Diamond Tiles: A Novel Modular Design Technique for Hexagonal Phased Arrays

P. Rocca, N. Anselmi, A. Polo, and A. Massa

Contents

1 Introduction

Phased array antennas have significantly improved since the first introduction, nowadays they are ubiquity in every-day life because a they are used for a lot of applications: radar, communications, remote sensing, navigation, radiostronomy, automotive, biomedical imaging, radiotherapy and a lot of research activity. Modern phased arrays permits to obtain high radiation performance and fast beam scanning, but this array are expensive. In the near future phased array will need big-scale production of economic antenna systems, so it will be necessary to reduce the overall costs of the array. In a fully populated solution there are a control point for each antenna element. Half of the cost is due to the TX/RX module, so to reduce the cost is possible to reduce the number of control points. Unconventional architecture such as sub-arrayed/clustered, thinned, or sparse arrays have been proposed instead of fully-populated array. Such sub-optimal solutions gaining more attention because of the most recent market requirements on radiation performance in modern radar and communication systems.

In this framework the exact tiling of a finite area with multiple tiles (or sub-array) of two or more radiating elements, is of great interest in nowadays phased array design. Each tile is feeded with a single control point, so there are less control point (TX/RX modules) than fully-populated architecture. It permits to reduce the number of control point by grouping two or more radiating element over a single control point, still yielding satisfactory radiation features. Unfortunately a new problem arise, because the periodicity introduced by the quantization of aperture illumination, permits the undesired high sidelobes. Therefore reducing the architectural complexity by partitioning the array aperture in sub-array (diamond tiles) of equal shapes and orientations, in the power pattern arises undesired grating lobes. To reduce unwanted lobes it is possible to use aperiodic sub-array or tiles having irregular shapes and irregular location or an aperiodic polyomino-based clustering methods. These possible solutions break the periodicity due to the quantization and reduce the level of undesired lobes.

The complete covering of a bounded region, using translated copies of tiles, without overlapping, it is well known in mathematical litterature and it is exploited for array design.

A lattice divides the plane (array aperture) in a small region called *fundamental region*. Tiling a plane means: cover the entire plane without leaving hole or overlaps by the union of at least two fundamental regions.

Tiling problems are very complex poblems, they are NP-complete problems. If simple tiles shapes are considered, it is possible to know if the aperture is completely tileable. The method presented in provides an algorithm for the generation of all existing tilings of a simply connected region using domino and lozenge tiles to cover rectangular and hexagonal aperture respectively.

On the actual state-of-art literature a recently solution is to use domino tiles (tiles with at least two square cells) which have been used for the optimization of small and large antenna aperture by using Exhaustive Tiling Method (ETM) and Genetic Algorithm Optimization Tiling Method (OTM-GA).

The first method, ETM, develops an enumerative solution by using mathematical tiling theorems and algorithms, it permits to obtain all the possible tiling configurations for a given tile shape and aperture. In particular it is possible to obtain the optimal coverage, so the optimal solution. By providing as input a fully-populated

reference solution, the algorithm gives as output a solution very close to reference one in terms of radiation performance. This algorithm works only for low/medium aperture, because when the array size increase, the possible tiling configurations increase exponentially; so it is computationally demanding to generate all the possible configurations.

The second method, namely OTM-GA (Optimization Tiling Method - Genetic Algorithm), is able to find the sub-optimal/optimal tiling configuration with an high success-rate by evaluating only a sub-set of the solution space. GA exploits a set of individuals characterized by a good genetic content in terms of radiation performance defined in according to mathematical tiling theorems and algorithms. This method is robust because radiation performance converge very close to reference ones.

Compared with the state-of-art, this thesis presents some methodological advances:

1. In tiling theorems and algorithms are used also for lozenge shaped tiles to cover hexagonal aperture. The objective is to apply ETM for low/medium hexagonal aperture array using lozenge tiles.

2. Apply OTM-GA based on *binary and integered coding* to explore a wide solution space for large hexagonal aperture by defining schemata blocks to obtain sub-optimal/optimal solution very close to reference solution.

3. Apply a synthesis methodologies on a single element to make a comparison between idel and real antenna element.

The most important novelty in this thesis is the use of hexagonal aperture arrays and OTM-GA based on integer coding:

- The hexagonal arrangements is chosen from many others geometries because it has the best steerable characteristics hence it permits to reduce the presence of grating lobes. Another reason to use hexagon is that hexagonal shape can be used to tile surfaces without leaving holes between every hexagon, in order to create an honeycomb structure. Finally in hexagonal arrays it is possible to obtain nulls more depper than circular array.

- The use of Genetic Algorithms is justified by perfomance compared with other optimization method, like Particle Swarm Optimization (PSO), GAs are also used in different scientific field. For example described some cases: in one example D. E. Golberg has developed algorithms that learns to control a gas pipeline system modeled on the one that carries natural gas from the Southwest and Northeast. In another example, L. Dawis has used similar techniques of Golberg to design communications systems, software's objective is to carry the maximum possible amount of data with the minimum number of transmission line and switches connected them.

  In literature different scientific paper that talk about genetic algorithms applied to electromagnetic field, antenna array design, tiling with GA-based strategy and the use of genetic algorithms are motivated by:

  - GAs work with discrete parameters

- In a solution space with a lot of local minimas, GAs operator as crossover and mutation permits to quickly go from a region to another
- The optimization process is able to deal with a lot of parameters, because the solution space has a lot of local minimas
- Simple to understand and program
- Useful for a large space of finite solutions, because GAs permit to obtain the optimal solution

In particular the novelty regard integer coding integer coding ensure that each chromosome is encoded with shorter strings than binary case, therefore less computational effort.

2 Variable Definitions

- Side's lenght of the domain L_d:

  Side's lenght of the lattice/geometry domain measured in λ

- Lenght of two opposite side of rectangular aperture L_x:

  Side's lenght of rectangular aperture along x axis measured in λ

- Lenght of two opposite side of rectangular aperture L_y:

  Side's lenght of rectangular aperture along y axis measured in λ

- Number of the points N_p^{tot}:

  Number of the points of the lattice/geometry

- Points along x M_p:

  Number of the points along x axis of the lattice/geometry

- Points along y N_p:

  Number of the points along y axis of the lattice/geometry of the lattice/geometry

- Total number of cells N_c^{tot}:

  total number of cells where to put the elements of the array of the lattice/geometry

- Number of cells along x M_c:

  total number of cells along x axis of the lattice/geometry

- Number of cells along y N_c:

  total number of cells along y axis of the lattice/geometry

- Number of points of the boundary $N_p^{(bound)}$:

  number of points of the array lattice/geometry

- Total numbers of elements N_{tot}:

  total number of elements which compose the array

- Number of pattern samples along u direction N_u:

  Total number of pattern samples along u direction for ETM software (must be even)

- Number of pattern samples along v direction N_v:

  Total number of pattern samples along v direction for ETM software (must be even)

