antennas has improved the performances in shaping compromise patterns with respect to reference approaches [10]-[11]. By considering a sub-arraying strategy [8], the procedure proposed in [13] is aimed at finding "*the sub-array configuration and the coefficients of the sub-array sum signals such that the corresponding radiation pattern has a null with the maximum possible slope in a given direction, while being bounded by an arbitrary function elsewhere.*" Such a solution allows one the use of simpler feeding networks that guarantee both a reduced circuit complexity and low electromagnetic interferences as well as to obtain patterns with user-defined characteristics. It is based on the exploitation of the convexity of the functional with respect to a subset of the unknowns (i.e., the sub-array gains) and it is carried out by means of a *Convex Programming* (CP) method [13]. However, since the sub-array memberships of the array elements are determined by means of a Simulated Annealing (SA) algorithm, the procedure involves non-negligible computational costs to achieve the global minimum or there is the possibility that the solution is trapped in a local minimum (whether the criterion for the SA convergence has not been verified [39]). In order to save computational resources, the *Contiguous Partition Method* (CPM) is used. The CPM takes advantage from the knowledge of the optimal excitations of the difference pattern [7][9][40] and from the concept of contiguous partitions [18] to reduce the searching space and, thus, effectively handling the problem of the optimal clustering. As a matter of fact, the arising computational burden turns out to be significantly reduced compared to that of previous optimization schemes.

In the following, a hybrid approach (called $Hybrid{-}CPM$ method), which integrates the CPM with a gradient-based CP procedure [13] to profitably benefit of the positive features of both CPM and CP approach is carefully described and validated. At the first step, the "optimal" sub-array configuration is computed according to the procedure described in Chapter 2 by exploiting the relationship between the excitation coefficients of the optimal sum [19][5][17][41] and difference [7][9][40] modes. Once the clustering has been determined, the sub-array gains are computed as in [13].

6.2 Synthesis of Linear Arrays

Let us consider a linear array of $N = 2M$ equally-spaced isotropic elements whose generic excitation coefficients are a_n, $n = -M, \ldots, -1, 1, \ldots, M$ and the corresponding space factor given by:

$$f(\theta) = \sum_{n=-M}^{M} a_n e^{j(n-sgn(n)/2)kd\cos(\theta)} \tag{6.1}$$

where k and $d = \frac{\lambda}{2}$ are the wavenumber of the background medium and the inter-element spacing, respectively. Moreover, θ indicates the angular rotation with respect to the direction orthogonal to the array.

The $Hybrid - CPM$ approach belongs to sub-arraying techniques, but unlike the $Hybrid - SA$, it considers a two-stage-iterative procedure instead of an iterative one step process wherein each step involves in turn the solution of a convex optimization problem. The first step is based on the CPM, just presented in Chapter 2. As already pointed out, the solution of such a problem is "*a contiguous partition of M completely ordered elements into Q subsets that may be represented by $Q-1$ points of division lying in any of the $M-1$ intervals between adjacent elements*" [18]. This solution represents the best step-wise approximation of the considered partition and "*the number of possible contiguous partitions is equal to the number of ways of choosing the division points, which is the number of combinations of $M-1$ different things taken $Q-1$ at a time* [i.e., $U^{CPM} = \begin{pmatrix} M-1 \\ Q-1 \end{pmatrix}$, U^{CPM} being the number of contiguous partition]". Accordingly, $\underline{C}^{CPM}$ is determined by generating a sequence of contiguous partitions $\left\{\underline{C}^{(k)};\, k = 0, ..., K\right\}$ starting from a guess aggregation $\underline{C}^{(0)}$ and updating the solution $[\underline{C}^{(k)} \leftarrow \underline{C}^{(k+1)}]$ just modifying the membership of the "*border elements*" of the array.

The second step exploits the following property [13]: "*the optimal compromise between sum and difference patterns is a convex problem with respect to the sub-array weights for a fixed sub-array configuration $\underline{C}$*". Accordingly, once the element membership has been determined [i.e., $\underline{C}^{(opt)} = \underline{C}^{CPM}$], the optimal weight vector $\underline{W}^{(opt)}$ is computed by minimizing the following cost function

$$\Psi^{CP}(\underline{W}) = \left.\frac{d\Re\left\{f^d(\theta)\right\}}{d\theta}\right|_{\theta=\theta_0} \tag{6.2}$$

subject to $\left.\frac{d\Im\left\{f^d(\theta)\right\}}{d\theta}\right|_{\theta=\theta_0} = 0$ and $\left|f^d(\theta)\right|^2 \leq \aleph(\theta)$, where θ_0 indicates the boresight direction and $\aleph(\theta)$ is a non-negative function that defines the upper bounds for the sidelobes. Moreover, $\underline{W} = \{w_q;\, q = 1, \ldots, Q\}$ is the sub-array weight vector and $\Re$ and $\Im$ denote the real part and the imaginary one, respectively. Towards this end, a standard gradient-based optimization is performed by generating a succession of trial solutions $\left\{\underline{W}^{(h)};\, h = 0, ..., H\right\}$ starting from the initial guess given by $\underline{W}^{(0)} = \left\{w_q^{CPM};\, q = 1, \ldots, Q\right\}$ being $w_q^{CPM} = \left[\frac{\sum_{j=1}^{M} \delta_{qc_j}\left(a_j^s a_j^d\right)}{\sum_{j=1}^{M} \delta_{qc_j}\left(a_j^s\right)^2}\right]$.

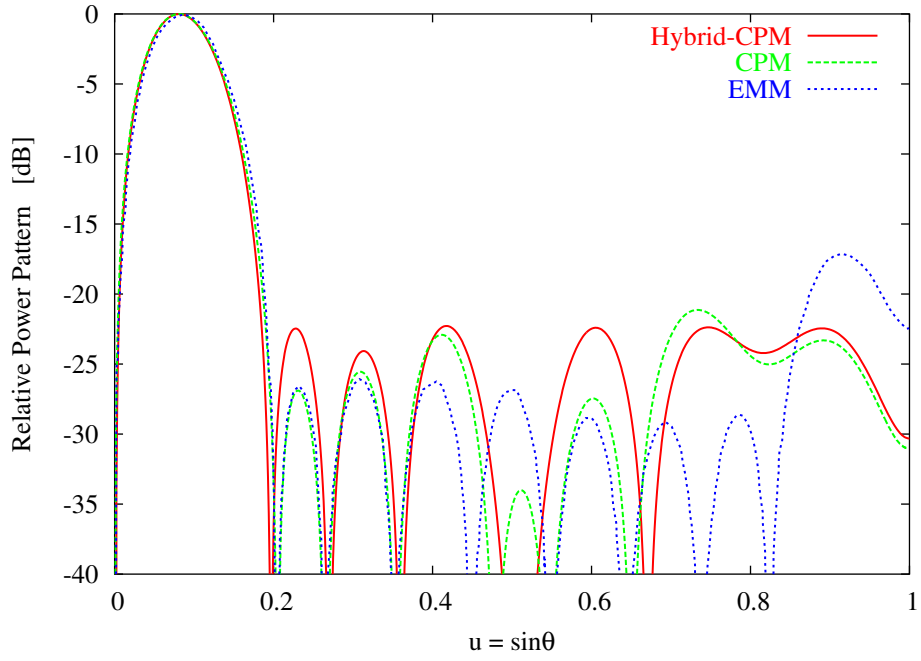


Figure 6.1: *Uniform Sub-arraying* ($M = 10$, $Q = 5$) - Normalized compromise difference patterns obtained by means of the $Hybrid-CPM$ method, the CPM, and the EMM [8].

6.2.1 Numerical Assessment

In this section, the effectiveness and potentialities of the proposed hybrid method will be assessed dealing with three benchmarks of the related literature. As a matter of fact, the test cases under analysis are concerned with linear arrays and, for the sake of completeness, with both a small ($M = 10$) and a large ($M = 100$) number of elements. Whatever the experiment, the synthesis is aimed at minimizing the SLL of the compromise difference pattern for a fixed beamwidth or, analogously, at maximizing the slope along the boresight direction [13] fixed at $\theta_0 = 0^o$.