- Weight of SLL w_{SLL} :

  weight of SLL for fitness function

- Weight of directivity w_D :

  weight of directivity for fitness function

- Weight of HPBW azimuth cut w_{HPBW}^{azm} :

  weight of Half Power Beamwidth for fitness function along azimuth direction

- Weight of directivityof HPBW elevation cut w_{HPBW}^{elv} :

  weight of Half Power Beamwidth for fitness function along elevation direction

- Weight of directivity w_{mask} :

  weight of mask for fitness function

- Barycenter along x B_x:

  barycenter of array along x direction

- Barycenter along y B_y:

  barycenter of array along y direction

- Variance along x σ_x:

  value of array variance along x direction

- Variance along y σ_y:

  value of array variance along y direction

- Element spacing along x d_x:

  value of the spacing between two nearest element along x direction

- Element spacing along y_1 d_{y1}:

  value of the spacing between two nearest element along y direction when two triangles has a common side

- Element spacing along y_2 d_{y2}:

  value of the spacing between two nearest element along y direction when two triangles has a common vertex

- Number of elements for cell N_{el}:

  number of elements for each cell of lattice/geometry

- Pointing Direction of θ angle θ_0 :

  pointing direction of main beam along θ

- Pointing Direction of ϕ angle ϕ_0 :

  pointing direction of main beam along ϕ

- Pointing Direction of $u = sen\theta cos\phi$ coordinate u_0 :

  pointing direction of main beam along u

- Pointing Direction of $v = sen\theta sen\phi$ coordinate v_0 :

  pointing direction of main beam along v

- Length of A side a :

  length of two opposite sides A of hexagon measured as adjacent cells

- Length of B side b :

  length of two opposite sides B of hexagon measured as adjacent cells

- Length of C side c :

  length of two opposite sides C of hexagon measured as adjacent cells

- Length of A side L_a :

  length of two opposite sides A of hexagon measured in λ

- Length of B side L_b :

  length of two opposite sides B of hexagon measured in λ

- Length of C side L_c :

  length of two opposite sides C of hexagon measured in λ

- Tiling configurations T :

  number of possible tiling configurations for an exhaustive generation

- Number of unknows N_u:

  unknows are the inner points of the array

- Maximum of word max U_{max}:

  upper bound of the word max

- Number of chromosome bits N_{ch}:

  lenght of the chromosome

- Number of bits of integer coding N_{bit}:

  number of bits to coding integer values in binary values

- Number of trials (seed) N_{seed}:

  number of trials (seed) launched for every simulation, useful to generate pseudo-random numbers

- Number of individuals N_I:

  population dimension, it is the set of trial solution, equal to the double of the number of array lattice internal points

- Number of flips N_{flips}:

  number of flips (rotation) of 180 $[deg]$ of a group of lozenges with a local minimum/maximum in their center

- Number of schemata N_{sch}:

  number of template that identifies a subset of strings with similarities at certain fixed string position

- Cross-Over probability p_{cx}:

  probability to obtain the best child by using the good features of the current trial solutions

- Mutation probability $p_m = 0.01$

  probability to introduce new features

- Diversity Percentage - $d_{\%}$

  percentage of bits that every word must have different between each word

3 Mathematical Formulation

3.1 Array Geometry

Consider a planar phased array radar with hexagonal aperture discretized into equilateral triangular elementary cells where the aperture dimension is the number of adjacent triangle for evey side of the hexagon, the number adjacent triangle for each side is indicated as $a,\ b,\ c,\ d,\ e,\ f$, the effective lenght ($L_{a,b,c,d,e,f}$) of each side is equal to the product between the lenght of side of triangular cell (L) and the number of adjacent triangles $a,\ b,\ c,\ d,\ e,\ f$ to every side:

$$L_\gamma = L \times \gamma,\ \gamma \in (a,\ b,\ c,\ d,\ e,\ f) \tag{1}$$

Every cell contains in its barycenter (x_n, y_n) a radiating element. The arrangement of radianting elements is due to the shape of elementary cells in which the aperture is discretized. In particular, triangles are disposed upward and downward, so there are three different inter-element spacing to consider: $d_x,\ d_{y1},\ d_{y2}$[Fig.1]. To calculate spacing we have to consider that the spacing along x-axis (d_x) is constant between each element, but along y-axis is different between elements that belongs to a couple of triangles with common side (d_{y1}) or with one common vertex (d_{y2})[Fig.1], then cell barycenter coordinates are: $x_n = \frac{L}{2},\ y_n = \frac{L}{3} \cdot \frac{\sqrt{3}}{2}$, after this considerations the inter-element spacing value are:

$$d_x = \frac{L}{2} \tag{2}$$

$$d_{y1} = \frac{L}{\sqrt{3}} \tag{3}$$

$$d_{y2} = L \times \frac{2}{\sqrt{3}} \tag{4}$$

To calculate the aperture dimension, given the inter-element spacing and the number of triangle for each side, let start from the highest spacing d_{y2}. From d_{y2} 4 calculate L with the inverse formula:

$$L = d_{y2} \times \frac{\sqrt{3}}{2} \tag{5}$$

then calculate the side leght of the hexagon using 1.

As example, if we consider $d_{y2} \leq 0.5\lambda$, the side lenght obtained is $L \leq 0.43\lambda$, then the number of adjacent triangles are: $a = b = c = d = e = f = 10$, so the side lenght is: $L_\gamma = 0.43\lambda \times 10 = 4.3\lambda,\ \gamma \in (a,\ b,\ c,\ d,\ e,\ f)$.

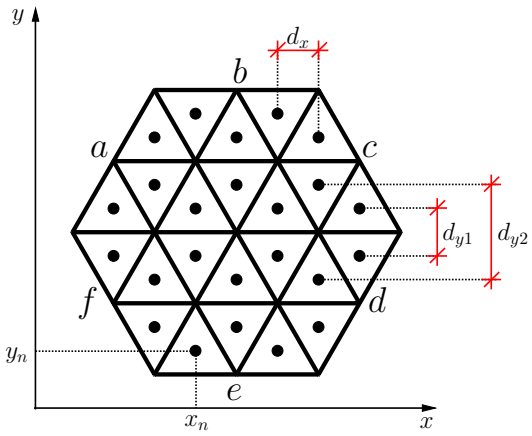

Figure 1: Array Lattice - Intra Element Distances

Pile of cubes: Hexagon as a 3D box

An hexagon can be seen as a big 3-dimensional cube with small 3D cubes inside, where each lozenge correspond to a small part inside the entire 3D box, therefore the lozenges are pile of cubes inside the hexagon [Fig.2]. The hexagon with side lenght (a, b, c, a, b, c) can be seen as a 3D box with dimension $a \times b \times c$ composed by vertical and horizontal lozenges. So states that if the hexagon is regular ($a = b = c = n$) the number of diamond for each one of the three possible orientation is n^2 so the total number of diamond is $3n^2$. If hexagon is irregular ($a \neq b \neq c$) the 3D structure can be seen as a parallelepiped. For every face the number of lozenge of a fixed orientation is:

- number of array elements:

$$N_{el} = 2 \times (a \times c + b \times c + a \times b) \tag{6}$$

- number of vertical diamond:

$$N_{\sigma^V} = a \times c \tag{7}$$

- number of horizontal left diamond:

$$N_{\sigma^{H_{left}}} = b \times c \tag{8}$$

- number of horizontal right diamond:

$$N_{\sigma^{H_{right}}} = a \times b \tag{9}$$

- total number of diamond is:

$$N_{diamond} = N_{\sigma^V} + N_{\sigma^{H_{left}}} + N_{\sigma^{H_{right}}} = \frac{N_{el}}{2} \tag{10}$$

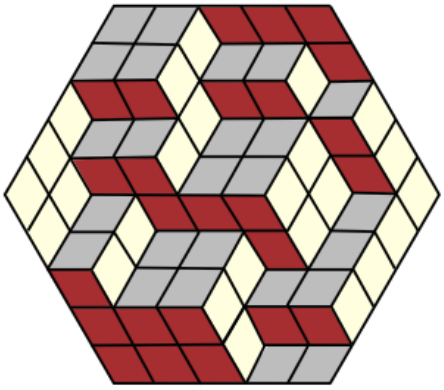

Figure 2: Pile of cubes example

3.2 Tile-Based Sub-Array Architecture

Starting from a fully populated case fills by N radiating elements Fig.3, where every element is positioned in the barycenter of elementary cells (Sec.3.1), the amplitude and phase of each element is controlled by a single T/R module (N control points), but in the architecture described in this section the objective is to reduce the number of control points, by clustering two elements that belong to adjacent cells.

The elementary cells of this type of tiling are equilateral triangles, the combination of a couple of elementary cells those share a side and they permit to obtain a diamond shaped tile, this shape has internal angles of 120 degrees and 60 degrees. There are three different types of orientations: vertical $\left(\sigma^{V}\right)$, horizontal-left $\left(\sigma^{H_{left}}\right)$ and horizontal-right $\left(\sigma^{H_{right}}\right)$ [Fig.4-5]. Therefore the sub-array architecture is composed by $\frac{N}{2}$ control points composed by amplitude attenuator (α_q) and phase shifter (β_q)[Fig.6].

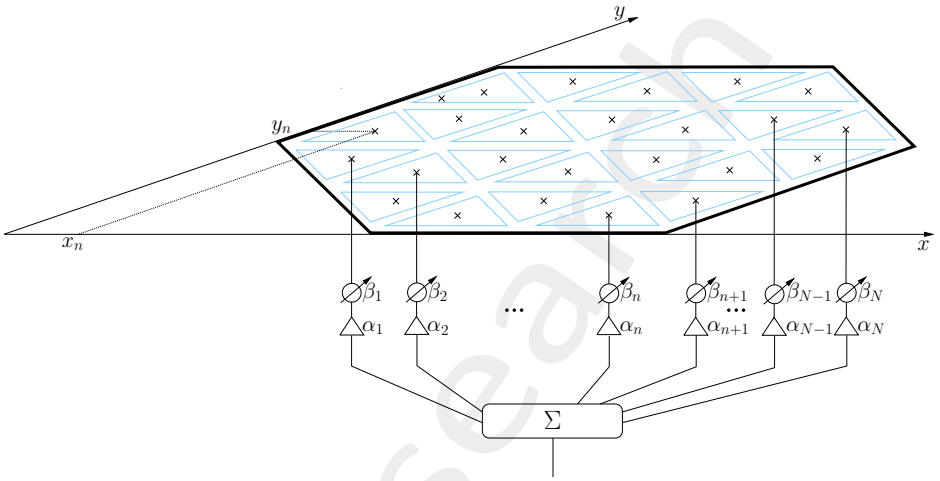

Figure 3: Sketch of hexagonal aperture of fully populated architecture

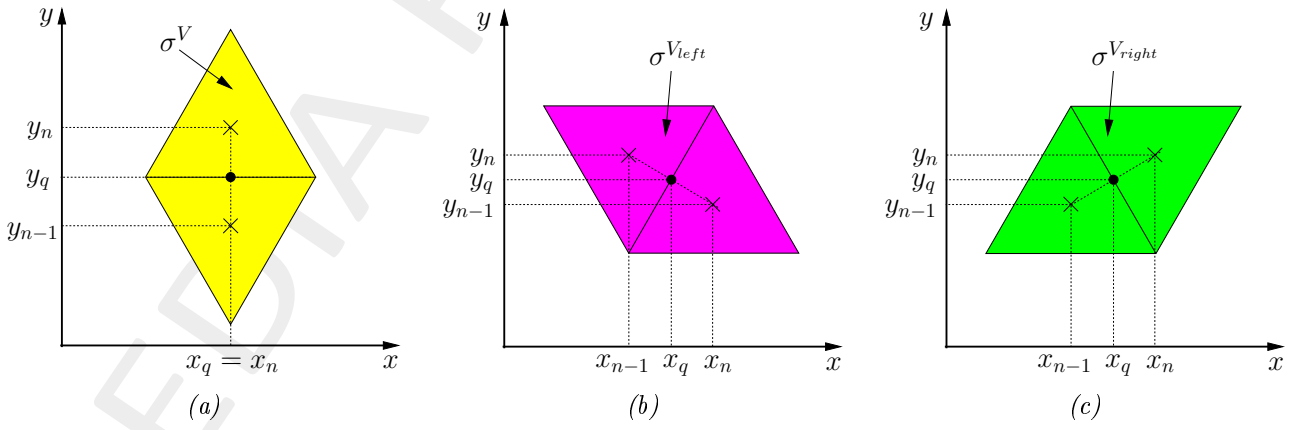

Figure 4: Diamond-like tile: (a) Vertical, (b) Horizontal left, (c) Horizontal right

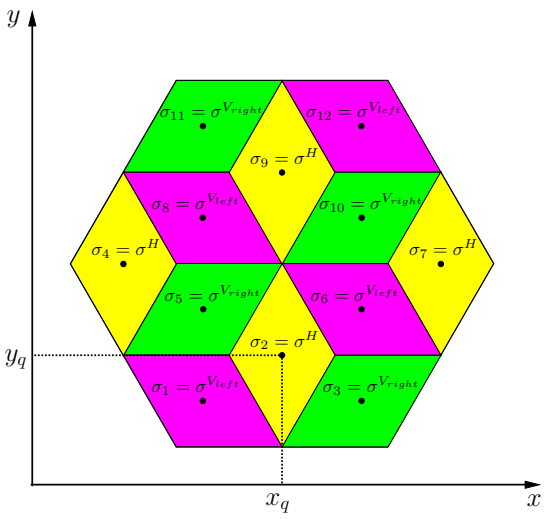

Figure 5: Array Aperture Tiling with $\mathbf{c} = \{1,\ 1,\ 2,\ 3,\ 3,\ 4,\ 5,\ 5,\ 2,\ 6,\ 6,\ 7,\ 4,\ 8,\ 8,\ 9,\ 10,\ 10,\ 7,\ 11,\ 11,\ 9,\ 12,\ 12\}$ and $\sigma = \{\sigma^{V_{left}},\ \sigma^H,\ \sigma^{V_{right}},\ \sigma^H,\ \sigma^{V_{right}},\ \sigma^{V_{left}},\ \sigma^H,\ \sigma^{V_{left}},\ \sigma^H,\ \sigma^{V_{right}},\ \sigma^{V_{right}},\ \sigma^{V_{left}}\}$, being $a = 2,\ b = 2,\ c = 2$ and $Q = 12$