The first test case deals with a linear array of $N = 20$ elements. As far as the sum mode is concerned, it has been fixed to a Villeneuve sum pattern [17], with $\bar{n} = 4$ and $SLL = -25\,dB$, in the first experiment, whereas a Dolph-Chebyshev [19] pattern with $SLL = -20\,dB$ has been chosen for the second one. In the first experiment, a configuration with $Q = 5$ sub-arrays and uniform clustering is considered. Moreover, as regards the optimal/reference difference pattern of the approaches that exploit the concept of contiguous partitions, the optimal difference excitations have been fixed to a modified Zolotarev distribution ($\overline{n} = 4$, $\varepsilon = 3$) whose pattern is characterized by $SLL_{ref} = -25\,dB$. Figure 6.1 pictorially compares the patterns obtained with the EMM [8], the CMP, and the $Hybrid-$

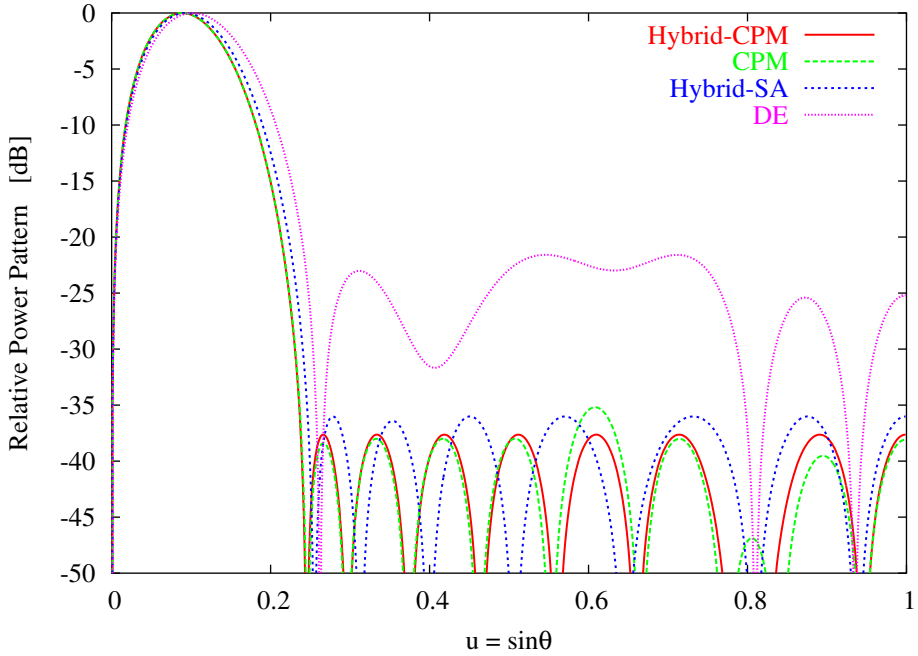
$[dB]$		$Reference$	$Hybrid-CPM$	CPM	EMM	$Hybrid-SA$	DE
$M=10$	$Q=5$	-25.0	-22.4	-21.0	-17.0	–	–
$M=10$	$Q=8$	-39.0	-37.5	-35.2	–	-36.5	-21.6
$M=10$	$Q=8$	-41.0	-38.0	-32.7	–	-36.5	-21.6
$M=100$	$Q=6$	-30.0	-28.3	-25.7	–	–	–

Table 6.1: Values of the SLL of the array factors in Figs. 6.1-6.3.

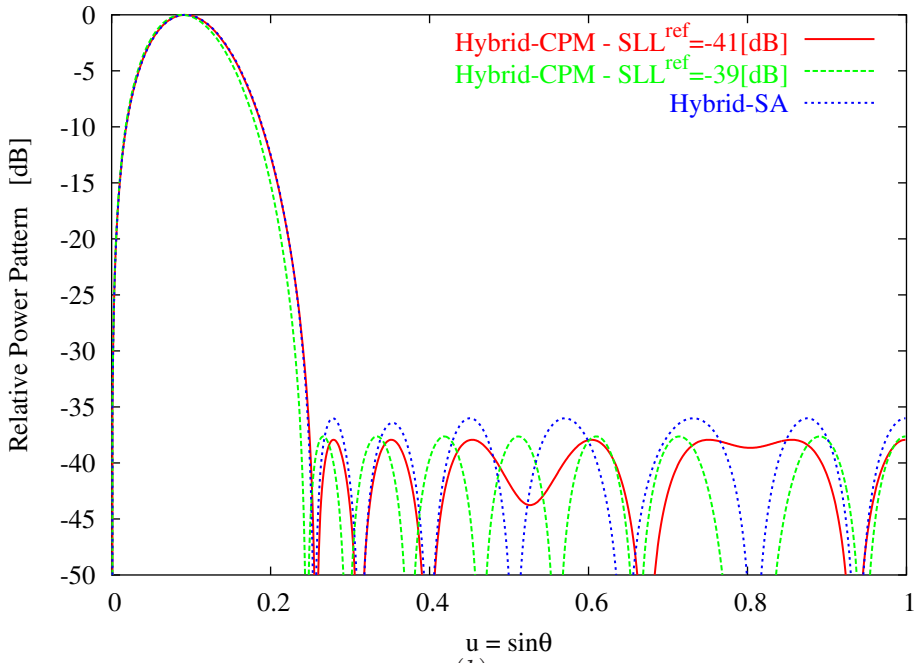
CPM approach, whose final sub-array configuration and weights are $\underline{C}^{(opt)} = \{1\,1\,2\,3\,3\,5\,5\,4\,4\,2\}$ and $\underline{W}^{(opt)} = \{0.3352,\ 1.1299,\ 1.3708,\ 1.8309,\ 1.8699\}$, respectively. It is worth noting that the $Hybrid-CPM$ approach outperforms other methods with a reduction of over $5\,dB$ and more than $1\,dB$ of the the SLL with respect to the EMM and the CPM, respectively (Tab. 6.1) .

The second experiment is devoted to complete the comparison by considering the state-of-the-art methods based on stochastic optimizations. In particular, the results from the $Hybrid-SA$ [13] and the Differential Evolution (DE) optimization algorithm [11] have been taken into account. The array configuration is that with $Q=8$. The array patterns obtained from the application of the CPM-based methods and by assuming a reference Zolotarev pattern [9] with $SLL_{ref} = -39\,dB$ are shown in Fig. 6.2(a) together with those from the other approaches. With reference to Fig. 6.2(a) and as quantitatively estimated in Tab. 6.1, the $Hybrid-CPM$ plot presents a SLL of $-37.5\,dB$ (i.e., almost $1\,dB$ below the SLL of the $Hybrid-SA$ [13] and more than $15\,dB$ when compared to the pattern in [11] with the same number of sub-arrays), with $\underline{C}^{(opt)} = \{2\,3\,5\,7\,8\,8\,6\,4\,3\,1\}$ and $\underline{W}^{(opt)} = \{1.1836,\ 1.8818,\ 4.9795,\ 6.9286,\ 7.3462,\ 8.5109,\ 9.1480,\ 9.7003\}$. Furthermore, it is worth analyzing the beamwidths (BWs) (or, similarly, the first null positions) of the results in Fig. 6.2(a). As a matter of fact, the $Hybrid-CPM$ solution presents not only the lowest SLL value, but also the narrower BW (i.e., $BW_{Hybrid-CPM} = 0.097$ vs. $BW_{Hybrid-SA} = 0.102$ and $BW_{DE} = 0.113$). Such a result further confirms the effectiveness of the $Hybrid-CPM$ in dealing with the non-convex part of the problem at hand, thus allowing the synthesis of compromise patterns with better characteristics. As expected, the improvements in terms of SLL are even larger by setting the same BW constraint used with $Hybrid-SA$ [13]. Towards this aim, the reference excitations have been chosen to afford a Zolotarev difference pattern [9] with $SLL_{ref} = -41\,dB$. In such a case, the achieved solution has a $SLL = -38.0\,dB$ with an improvement of about $0.5\,dB$ [Tab. 6.1] compared to that in Fig. 6.2(a). For completeness, the values of the obtained clustering and sub-array weights are equal to $\underline{C}^{(opt)} = \{2\,4\,6\,8\,8\,8\,7\,5\,3\,1\}$ and $\underline{W}^{(opt)} = \{0.7461,\ 2.0518,\ 4.0934,\ 5.4616,\ 6.5563,\ 8.2545,\ 8.5060,\ 10.0768\}$, respectively.