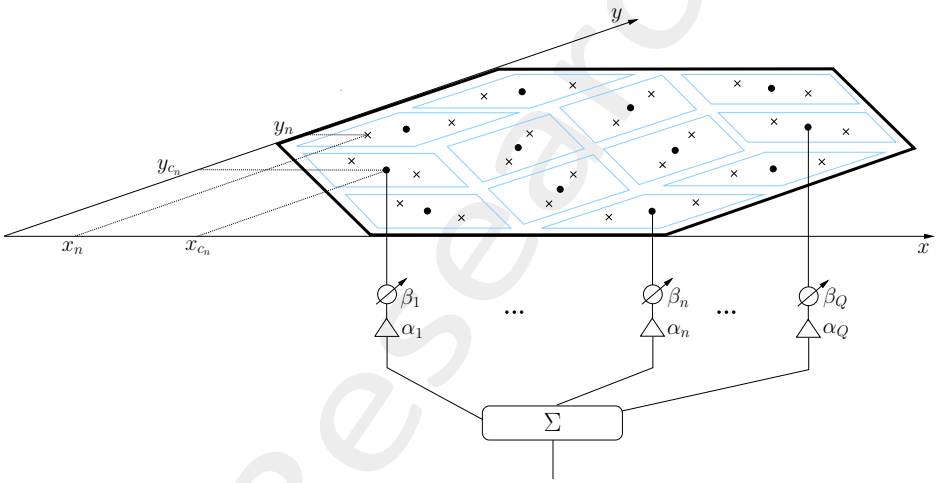

Figure 6: Sketch of hexagonal aperture of sub-array architecture with feeding network

The locations and orientations of the elementary diamond-shaped tile must be properly optimized to yield the maximum (total) converage of the array aperture without leaving hole and to obtain an irregular sub-array arrangement for minimazing the level of "quantization lobes".

Hence we have move from a fully populated architecture to a clustered architecture. This solution has the advantage to reduce the cost, due to decreasing number of control points, but there is a drawback due to deterioration of radiation properties, because less control points mean less degree of freedom to synthesize the array.

The array factor of clustered array with element spacing $d_x,\ d_{y1}\ and\ d_{y2}$ is:

$$AF(\theta,\phi) \quad = \quad \sum_{n=1}^{N} I_n e^{jk(x_n sin\theta cos\phi + y_n sin\theta sin\phi)} \tag{11}$$

with $(x_n, y_n)\ n = 1, ..., N$ are the barycenter of a single array element.

where $k = \frac{2\pi}{\lambda}$ is the wave number, λ is the wavelenght, I_n is the cluster excitations:

$$I_n = \sum_{q=1}^{Q} \alpha_q e^{j\beta_q} \delta_{c_n q},\ n = 1, ..., N \tag{12}$$

where Q is the number of subarray that covering totally or partially the array surface, α_q and β_q are the $q-th$ sub-array amplitude and phase coefficients, $c_n \in [1, Q]$, $n = 1, ..., N$ is the membership of each $n-th$ element of the array to one of the Q subarrays, $\delta_{c_n q}$ is the Kronecker delta function equal to $\delta_{c_n q} = 1$ when the $n-th$ element belong to the $q-th$ sub-array σ_q ($c_n = q$), while $\delta_{c_n q} = 0$ otherwise ($c_n \neq q$). The clustered amplitudes α_q is the average of amplitudes $\alpha_n^{(REF)}$, $n = 1, ..., N$ of the reference/ideal fully-populated array that belongs to the same cluster:

$$\alpha_q = \frac{\sum_{n=1}^{N} \alpha_n^{ref} \delta_{c_n q}}{D_q},\ q = 1, ..., Q \tag{13}$$

where

$$D_q = \sum_{n=1}^{N} \delta_{c_n q},\ q = 1, ..., Q \tag{14}$$

The corresponding phases are:

$$\beta_q = -k(x_q sen\theta_0 cos\phi_0 + y_q sen\theta_0 sen\phi_0),\ q = 1, ..., Q \tag{15}$$

where θ_0 and ϕ_0 are the pointing direction angles and the diamonds barycenter positions inside the aperture are calculated based on barycenter coordinates along x (x_n)and along y (y_n) of elementary cells that belong to the tile [Fig.6]:

$$x_q = \frac{1}{D_q} \sum_{n=1}^{N} x_n \delta_{c_n q} \ \ and \ \ y_q = \frac{1}{D_q} \sum_{n=1}^{N} x_n \delta_{c_n q},\ q = 1, ..., Q \tag{16}$$

From (11) the power pattern with isotropic radiating elements is:

$$P(\theta, \phi) = |AF(\theta, \phi)|^2 \tag{17}$$

If we consider real radiating elements, in order to have a close approximation of the real antenna array pattern, we have to consider the average embedded/active element pattern $|EP(\theta, \phi)|$ for all radiating element, so the power pattern become:

$$P_{real}(\theta, \phi) = |EP(\theta, \phi) \times AF_n(\theta, \phi)|^2 \tag{18}$$

3.3 Tiling Theory and Theorems

In this section, the tiling theorems and theory regarding the exact coverage of hexagonal regions with diamond tiles, are reported.

Tilability Condition: Is the region tileable?

First of all is important to define when it is possible to tile an hexagonal region. Saldanha in states that the necessary condition for tileability is: "*the black and white triangles that compose the hexagonal region* (like Fig.9) *must be equal in number*", therefore the strong condition is:
"**Theorem 1:** *Let R be a triangulated simply connected bounded region and Σ a finite set of tile shapes. Let G be the group with generators corresponding to edges and relations given by boundaries of elements of Σ. Then a necessary condition for the tileability of R by translates of elements of Σ is that the word induced by the boundary of R is trivial in G*".

Helfgott in states the followind theorem:
"The necessary and sufficient condition for the existence of hexagon with side length $(a,\ b,\ c,\ d,\ e,\ f)$ is that the parameters be nonnegative integers satisfying a $a - d = c - f = e - b$. The number of upward pointing triangles minus the number of downward pointing triangles in an $(a,\ b,\ c,\ d,\ e,\ f)$ hexagon is a $a - d$, since every lozenge covers one upward pointing triangle and one downward pointing triangle, an $(a,\ b,\ c,\ d,\ e,\ f)$ hexagon can be tiled by lozenges only if $a = d$ and this implies that that the hexagon is an $(a,\ b,\ c)$ hexagon. $a,\ b,\ c$ is the length of opposite side of hexagon."

Cardinality of the solutions space

It is important to understand how many configuration is possible to generate. From a semiregular hexagon with side-lengths $a,\ b,\ c,\ a,\ b,\ c$ can be tiled by lozenges according with the formula:

$$\prod_{i=1}^{a}\prod_{j=1}^{b}\prod_{k=1}^{c}\frac{i+j+k-1}{i+j+k-2} \tag{19}$$

I have calculate the number of configurations assuming $a = b = c$ from $(a,\ b,\ c) = (1,\ 1,\ 1)$ to $(a,\ b,\ c) = (10,\ 10,\ 10)$ as shown in Tab. 2 and Fig. 7.

(a, b, c)	**T**
1	2
2	20
3	980
4	232848
5	267227532
6	$1.47861942 \times 10^{12}$
7	$3.94059963 \times 10^{16}$
8	$5.05516068 \times 10^{21}$
9	$3.12034478 \times 10^{27}$
10	$9.26503772 \times 10^{33}$

Table 2: $(a,\ b,\ c)$ *vs.* T, *number of tiling configurations*

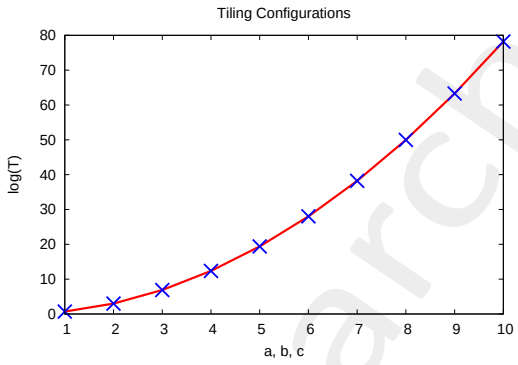

Figure 7: $(a,\ b,\ c)$ *vs.* T, *number of tiling configurations*

Fig. 7 shows that when a, b, c increase, the number of tiling configurations increases exponentially.

3.4 Array Tiling Synthesis Problem

Given an array of N elements positioned at the barycenter of equilateral triangular unit cells combined into diamond-like tiles, within an hexagonal shaped aperture, find the optimal tiling configuration $\underline{c}$ and the corresponding sub-array weights $\underline{\alpha}$, $\underline{\beta}$ such that the radiated pattern fits user-defined requirements $\Phi\left(\underline{c},\ \underline{\alpha},\ \underline{\beta}\right)$ with the main lobe steered at $\theta_0,\ \phi_0$.

The starting array to cluster is a fully populated architecture with reference excitation amplitudes have been synthesized through an optimal Convex Programming (CP)-based method, with array weights $\underline{\alpha}$, $\underline{\beta}$ that fits the reference mask $M\left(\theta,\phi;\underline{c}\right)$, while the clustered architecture is a sub-optimal solution compared to reference one. The goal is to obtain clustered power patter very close to reference one, to do this the only DoFs is tiling configuration. Therefore the objective is to find the tile locations and orientations that minimize the power pattern area of clustered array outside the power pattern mask $M(\theta,\phi)$. The mismatch between the reference and tiled amplitude and phase weights is called **mask matching** (green area of Fig.8) and it is represented with the following cost function (eq. 20):

$$\Phi(\underline{c},\ \underline{\alpha},\ \underline{\beta}) = \int_{-1}^{1}\int_{-1}^{1} |P(\theta,\phi;\underline{c}) - M(\theta,\phi)| \cdot H[P(\theta,\phi;\underline{c}) - M(\theta,\phi)]d\theta d\phi \tag{20}$$

where $P\left(\theta,\phi;\underline{c}\right)$ is the clustered power pattern defined in (17) or (18), $M\left(\theta,\phi;\underline{c}\right)$ is the reference mask, $\underline{\alpha} = \{\alpha_q;\ q = 1,\ldots,Q\}$ is the amplitude coefficient vector, $\underline{\beta} = \{\beta_q;\ q = 1,\ldots,Q\}$is the weight coefficient vector, $\underline{c} = \{c_n;\ n = 1,\ldots,N\}$ is the clustered vector and $H(\cdot)$ is the Heaviside function (step function):

$$H(\underline{x}) = \begin{cases} 1\ if\ x > 0 \\ 0\ otherwise \end{cases}$$

The objective is mask matching minimization, it means minimize the cost function $\Phi(\underline{c},\ \underline{\alpha},\ \underline{\beta})$ and the only degree of freedom to fit the design constraints defined by the power pattern mask is the tile positions within the antenna aperture, defined by the tiling vector $\mathbf{C}$; this implies that the optimal solution depends on tiling configuration, in particular the best tiling solution is that minimize the cost function:

$$\mathbf{C}^{opt} = arg(min_{t=1,\ldots,T}\{\Phi(\underline{c},\ \underline{\alpha},\ \underline{\beta})\}) \tag{21}$$

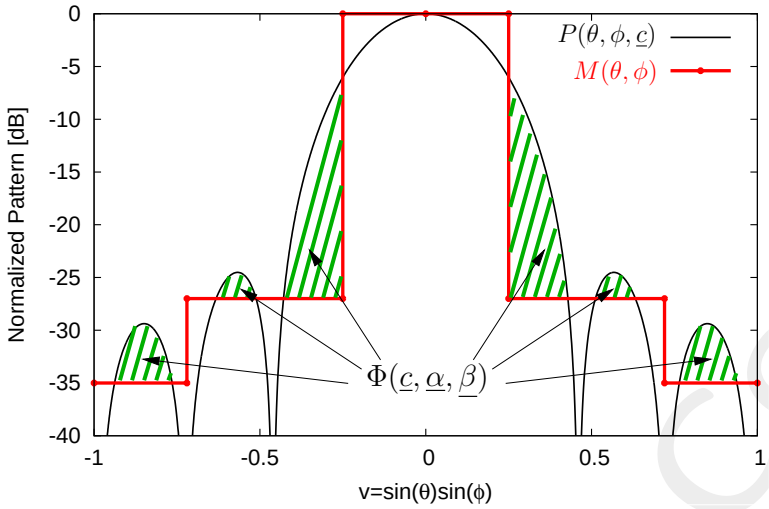

Figure 8: Example of pattern and mask to minimize the cost function

3.5 Tiling Methodologies

3.5.1 Enumerative Tiling Method (ETM)

If the aperture considered is completely tileable and it has small size, it is possible to generate an exhaustive analysis of the solution space. So it is possible to use ETM to generate all the combinations to find the optimal solution for finding the optimal tiling. A key-issue is the generation of all tiling configurations, to do this, the approach described in solves this problem. It uses the *height function* to univocally identify a generic *t-th* solution $\mathbf{c}^{(t)}$ $(t = 1, ..., T)$ and it describes diamond tiles organization $(\sigma^V,\ \sigma^{H_{left}},\ \sigma^{H_{right}})$ within the array aperture A.