As far as the computational costs are concerned, the number of iterations, K, required to get the final clustering starting from a uniform one at the initial-



(a)



(b)

Figure 6.2: *Non-Uniform Sub-arraying* ($M = 10$, $Q = 8$) - Normalized compromise difference patterns obtained by means of the $Hybrid - CPM$ method, the CPM , the $SA - CP$ approach [13], and the DE optimization [11].

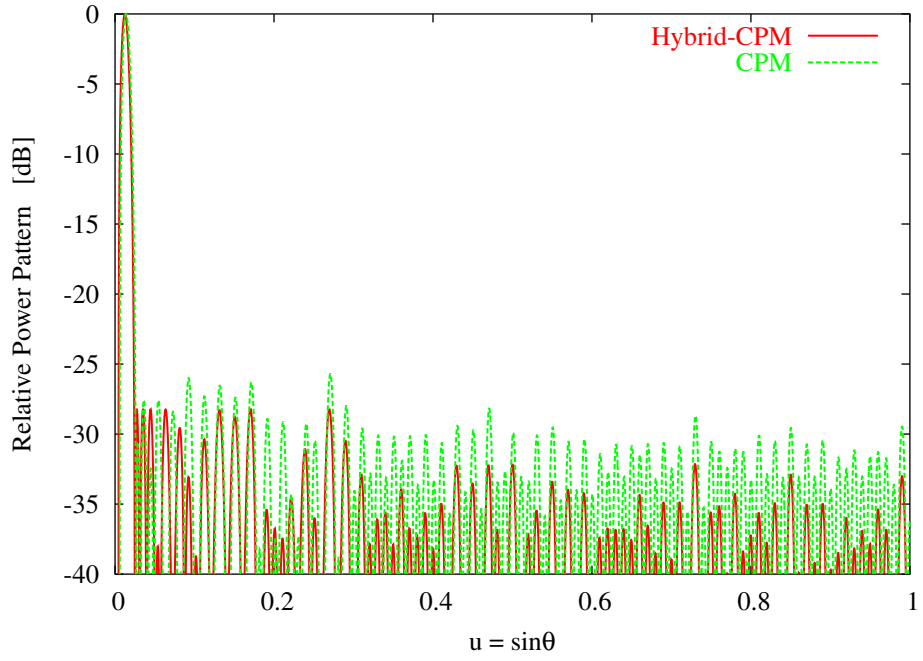


Figure 6.3: *Large Arrays* ($M = 100$, $Q = 6$) - Normalized compromise difference patterns obtained with the $Hybrid - CPM$ method and the CPM.

ization, is $K_{CPM} = 4$ and $K_{CPM} = 3$, for the two CPM-based syntheses, respectively, and the total CPU-time is shorter than 10 [μsec] in both cases. Moreover, the whole synthesis time of the $Hybrid - CPM$ amounts to 3.078 [sec] and 3.781 [sec], respectively. As regards to the higher burden of the $Hybrid - CPM$ compared to the CPM, this is due to the solution of the CP problem, which ends in $K_{CP} = 18$ iterations. For comparative purposes, let us notice that a greater computational burden affects the $Hybrid - SA$ [13] method since $K_{Hybrid-SA} = 25$ have been chosen and CP problem is solved at each iteration. Similar conclusions hold true also for the DE approach [11] where the number of iterations has been set to $K_{DE} = 10$.

The last comparative example deals with the synthesis of a large array ($N = 200$). Thanks to the computational saving, the CPM-based procedures are able to effectively face with such a problem dimensionality. The sum coefficients have been chosen to generate a Dolph-Chebyshev [19] pattern with $SLL = -25\,dB$, while the values of the reference difference excitations have been fixed to those of the Zolotarev difference pattern with $SLL_{ref} = -30\,dB$. The behaviors of the patterns in Fig. 6.3 clearly point out that the integration of the CP optimization with the CPM allows a non-negligible enhancement of the SLL performances. As a matter of fact, the SLL computed in correspondence with the clustering determined by the $Hybrid - CPM$ method (Tab. 6.2) is of about $3\,dB$ lower

$M = 100$	$\underline{C}$	1111111111111122222222333333334444444455555555566666666666666666666666666666666555555555444444433331					
$Q = 6$	$\underline{W}$	0.2133	0.7235	0.9417	1.0909	1.2752	1.4294

Table 6.2: *Large Arrays* ($M = 100$, $Q = 6$) - Sub-array configuration and weights determined by the $Hybrid - CPM$ method (see Fig. 6.3 for the corresponding pattern).

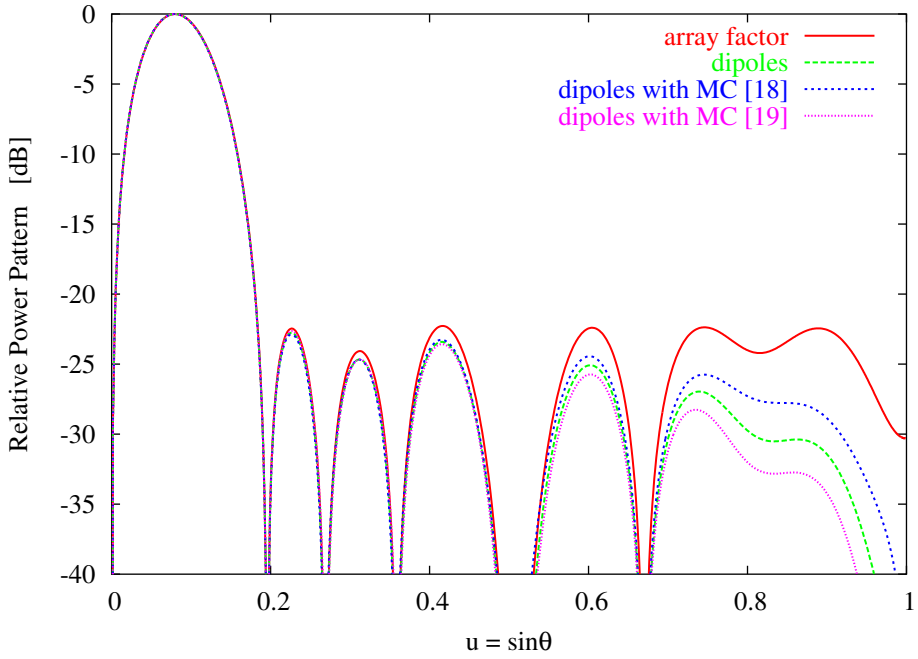


Figure 6.4: *Large Arrays* ($M = 100$, $Q = 6$) - Normalized compromise difference patterns obtained with the $Hybrid - CPM$ method and the CPM.

than that of the standard version of the CPM (see Tab. 6.1).

Finally, in order to assess the reliability of the synthesized solutions, let us evaluate the radiated power patterns when mutual coupling (MC) effects are included into the array model. Towards this purpose, the MC models proposed in [42] and [43] have been taken into account and compared as in [44]. The case-of-study example deals with a 20-element uniform linear array of thin $\lambda/2$ dipoles oriented along the z axis [45]. As a representative example, the effects of the MC on the solution obtained with the $Hybrid - CPM$ approach and shown in Fig. 6.1 are analyzed. Figure 6.4 shows the pictorial representations of the relative power patterns for different situations. As it can be observed, the radiation pattern obtained by including the MC effects is similar to the ideal case whatever the considered MC model. More in detail, the null positions are equal

to those of the ideal pattern, while some perturbations only affect the behavior of the secondary lobes without compromising the performance of the difference beam.

6.3 Synthesis of Linear Arrays

A hybrid version of the $ICPM$ (i.e., the $Hybrid-ICPM$) presented in Chapter 3 is customized to the synthesis of planar arrays in order to extend the range of applicability of the planar CPM from excitation matching to pattern optimization allowing, unlike the $ICPM$, a direct control of the pattern features (i.e., SLL, BW, etc...).