Height Function Computation

To define the heigh function $h(\cdot)$, it is uselful to describe before the array aperture A. It is composed by N elements. Every elements is inside a pixel defined as vertexs $\{v_n;\ n = 0, ..., N\}$ and
edges $\{e_{n\to n\pm 1},\ e_{n\to n\pm n_x^{lattice}},\ e_{n\to n\pm n_x^{lattice}+1},\ n = 0, ..., N\}$. Every pixel is colored in black or white like a checkerboard pattern. Black means that the edges for that pixel are oriented counterclockwise, white means clockwise (Fig. 9). The height function is defined on the pixel vertexes $\{h_n = h(v_n),\ n = 0, ..., N\}$, while the *h-values* are calculated on the edge orientations.

The algorithm used to calculate *h-values* is divided in two step:

- *Step 1: Computation of the h-value of the boundary vertices of A*

  Lozenge case described in is equal to domino case, so consider only the vertexes on the boudary aperture $(\mathbf{v}_{ext}^{(t)} \in \partial A,\ \mathbf{v}_{ext}^{(t)} = \{v_n^{(t)},\ n = 1, ..., N_{ext}\})$, starting from an origin vertex $v_0 \in \mathbf{v}_{ext}^{(t)}$ which height function is set to $h(v_0) = 0$ and moving along ∂A clockwise, two cases have to be considered:

  - If edge belongs to a white cell the height function value of the next vertex v_{n+1} is $h^{(t)}(v_{n+1}) = h^{(t)}(v_n) + 1$
  - If edge belongs to a black cell the height function value of the next vertex v_{n+1} is $h^{(t)}(v_{n+1}) = h^{(t)}(v_n) - 1$

  The height function value on the boundary is independently on the tiling configuration $(\forall t \in [1, T])$. At the end of computation the last vertex is the starting vertex, v_0, its height function value is equal to the starting value $h(v_0) = 0$.

- *Step 2: Computation of the h-value of the internal vertices of A*

  Select a vertex that belongs to the set of internal vertexes $v_n^{(t)} \in \mathbf{v}_{int}^{(t)},\ \mathbf{v}_{int}^{(t)} = \{v_n^{(t)},\ n = 1, ..., N - N_{ext}\}$, with a neighbor vertex $\mathbf{v}_p^{(t)} \in [v_{n+n_x^{lattice}}^{(t)},\ v_{n+n_x^{lattice}-1}^{(t)},\ v_{n-n_x^{lattice}-1}^{(t)}]$ which has height function already set $h_p^{(t)} = h^{(t)}(v_p)$. Every tile is composed by two pixels, black and white respectively, so the value of all internal vertexes depend on pixel colour. Starting from vertex v_p with know height function $h^{(t)}(v_p) = h_p^{(t)}$, the height function of neighbor vertexes are determined with the following rule:

– if cell is white turns around it in a clockwise direction and there are two different case:

  * if cell edge belongs to tile edge, the height function of next vertex v_{p+1} is equal to domino case: $h_{p+1}^{(t)} = h_p^{(t)} + 1$.
  * if cell edge doesn't belong to tile edge (it means that cell edge is the tile central axis) the height function of v_{p+1} is different to domino case, for lozenge states: $h_{p+1}^{(t)} = h_p^{(t)} - 2$.

– if cell is black turns around it in a counterclockwise direction and there are two different case:

  * if cell edge belongs to tile edge, the height function of next vertex v_{p+1} is equal to domino case: $h_{p+1}^{(t)} = h_p^{(t)} - 1$.
  * if cell edge doesn't belong to tile edge (it means that cell edge is the tile central axis) the height function of v_{p+1} is different to domino case, for lozenge states: $h_{p+1}^{(t)} = h_p^{(t)} + 2$.

Iterate the algorithm for all internal vertexes $v_n^{(t)} \in \mathbf{v}_{int}^{(t)}$, $\mathbf{v}_{int}^{(t)} = \{v_n^{(t)},\ n = 1, ..., N\}$.

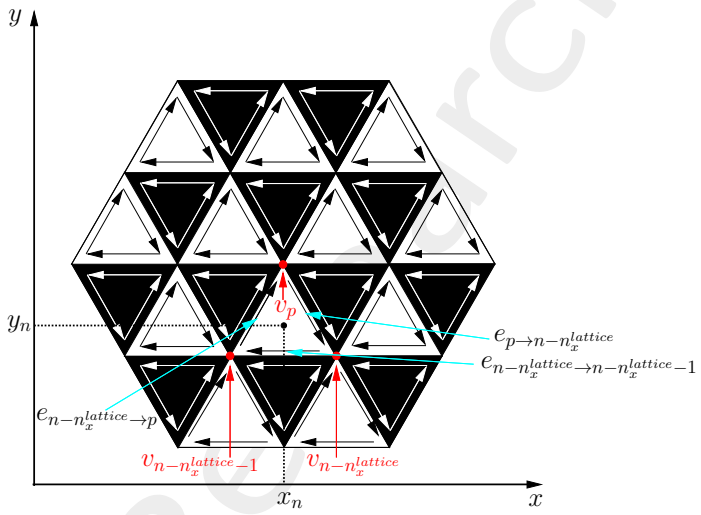

Figure 9: Black and white checkerboard representation of the array aperture A with pixel vertices v_n, $n = 1, ..., N$ and edges $e_{n \to n \pm 1}$, $e_{n \to n \pm n_x^{lattice}}$, $e_{n \to n \pm n_x^{lattice} + 1}$, $n = 0, ..., N$

Enumerative Tiling Method (ETM)

This method is used when the aperture dimension is small.

Starting from the height funtion, the procedure to generate the complete set of tiling configurations is based on the definition of tiling words: $\mathbf{w}^{(t)} = \{w_l^{(t)} : l = 1, ..., L\}$ $(t = 1, ..., T)$. Each one correspond to a configuration, $\mathbf{c}^{(t)}$, and the lenght of the word is equal to the number of internal vertex ($N_{int} = N - N_{ext}$) of the aperture A. Every letter of the word is an integer value calculate from the height function, it is different to domino case, for lozenge is stated is:

$$w_l = \frac{h_n^{(t)} - h_n^{min}}{3}, l = 1, ..., L;\ t = 1, ..., T \tag{22}$$

where $h_n^{(t)}$ is the height function of the t-th configuration ($\mathbf{c}^{(t)}$) for n-th vertex (v_n) and h_n^{min} is the height function of the minimal tiling ($\mathbf{c}^{min}$) for n-th vertex (v_n).

The **minimal tiling** $\mathbf{c}^{min} = \mathbf{c}^{(1)}$ is the first configuration; it is obtained when the height function has no maximum value, except on the aperture boundary ∂A. The algorithm is described by Thurston in, it is the same described in used for domino:

- *Step 1: Vertex Selection*

  Find the boundary vertex, $v_n^{(1)} \in \mathbf{v}_{ext}^{(1)},\ n = 1, ..., N_{ext}$, with the maximum height function value: $v_n^{(1)} = arg\left\{max_{n=1,...,N_{ext}}\left[h^{(1)}(v_n);\ v_n \in \mathbf{v}_{ext}^{(1)}\right]\right\}$.