Similarly to linear array case, the hybrid approach consists of a two-step procedure where at the first step the sub-array configuration is computed according to the $IBEM$ (i.e., $\underline{C}^{\triangle}_{Hybrid-ICPM} = \underline{C}^{\triangle}_{opt}$). Successively, the weights $\underline{W}^{\triangle}_{Hybrid-ICPM}$, $\triangle = E, H$, of the sub-arrayed difference network are computed by means of a standard CP procedure minimizing the following cost function (where the notation is the same of Chapter 4)

$$\Psi^{CP}\left(\underline{W}^{\triangle}\right) = \min_{\left\{w_q^{\triangle};\, q=1,\ldots,Q\right\}} \left. \frac{\partial \left\{ \sum_{r=-R}^{R} \sum_{s=-S(r)}^{S(r)} \left[\Re\left(\gamma_{rs}^{\triangle}\right) \cos \Upsilon\left(\theta,\phi\right) - \Im\left(\gamma_{rs}^{\triangle}\right) \sin \Upsilon\left(\theta,\phi\right) \right] \right\}}{\partial \chi} \right|_{\substack{\theta=0 \\ \phi=0}} \tag{6.3}$$

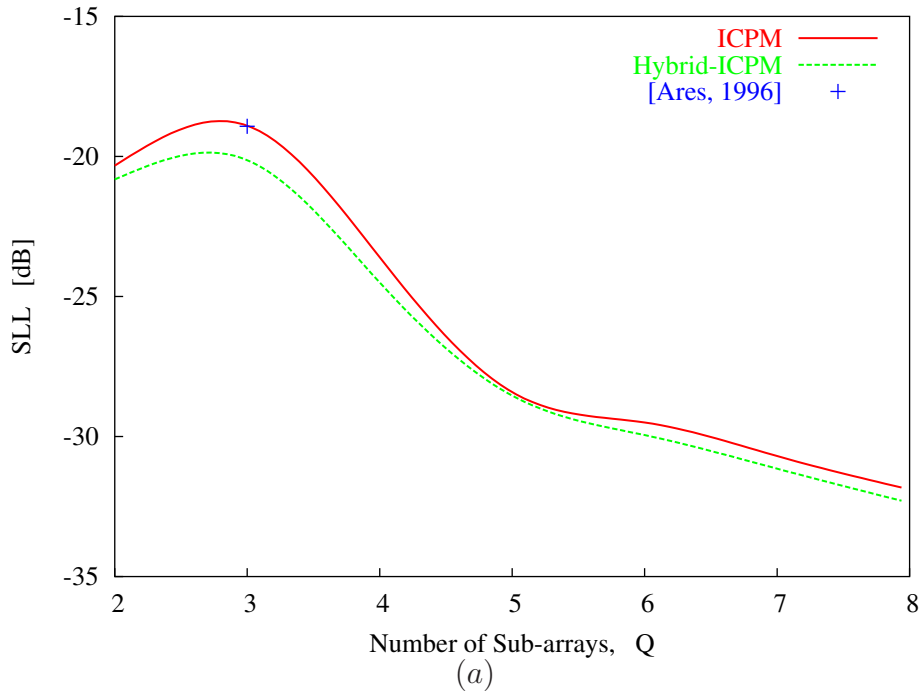
χ being either θ or ϕ and $\Upsilon\left(\theta,\phi\right) = k_x x_m + k_y y_n$, subject to

$$\left. \frac{\partial \left\{ \sum_{r=-R}^{R} \sum_{s=-S(r)}^{S(r)} \left[\Re\left(\gamma_{rs}^{\triangle}\right) \sin \Upsilon\left(\theta,\phi\right) + \Im\left(\gamma_{rs}^{\triangle}\right) \cos \Upsilon\left(\theta,\phi\right) \right] \right\}}{\partial \chi} \right|_{\substack{\theta=0 \\ \phi=0}} = 0 \tag{6.4}$$

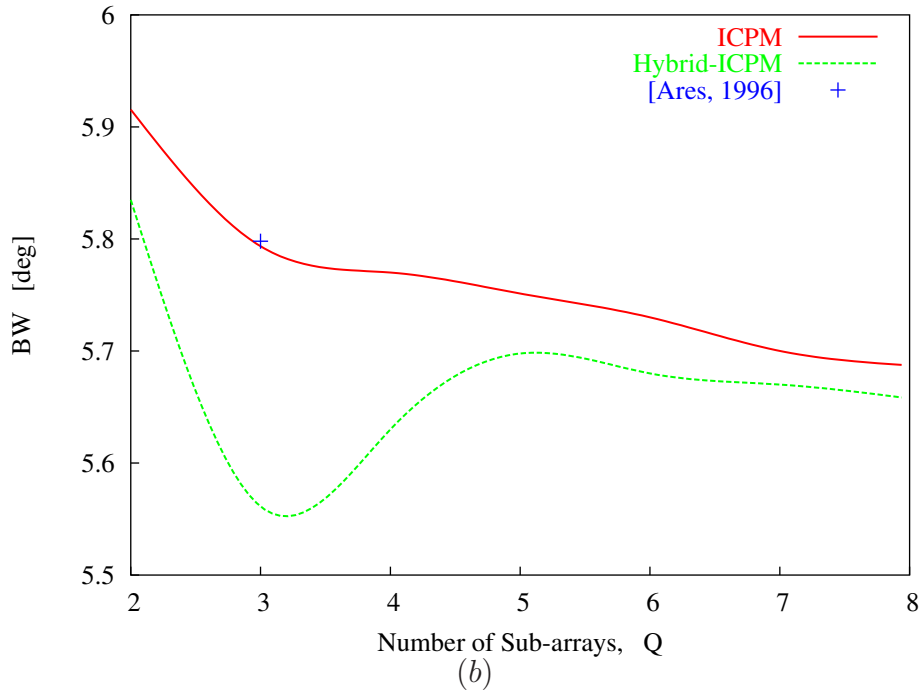
and

$$AF\left(\theta,\phi\right)\Big|_{\substack{\theta=0 \\ \phi=0}} = \sum_{r=-R}^{R} \sum_{s=-S(r)}^{S(r)} \gamma_{rs}^{\triangle} = 0 \tag{6.5}$$

and to $\left|AF\left(\theta,\phi\right)\right|^2 \leq \mathcal{M}\left(\theta,\phi\right)$ where $\mathcal{M}\left(\theta,\phi\right)$ is a function descriptive of a user-defined mask on the synthesized difference power pattern. In Eq. (6.3), $\Re\left(\cdot\right)$ and $\Im\left(\cdot\right)$ denote the real and imaginary part, respectively. At the initialization of the CP procedure, the guess solution is set to the values of the sub-array weights obtained at the end of the $ICPM$, $\underline{W}^{\triangle,(0)} = \underline{W}^{\triangle}_{opt}$.



(a)



(b)

Figure 6.5: *Hybrid Formulation* ($N = 300$, $d = \frac{\lambda}{2}$, $r = 4.85\lambda$) - Behavior of the (a) SLL and of the (b) BW for the compromise patterns synthesized by means of the $ICPM$ and the $Hybrid - ICPM$ when $Q \in [2, 8]$.

	$CPU - Time\ [sec]$				T_{CP}			
Q	2	3	5	8	2	3	5	8
$ICPM$	2.30	2.64	3.12	7.23	37	45	57	120
$Hybrid - ICPM$	7554.68	8678.15	9623.57	7314.06	2114	2415	2675	2113

Table 6.3: *Hybrid Formulation* ($N = 300$, $d = \frac{\lambda}{2}$, $r = 4.85\lambda$) - Computational indexes for the solution obtained with the $ICPM$ and the $Hybrid - ICPM$.

In order to show the SLL/BW control allowed by the hybrid approach, Figure 6.5 summarizes the results from a comparative study between the $ICPM$ and its hybrid version in terms of maximum SLL [Fig. 6.5(a)] and corresponding BW computed on the principal plane [i.e., the $\phi = 0^o$] [Fig. 6.5(b)] dealing with the same array configuration of Sect. 4.3.2. To better and more exhaustively analyze the potentialities of the proposed hybrid approach, the number of sub-arrays has been varied in the range $Q \in [2, 8]$ and the synthesized sub-arrays configurations and weights are shown in Fig. 6.6 . For completeness, the corresponding patterns are also given [Fig. 6.7] . As it can be observed (Figs. 6.7-6.5), the solutions from the $Hybrid - ICPM$ outperform those of the $ICPM$ in terms of pattern indexes even though with heavier computational costs. As far as the computational issues are concerned, the dimension of the solution space $U^{(DAG)}$ and the storage resources $M^{(DAG)}$ are given in Fig. 6.8 , whereas the CPU-time and number of iterations T_{CP} required to get the final solution for the $Hybrid - ICPM$ and $ICPM$ are reported in Tab. 6.3 to point out the trade-off between pattern efficiency and computational burden.

6.4 Discussions

Concerning the optimization problem at hand, the proposed CPM-based procedure does not guarantee that the retrieved sub-array configuration is the best choice for optimizing the SLL. As a matter of fact, such a configuration can be (theoretically) obtained only by means of global optimization procedures. However, the proposed procedure has shown to outperform state-of-the-art global optimization strategies. Furthermore, starting from the assumption that CPM-based strategies are matching techniques, the proposed approach can be easily extended to arbitrary sidelobe masks or pattern shapes (for both sum and difference patterns) by profitably using the state-of-the-art approaches (e.g., [40][41]) to set the reference patterns.

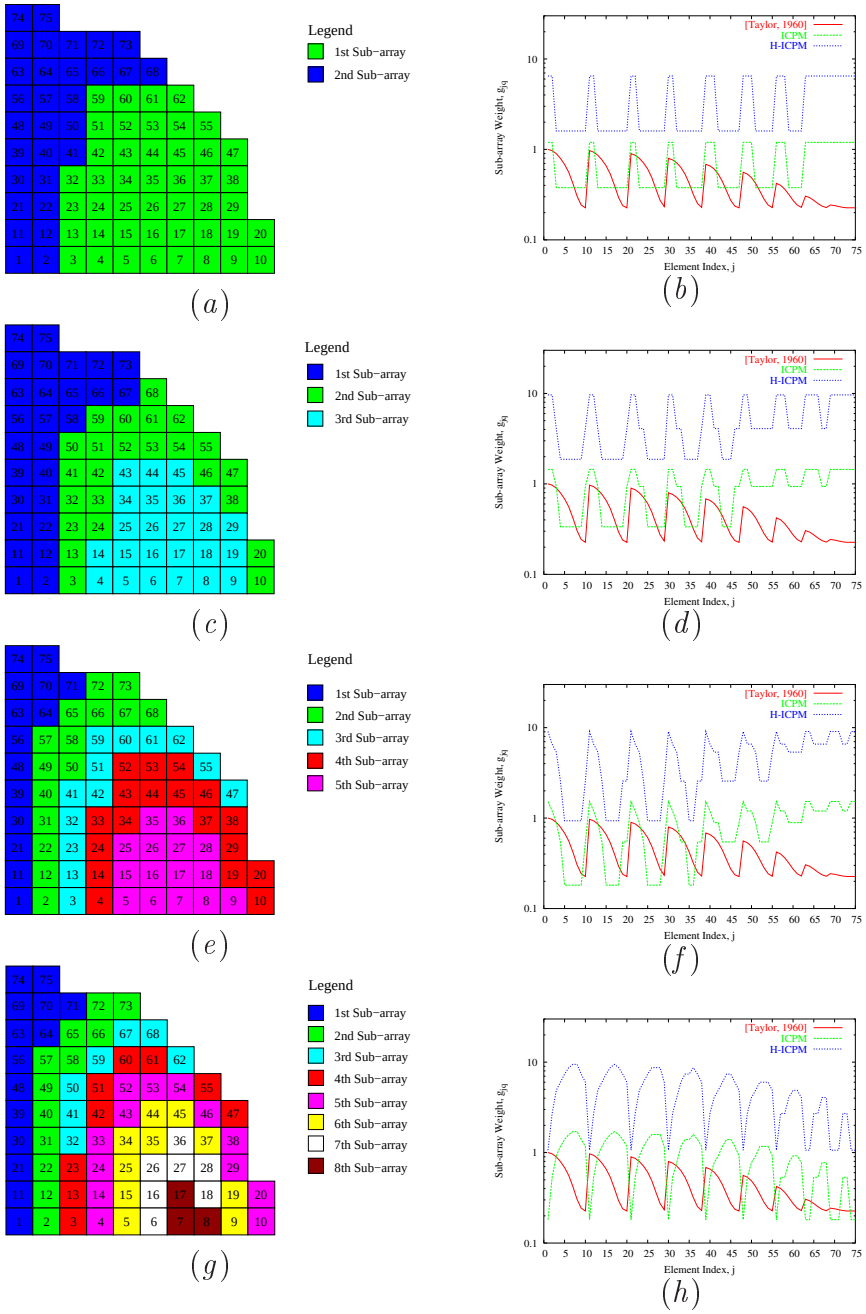


Figure 6.6: *Hybrid Formulation* ($N = 300$, $d = \frac{\lambda}{2}$, $r = 4.85\lambda$, $Q = 3$) - Sub-array configurations (*left column*) and array element weights (*right column*) synthesized with the $ICPM$ and the $Hybrid - ICPM$ for different values of Q [$Q = 2$ (*first row*), $Q = 3$ (*second row*), $Q = 5$ (*third row*), and $Q = 8$ (*fourth row*)].