- *Step 2: Diamond Tile Placement*

  Place a diamond tile, σ^V, $\sigma^{H_{left}}$ or $\sigma^{H_{right}}$, in order to cover the maximum height function value of the boundary and without add local maxima inside the aperture A. There is only a way to do this: the adjacent vertex of $v_n^{(1)}$ have to correspond with those of the newly placed tile.

- *Step 3: Update boundary, aperture and h-value*

  When tile is placed, compute the height function of internal vertex covered, in according with algorithm described in Sec.3.5.1. Then update the aperture boundary $\partial A \leftarrow \partial\left(A - \sigma^{V/H_{left}/H_{right}}\right)$ and the aperture $A \leftarrow \left(A - \sigma^{V/H_{left}/H_{right}}\right)$.

- *Step 4: Stop Criterion*

  Continue to iterate the algorithm from step 1 to 3 until all the aperture is covered and the height functions of all internal vertexes are calculated.

The coding of minimal tiling word is always $\mathbf{w}^{(1)} = \mathbf{0}$ because $h_n^{(t)} = h_n^{min},\ n = 1, ..., N_{int}$.

The same algorithm is applied for calculating the **maximal tiling** configuration as for domino tile. It is obtained when the height function has no minimum value, except on the aperture boundary ∂A. Therefore the difference is in *Step 1 - Vertex Selection.* Because to obtain in the maximal configuration the starting vertex is $v_n^{(1)} \in \mathbf{v}_{ext}^{(1)},\ n = 1, ..., N_{ext}$ with the minimum height function value: $v_n^{(1)} = arg\left\{max_{n=1,...,N_{ext}}\left[h^{(1)}(v_n);\ v_n \in \mathbf{v}_{ext}^{(1)}\right]\right\}$.

Thurston algorithm permits to construct tiling configuration only when there are local maximum on ∂A. To generate all the aperture tiling, Thurston algorithm have to be reinterpret in order to generate tiling starting from local maxima on the interior of A. To solution used exploits Birkhoff's representation theorem for finite distributive lattices.

Consider a set of vertexes $\mathbf{v}_s^{(t)} \in \mathbf{v}_{int}^{(t)},\ \mathbf{v}_s^{(t)} = \{v_s^{(t)},\ s = 1, ..., M;\ M \leq N_{int}\}$ inside the aperture A on which the height function of $\mathbf{v}_s^{(t)}$ reachs a local maximum. The set $\mathbf{v}_s^{(t)}$ have to contains at least one vertex, in this set by applying a downward/upward flip permits to obtain a new tiling configuration. If $\mathbf{v}_s^{(t)}$ contains more than one vertex, it is viewes as a collection of meet-irreducible elements.

This ideas is formalized with **Birkhoff's representation theorem:** *Any finite lattice distributive is isomorphic to the lattice of the ideals of the order of its meet-irreducible elements.*

This theorem permits to characterize the tiling as a collection of meet-irreducible elements if and only if its height function admits exactly one local maximum in the interior of A. So the minimal tiling obtained with Thurston algorithm is the empty case.

Now the idea is to used a generalized version of Thurston Algorithm to generate all the tiling configurations. The objective is to cover the internal vertexes of A such that doesn't appear local maximum in the height function. The solution proposed in uses the meet-irreducible elements with only a vertex with a local maximum on height function value. This solution undertake that for each meet-irreducible element exist at least a tiling with a local maximum outside the region of meet-irreducible element.

The exhaustive generation algorithm for diamond and it is very similar with domino case described in is divided in. The algorithm is the following:

- *Step 1: Select height function minimum and update tiling word*

  Determine the local minimum of height function by scanning the associated tiling word $\mathbf{w}^{(t)}$ in backward direction starting from the last letter to the first one: $h_{i-1}^{(t)} > h_i^{(t)},\ i \in [2, ..., N_{int} - 1]$, so the local minimum is in *i-th* position. Then update the word by appling an upward flip in *i-th* position to obtain a new word $\mathbf{w}^{(t)}$as follows:

$$w_n^{(t+1)} = \begin{cases} w_n^{(t)} & n = 1, ..., i-1 \\ w_n^{(t)} + 1 & n = i \end{cases} \tag{23}$$

  If no vertex is found, it means that $\mathbf{w}^{(t)}$ encodes the maximal tiling word.

- *Step 2: Update height function*

  Compute the new height function value from the updated tiling word $\mathbf{w}^{(t+1)}$for the first *i-th* inner vertexes, it is different from that described in for domino tile. In this case for lozenge is describe in: $h_n^{(t+1)} = 3 \cdot w_n^{(t+1)} + h_n^{(1)},\ n = 1, ..., i$

- *Step 3: Feasibility check*

  Check the height function difference between the n-th vertex its neighbor, it is different from that described in for domino tile. In this case for lozenge is describe in: if the condition$\left|h_n^{(t)} - h_p^{(t)}\right| = \{1; 2\}$ is true continue with the next step, otherwise go to *Step 1 - Select height function minimum and update tiling word*

- *Step 4: New tiling generation*

  Place a tile inside the aperture A in according with rule *Step 2 - Computation of the h-value of the internal vertices of A* of height function computation algorithm, then cover all the aperture following

the algorithm used to generate minimal tiling, compute height function value $h_n^{(t)},\ n = 1, ..., N_{int}$ then compute the remaining letters of $w_n^{(t)},\ n = i+1, ..., N_{int}$ using the rule 22.

- *Step 5: Stopping criterion*

  If $t = T-1$ stop the tiling generation, because all the possible tiling configurations are generated; otherwise go to *Step 1 - Select height function minimum and update tiling word*

Finally the optimal solution is found by selecting the configuration that fits the requirements (i.e.: minimum SLL, minimum mask matching, ...). More details state in Sec.3.4.

3.5.2 Binary-GA Optimization Tiling Method (OTM-BGA)

When the aperture is large, the aperture tiling is obtained through an innovative binary GA. This solution exploits "schemata" to explore the solution space for enabling the array syntesis. The key-points of GAs those justified its effective/profitable are: GA-Schemata, GA-Implicit Parallelism and GA-coding. These three key-points are here described:

- *GA-Schemata:*

  Schemata is a template that describes chromosome subset with similarities at certain positions and schemata associated to "good" individuals reproduce faster during the optimization process. To generate the schemata, in an important theorem is stated:

  "The Schemata Theorem: the expected number of schemata H at generation $t+1$ when using a canonical GA with serial operator (crossover and mutation) is:

$$E\left[m\left(H,t+1\right)\right] \geq \frac{m\left(H,t\right)}{\overline{\Phi\left(t\right)}}\left\{1-p_c\frac{\delta\left(H\right)}{1-l}p_{diff}\left(H,t\right)-o\left(H\right)p_m\right\} \qquad " \tag{24}$$

  where:

  - $E\left[\cdot\right]$ = expectation of number of individuals for schemata H at iteration $t+1$
  - $m\left(H,t\right)$ = number of instances for schemata H at iteration t
  - $\overline{\Phi\left(t\right)}$ = mean fitness of individuals in the population
  - p_c = crossover probability
  - p_m = mutation probability
  - $\delta\left(H\right)$ = schemata lenght
  - l = schemata string lenght
  - p_{diff} = probability that a parent doesn't match with schemata H
  - $o\left(H\right)$ = schema order

- *GA-Implicit Parallelism:*

  implicit parallelism refers to the fact that every generation of genetic algorithm not just deals with n individuals, but the Genetic Algorithm actually manages about $O\left(n^3\right)$ models. A single population member simultaneously belongs to a plethora of schemata.