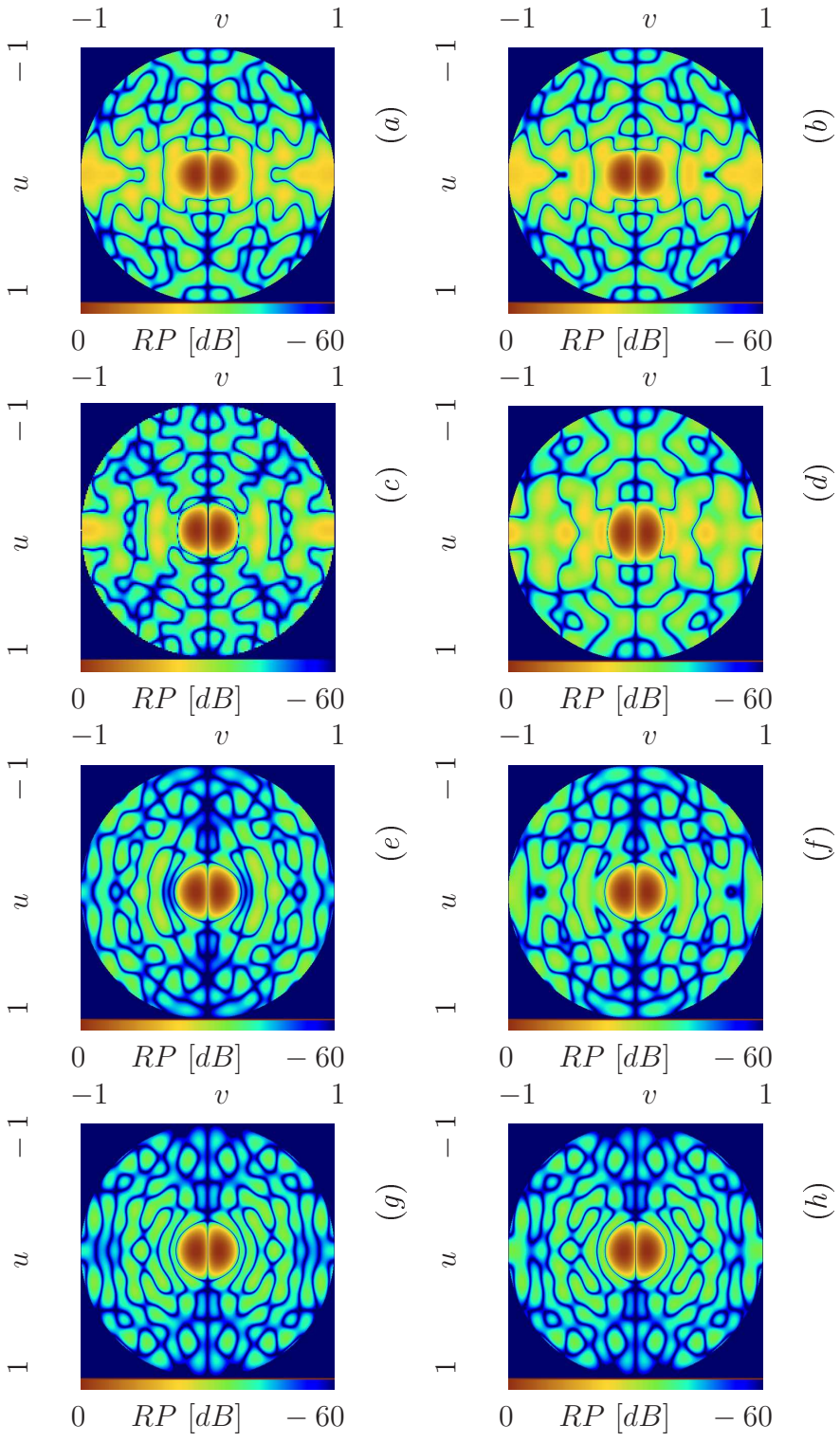


Figure 6.7: *Hybrid Formulation* ($N = 300$, $d = \frac{\lambda}{2}$, $r = 4.85\lambda$, $Q = 3$) - Beam patterns synthesized with the $ICPM$ (*left column*) and the $Hybrid - ICPM$ (*right column*) for different values of Q [$Q = 2$ (*first row*), $Q = 3$ (*second row*), $Q = 5$ (*third row*), and $Q = 8$ (*fourth row*)].

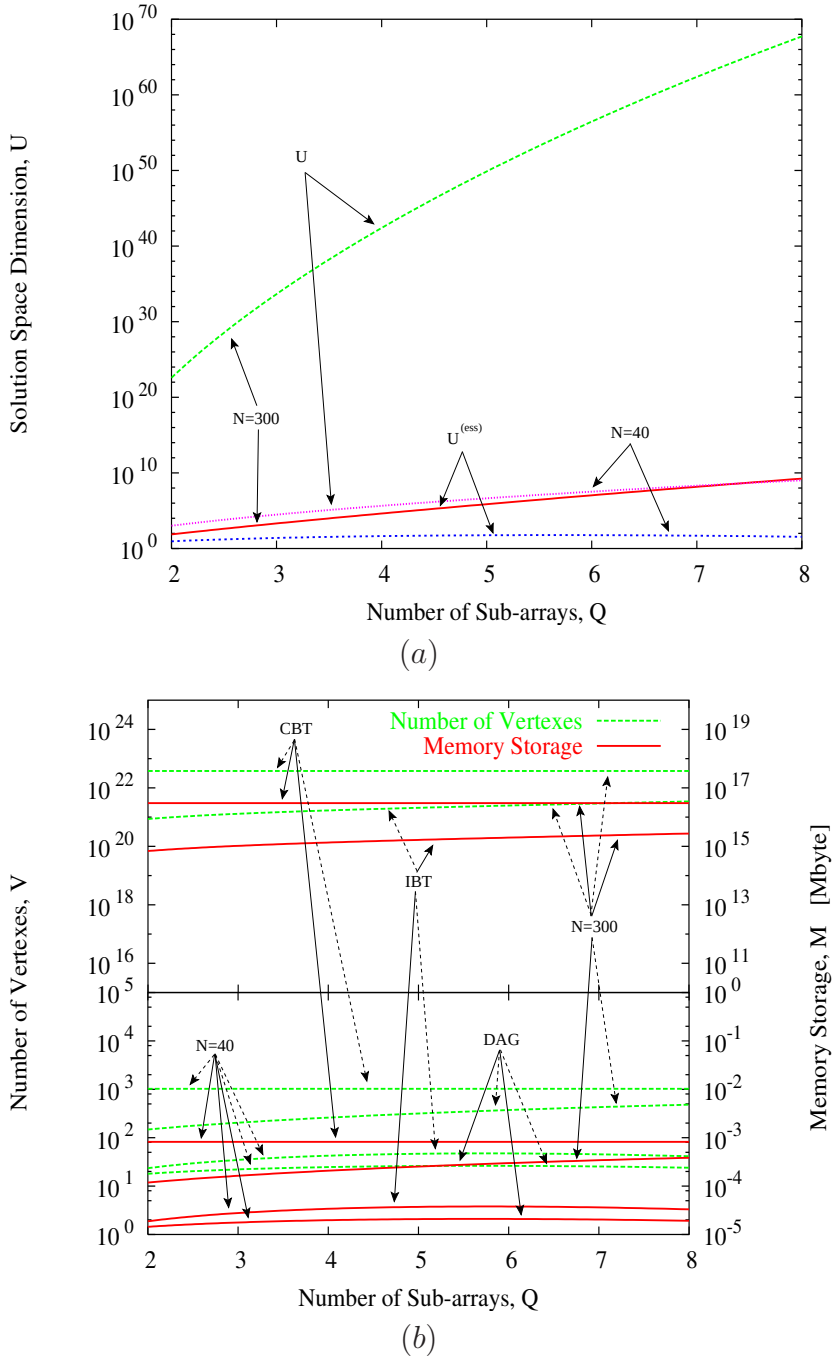


Figure 6.8: *Sub-Arrayed Planar Array Synthesis* ($d = \frac{\lambda}{2}$, $r = 4.85\lambda$). *Computational Analysis* - (a) Dimension of the solution space U and (b) memory resources, M, and number of vertexes, V, for the storage of the representations of the solution space versus Q in correspondence with $N = 300$ and $N = 40$ (CBT →Complete Binary Tree, IBT →Non-Complete Binary Tree, and DAG →Direct Acyclic Graph).

Chapter 7

Conclusions and Future Developments

In this last section, some conclusions are drawn and further advances are envisaged in order to address the possible developments of the proposed technique.

In this thesis, an approach for the synthesis of monopulse array antennas has been presented and validated. The method is based on an excitation matching procedure to design sub-arrayed antennas generating an optimal sum and compromise difference patterns. Thanks to the knowledge of the reference excitation set, the synthesis problem has been reformulated as a combinatorial one to allow a considerable saving of computational resources. Thanks to a graph-based representation of the solution space, the use of an efficient path-searching algorithm to speed-up the convergence of the procedure for the synthesis of large array antennas as well as the use of the Ant Colony Optimizer (ACO) to benefit of its hill-climbing properties in dealing with the non-convexity of the sub-arraying problem have been considered. Moreover, a hybrid approach has been developed to individually control the level of the secondary lobes. A set of representative examples concerned with both pattern matching problems and pattern-feature optimization have been reported in order to assess the effectiveness and flexibility of the proposed approach. Comparisons with previously published results have been shown and discussed, as well.

Concerning the methodological novelties of this work, the main contribution is concerned with the following issues:

- an appropriate definition of the solution space by means of a graph structure;
- an original and innovative formulation of the sum-difference problem in terms of a search in a graph;
- a simple and fast solution procedure based on swapping operations among border elements and cost function evaluations.

Moreover, the main features of the proposed graph-based techniques are the following:

- a reduction of the dimensionality of the solution space for the synthesis problem at hand, by exploiting the information content of independently optimal sum and difference excitations;
- a significant reduction of the computational burden, by applying a fast solution algorithm for exploring the solution graph (i.e., sampling the solution space);
- the capability to deal with the synthesis of large linear and planar arrays in an effective and reliable way.

As far as future developments are concerned, this approach promises to show its flexibility and capability also with time-varying scenarios and not only with the "static" array synthesis. In such a framework, techniques for the control and

synthesis of adaptive phase-array are of interest since they allow to shape in real time the radiation pattern and in particular the secondary lobes for noise and interference rejection.

Moreover, the possibility of integrating the time modulating strategy for the synthesis of patterns with low and ultra-low sidelobes can be investigated where a set of RF switches are used to commute between the open and short circuit state in order to enforce a time modulation on the element/sub-array excitations.

References

[1] S. M. Sherman. *Monopulse principles and techniques*, Artech House, 1984.

[2] M. I. Skolnik, *Radar Handbook*. New York: McGraw-Hill, 1990.

[3] W. Wirth, *Radar Techniques using Array Antennas*. IET Press, 2001.

[4] P. W. Hannan, "Optimum Feeds for All Three Modes of a Monopulse Antenna I: Theory," *IRE Trans. Antennas Propag.*, vol. 9, no. 5, pp. 444-454, Sep. 1961.

[5] T. T. Taylor, "Design of line-source antennas for narrow beam-width and low side lobes," *Trans. IRE Antennas Propag.*, vol. 3, pp. 16-28, 1955.

[6] T. T. Taylor, "Design of a circular apertures for narrow beamwidth and low sidelobe," *Trans. IRE Antennas Propag.*, vol. 8, pp. 17-22, 1960.

[7] E. T. Bayliss, "Design of monopulse antenna difference patterns with low sidelobes," *Bell System Tech. Journal*, vol. 47, pp. 623-640, 1968.

[8] D. A. McNamara, "Synthesis of sub-arrayed monopulse linear arrays through matching of independently optimum sum and difference excitations," *IEE Proc. H Microwaves Antennas Propag.*, vol 135, no. 5, pp. 293-296, 1988.

[9] D. A. McNamara, "Direct synthesis of optimum difference patterns for discrete linear arrays using Zolotarev distribution," *IEE Proc. H Microwaves Antennas Propag.*, vol. 140, no 6, pp. 445-450, 1993.

[10] F. Ares, J. A. Rodriguez, E. Moreno, and S. R. Rengarajan, "Optimal compromise among sum and difference patterns," *J. Electromag. Waves Appl.*, vol. 10, pp. 1543-1555, 1996.

[11] S. Caorsi, A. Massa, M. Pastorino, and A. Randazzo, "Optimization of the difference patterns for monopulse antennas by a hybrid real/integer-coded differential evolution method," *IEEE Trans. Antennas Propag.*, vol 53, no. 1, pp. 372-376, Jan. 2005.

[12] P. Lopez, J. A. Rodriguez, F. Ares, and E. Moreno, "Subarray weighting for difference patterns of monopulse antennas: Joint optimization of subarray configurations and weights," *IEEE Trans. Antennas Propag.*, vol. 49, no. 11, pp. 1606-1608, Nov. 2001.

[13] M. D'Urso and T. Isernia, "Solving some array synthesis problems by means of an effective hybrid approach," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 750-759, Mar. 2007.

[14] M. D'Urso, T. Isernia, and E. F. Meliado' "An effective hybrid approach for the optimal synthesis of monopulse antennas," *IEEE Trans. Antennas Propag.*, vol. 55, no. 4, pp. 1059-1066, Apr. 2007.

[15] Y. Chen, S. Yang, and Z. Nie, "The application of a modified differential evolution strategy to some array pattern synthesis problems," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 7, pp. 1919-1927, Jul. 2008.

[16] D. B. West, *Introduction to Graph Theory*. Englewood Cliffs, NJ: Prentice-Hall, 2000.

[17] A. T. Villeneuve, "Taylor patterns for discrete arrays," *IEEE Trans. Antennas Propag.*, vol. 32, no. 10, pp. 1089-1093, Oct. 1984.

[18] W. D. Fisher, "On grouping of maximum homogeneity," *American Statistical Journal*, 789-798, 1958.

[19] C. L. Dolph, "A current distributionoptimizes for broadside arrays which optimizes the relationship between beam width and sidelobe level," *Proc. IRE*, vol. 34, pp. 335-348, 1946.

[20] A. Massa, M. Pastorino, and A. Randazzo, "Optimization of the directivity of a monopulse antenna with a subarray weighting by a hybrid differential evolution method," *IEEE Antennas Wireless Propag. Lett.*, vol. 5, no. 1, pp. 155-158, 2006.

[21] T. Isernia, F. Ares-Pena, O.M. Bucci, M. D'Urso, X.F. Gomez, and A. Rodriguez, "A hybrid approach for the optimal synthesis of pencil beams thorugh array antennas," *IEEE Trans. Antennas Propag.*, vo. 52, no. 11, pp. 2912-2918, Nov. 2004.

[22] D.A. McNamara, "Performance of Zolotarev and modified-Zolotarev difference pattern array distributions," *IEE Proc. Microwave Antennas Propagat.*, vol. 141, no. 1, pp. 37-44, Feb. 1994.

[23] R. S. Elliott, *Antenna theory and design*. Wiley-Interscience, IEEE Press, 2003.

[24] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Trans. Syst. Man and Cybern. B* , vol. 26, no. 1, pp. 29-41, Feb. 1996.

[25] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28 - 39, Nov. 2006.

[26] M. Dorigo and L. M. Gambarella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53-66, Apr. 1997.

[27] M. Reimann, K. Doerner, and R. F. Hartl, "D-ants: Savings based ants divide and conquer the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 4, pp. 563-591, 2004.

[28] V. Maniezzo, "Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem," *INFORMS J. Comput.*, vol. 11, no. 4, pp. 358-369, 1999.

[29] D. Costa and A. Hertz, "Ants can colour graphs," *J. Operat. Res. Soc.*, vol. 48, pp. 295-305, 1997.

[30] D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource constrained project scheduling," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 333-346, 2002.

[31] D. Merkle and M. Middendorf, "Ant colony optimization with global pheromone evaluation for scheduling a single machine," *Appl. Intell.*, vol. 18, no. 1, pp. 105-111, 2003.

[32] C. Blum and M. J. Blesa, "New metaheuristic approaches for the edge-weight ed k-cardinality tree problem," *Computers & Operations Research*, vol. 32, no. 6, pp. 1355-1377, 2005.

[33] C. M. Coleman, E. J. Rothwell, and J. E. Ross, "Investigation of simulated anneling, ant-colony optimization, and genetic algorithms for selfstructuring antennas," *IEEE Trans. Antennas Propag.*, vol. 52, no. 4, pp. 1007-1014, Apr. 2004.

[34] O. Quevedo-Teruel and E. Rajo-Iglesias, "Ant colony optimization for array synthesis," in *Proc. Antennas Propag. Soc. Int. Symp.*, Jul. 2006, pp. 3301-3304.

[35] O. Quevedo-Teruel and E. Rajo-Iglesias, "Ant colony optimization in thinned array synthesis with minimum sidelobe level," *IEEE Antennas Wireless Propag. Lett.*, vol. 5, pp. 349-352, 2006.

[36] E. Rajo-Iglesias and O. Quevedo-Teruel, "Linear Array synthesis using an ant-colony-optimization-based algorithm," *IEEE Antennas Propag. Mag.*, vol. 49, no. 2, pp. 70-79, Apr. 2007.

[37] M. Pastorino, "Stochastic optimization methods applied to microwave imaging: A review," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 538-548, Mar. 2007.

[38] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67-82, Apr. 1997.

[39] B. Hajek, "Cooling schedules for optimal annealing," *Math. Op. Research*, vol. 13, pp. 311-329, 1988.

[40] O.M. Bucci, M. D'Urso, and T. Isernia, "Optimal synthesis of difference patterns subject to arbitrary sidelobe bounds by using arbitrary array antennas," *IEE Proc. Microwave Antennas Propag.*, vol. 152, no. 3, pp. 129-137, 2005.

[41] T. Isernia, P. Di Iorio, and F. Soldovieri, "An effective approach for the optimal focusing of array fields subject to arbitrary upper bounds," *IEEE Trans. Antennas Propag.*, vol. 48, no. 12, pp. 1837-1847, Dec. 2000.

[42] T. Svantesson, "Modeling and estimation of mutual coupling in a uniform linear array of dipoles," *IEEE Int. Conf. Acoust., Speech, Signal Process.*, Phoenix, AZ, 1999, pp. 2961-2964.

[43] I. J. Gupta and A. K. Ksienski, "Effects of mutual coupling on the performance of adaptive arrays," *IEEE Trans. Antennas Propag.*, vol. 31, no. 5, pp. 785-791, May 1983.

[44] Z. Huang, C. A. Balanis, C. R. Birtcher, "Mutual coupling compensation in UCAs: simulations and experiment," *IEEE Trans. Antennas Propag.*, vol. 54, no. 11, pp. 3082-3086, Nov. 2006.

[45] R. E. Collin, *Antennas and Radiowave Propagation.* New York: McGraw-Hill, 1985.

Appendix A

Contiguous Partition

This appendix is aimed at proving that, given Q sub-arrays, the value of the cost function (3.5) is minimum provided that the elements belonging to each sub-array are consecutive elements of the ordered list $L = \{l_m;\ m = 1, \ldots, M;\ l_m \leq l_{m+1}\}$. With reference to a set of elements $\underline{V} = \{v_m;\ m = 1, ..., M\}$ be to be divided in Q sub-sets, the thesis to be proved is that the partition minimizing the cost function (3.5) is a *contiguous partition* (i.e., if two elements v_i and v_n belong to the same class and $v_i < v_j < v_n$, then element v_j is assigned to the same subset of elements). Towards this end, the proof follows the guidelines reported in [18]. Let us consider a *non-contiguous* partition $\underline{P}_Q = \{V_q;\ q = 1, ..., Q\}$ of the set $\underline{V}$ and three elements v_i, v_j, v_n such that $v_i < v_j < v_n$. Let elements v_i and v_n belong to a subset with mean value d_r and let v_j belong to a different subset having mean value d_s. Whatever the values of d_r and d_s, at least one the following statements holds true

$$\begin{cases} |v_j - d_s| \geq |v_j - d_r| > 0, \\ |v_i - d_r| \geq |v_i - d_s| > 0, \\ |v_n - d_r| \geq |v_n - d_s| > 0. \end{cases} \tag{A.1}$$

Let us denote with v_t the element satisfying (A.1) and its own subset as $\underline{V}_k = \{v_k;\ k = 1, ..., N_k\}$. Moreover, let us refer to the other subset as $\underline{V}_h = \{v_h;\ h = 1, ..., N_h\}$. Accordingly, the cost function (3.5) associated to the partition $\underline{P}_Q$ may be written as:

$$\Psi = \sum_{m=1}^{M} v_m^2 - N_k \cdot d_k^2 - N_h \cdot d_h^2 - \sum_{q=1;\, q \neq h,k}^{Q} N_q \cdot d_q^2 \tag{A.2}$$

N_q and d_q being the number of elements and the mean value of the q-th sub-array, respectively.
Now, let us consider a new partition $\underline{P}_Q^{(1)}$ obtained by moving the element v_t from the subset $\underline{V}_k$ to the subset $\underline{V}_h$. We obtain two new subsets $\underline{V}_k^{(1)} = \underline{V}_k \setminus \{v_t\}$ and $\underline{V}_h^{(1)} = \underline{V}_k \cup \{v_t\}$ [(4)][1] with mean values equal to $d_k^{(1)} = \frac{N_k d_k - v_t}{N_k - 1}$ and $d_h^{(1)} = \frac{N_h d_h + v_t}{N_h + 1}$,

[1] [(4)] We explicitly note that the new partition $\underline{P}_Q^{(1)}$ has the same number of subsets as $\underline{P}_Q$.

respectively. Accordingly, the cost function associated to the partition $\underline{P}_Q^{(1)}$ can be written as:

$$\Psi^{(1)} = \sum_{m=1}^{M} v_m^2 - \frac{(N_k d_k - v_t)^2}{N_k - 1} - \frac{(N_h d_h - v_t)^2}{N_h - 1} - \sum_{q=1;\, q\neq h,k}^{Q} N_q d_q^2. \tag{A.3}$$

Now, by subtracting (A.3) from (A.2), after some manipulations, it turns out that

$$\Psi - \Psi^{(1)} = \frac{N_k}{N_k - 1} (v_t - d_k)^2 - \frac{N_h}{N_h + 1} (v_t - d_h)^2 . \tag{A.4}$$

According to (A.1), $\Psi > \Psi^{(1)}$ and it can be concluded that for every *non-contiguous* partition we can find another one with the same number of subsets, but with a smaller cost. Hence, the partition minimizing the cost function (3.5) is a *contiguous* partition.

As a matter of fact, according to (A.1), the element v_t cannot be equal to the mean value d_k and thus, $\underline{V}_k$ has cardinality greater than one. It follows that the sub-set $\underline{V}_k^{(1)}$ has at least one element.

Appendix B

Dimension of the Essential Space

This section is devoted at quantifying the dimension $T^{(ess)}$ of the *essential* solution space $\Re^{(ess)} = \left\{\underline{C}_t^{(ess)};\ t = 1, ..., T^{(ess)}\right\}$, thus pointing out the computational saving allowed by the proposed approach compared to exhaustive or global sampling solution procedures. More in detail, the aim is that of determining the number $T^{(ess)}$ of candidate solutions or, in an equivalent fashion, the number of *allowed paths* in the solution tree.

Generally speaking, since a sub-array configuration $\underline{C}$ can be mathematically described by a sequence of M digits of a Q-symbols alphabet, the whole number of aggregations is equal to $T = Q^M$. Thanks to the *equivalence* relationship, the set of candidate solutions can be limited to the number of paths in a complete binary tree of depth M, thus the number of *non-redundant* solutions results $T = 2^{M-1}$. Moreover, by taking into account only *admissible* (i.e., grouping where there is at least one element in each sub-array) and *allowed* (i.e., sorted aggregations) complete sequences, the set of solution can be further reduced. With reference to the ordered list $L = \{l_m;\ m = 1, \ldots, M;\ l_m \leq l_{m+1}\}$, the *allowed* paths are mathematically described as

$$\underline{C}_t^{(ess)} = \left\{ c_{t,m}^{(ess)} \,\middle|\, c_{t,m}^{(ess)} \leq c_{t,m+1}^{(ess)},\ c_{t,1}^{(ess)} = 1,\ c_{t,M}^{(ess)} = Q \right\}, \qquad t = 1, ..., T^{(ess)}, \tag{B.1}$$

where $c_m^{(ess)}$ denotes the sub-array number to which the m-th element l_m of the ordered list L belongs.

In order to determine the essential dimension $T^{(ess)} = T^{(ess)}(Q, M)$ of the solution space, let us consider the "recursive" nature of the binary solution tree and, as a reference example, the case $Q = 2$. In such a situation, the grouping vector $\underline{C}_t^{(ess)}$ is a sequence of M symbols from the set $\{1, 2\}$ that satisfies the following constraints: (a) $c_{t,1}^{(ess)} = 1$, (b) $c_{t,M}^{(ess)} = 2$, and (c) if $c_{t,m}^{(ess)} = 2$ then $c_{t,m+1}^{(ess)} = c_{t,M}^{(ess)} = 2$. Thus, each possible solution $\underline{C}_t^{(ess)}$ is made up of a sub-sequence of consecutive symbols 1 followed by a sub-sequence of symbols 2. Accordingly, the trial solutions $\underline{C}_t^{(ess)}$, $t = 1, ..., T^{(ess)}$, are obtained by moving the starting point

of the sub-sequence of symbols 2 from $m = 2$ (being $c_1 = 1$) up to $m = M$,

$$T^{(ess)}(Q, M)\big\rfloor_{Q=2} = \begin{pmatrix} M-1 \\ 1 \end{pmatrix} = M - 1. \tag{B.2}$$

As far as the case $Q = 3$ is concerned, similar considerations hold true. In particular, each *allowed* trial solution $\underline{C}_t^{(ess)}$ ends with a sub-sequence of successive symbols 3. The number of elements of such a sub-sequence ranges from 1 to $M - 2$, leading to a complementary sub-sequence of symbols 1 and 2 of length $M - i$. Accordingly,

$$T^{(ess)}(Q, M)\big\rfloor_{Q=3} = \sum_{i=1}^{M-2} T^{(ess)}(Q, M-i)\big\rfloor_{Q=2} \tag{B.3}$$

Generalizing, since the smallest and largest number of occurrences of the symbol Q in a sequence is 1 and $M - (Q - 1)$, respectively, the essential dimension of the solution space when a M elements array is partitioned into Q sub-arrays is equal to

$$T^{(ess)}(Q, M) = \sum_{i=1}^{M-(Q-1)} T^{(ess)}(Q-1, M-i) = \begin{pmatrix} M-1 \\ Q-1 \end{pmatrix}. \tag{B.4}$$