  From the definition of schemata and implicit parallelism it is possible to yield the population of genetic algorithm, but it is difficult to obtain good schemata from a random generation of initial population for a

large array. Because schemata depend on population and to obtain a good population and a good schemata they have to be chosen from the complete set of tiling word, but if the array is large it is computationally infeasible to generate all the set. So the solution used is a random generation of the population as for domino tile. To generate the words (population), the rules used for lozenge tile, are the same used for domino tile:

- *Rule 1:*

  the difference between two consecutive letters in the same word is: $w_l^{(t)} - w_{l+1}^{(t)} = \{0, \pm 1\}$, $l = 1, ..., N_{int}$ and the difference between the same letters of two consecutive tiling word is: $w_l^{(t)} - w_l^{(t+1)} = \{0, \pm 1\}$, $l = 1, ..., N_{int}$

- *Rule 2:*

  prove that the letters of the maximal tiling word $\mathbf{w}^{(T)}$ with the same value belongs to connected regions over A

- *Rule 3:*

  the *minimal tiling word* correspond to word $\mathbf{w}^{(1)} = \mathbf{0}$ and the *maximal tiling word* has all letters grater/equal than zero: $\mathbf{w}^{(T)} = \left\{w_l^{(T)},\ l = 1, ..., N_{int}\right\}$, $w_l^{(T)} \geq 0$; therefore all the tiling word between word mininimum word and maximum word are positive: $w_l^{(1)} \leq w_l^{(t)} \leq w_l^{(T)}$, $w_l^{(t)} \geq 0$, $l = 1, ..., N_{int}$

- *GA-Coding:*

  the computational burden and the cardinality of the solution space is the same described for domino tile:

  - *computational burden:*

    $\triangle\tau_\Phi \times I \times U$ where $\triangle\tau_\Phi$ is the CPU-time for a single cost function evaluation, I is the number of iteration and U is the population dimension

  - *cardinality of the solution space:*

    it depends on the number of unknows. To reduce the number of unknowns it is preferable to use the tiling word $w_l^{(t)}$, $l = 1, ..., N_{int}$ rather than the tiling configuration $c_n^{(t)}$, $n = 1, ..., N$, because the number of array elements are less than number of internal lattice point of array: $N_{int} \leq N$.

By exploiting such guidelines, equal to the case described in, the following innovative optimization strategy has been implemented based on GA with binary string:

- *Step 1: Population Initialization*

  The first individual $u^{(1)}$ is the *minimal tiling word* $\mathbf{w}_1^{(u)} = \mathbf{w}^{(1)}$ and the last individual is the *maximal tiling word* $\mathbf{w}_T^{(u)} = \mathbf{w}^{(T)}$. For the individuals from $u = 2$ to $u = U - 1$ use *"Rule 1"* and *"Rule 3"* to increase by one the letters that belong to vertex of internal region of the aperture ($v_n \in A_{int}$): $w_l^{(u)} \mid_{u=2} = w_l^{(u)} \mid_{u=1} + 1$. Then exploit *"Rule 2"* and *"Rule 3"* to generate all the others individuals. If the individuals generated

$\widetilde{U}$ are less the number of individuals required U $\left(\widetilde{U} < U\right)$. The remaining words are generated using algorithm for *minimal tiling*. Otherwise if $\widetilde{U} > U$ select randomly U solutions from $\widetilde{U}$.

- *Step 2: Binary Coding*

  The strings associated with population individuals and height function have integer value. To encode integer into binary, the first step is to find the maximum value of *maximal tiling word* then compute the number of bits to encode that value: $n_{bit} = log_2 \left\lceil max \left\{ \mathbf{w}^{(T)} \right\} \right\rceil$; finally the number of bits that encode an individual is $N_{bit} = N_{int} \times log_2 \left\lceil max \left\{ \mathbf{w}^{(T)} \right\} \right\rceil$.

- *Step 3: Reproduction Cycle:*

  - Apply the *roulette-wheel* selection, crossover with probability p_c and mutation with probability p_m to geneate new individual (tiling word). Generally $p_c = 90\%$ and $p_m = 0.1\%$
  - Compute the new height function: $h_n^{(t)} = 3 \cdot w_n^{(t)} + h_n^{(1)},\ n = 1, ..., N_{int}$
  - Check the admissibility of the individual using the relation $\left| h_n^{(t)} - h_p^{(t)} \right| = \{1; 2\}$, if it is true the word (individual) is admissible, otherwise discrad the individual and generate a new one
  - Iterate the cycle until the population isn't complete

- *Step 4: Fitness Evaluation*

  - Determine the GA-Population corresponding to word set
  - Compute the fitness associated to each individual
  - Apply elitism operator to keep the best individual

- *Step 5: Convergence Check*

  - The optimization stop when the number of iterations (i) are equal to the number of iteration fixed (I): $i = I$
  - If the condition is true the convergence is reached and $\mathbf{c}^{opt} = \mathbf{c}_I^{opt}$, otherwise increase iteration index $i = i + 1$ and go to *Step 3 - Reproduction Cycle*

3.5.3 Integer-GA Optimization Tiling Method (OTM-IGA)

One problem of Binary-GA is due to the lenght of string when there are a lot of unknows. For example if the maximum letter of maximal tiling word is equal to ten, it need four bits to encode every letter, if word lenght is 37 the total lenght of binary string is $37 \times 3 = 111$. It is necessary to manage 111 binary value during the optimization process, so a lot of time computational effort is needed. One way to reduce the string lenght is to represent letter with integer rather than binary (Tab. 4). This solution is justified from the height function and tiling word value, because they have integer value.

The main advantages are:

- Avoid conversion of height function value from integer to binary,
- Integer coding permits to reduce computational effort, as result the simulation period descreases,
- GA operators have more probability to generate the admissible word due to string more shorter than binary case.

Word type	**Word value**	**Word lenght**	$\mathbf{N_{bit}}$ to coding
Binary	001001001001001010010010001001010011011010 001001010011100011010001001010011011010 00100101001001000100100100 1001	111 bit	3 bit
Integer	111112221123321123432112332112221111	37 integer value	/

Table 4: Integer String vs. Binary String. Group of 3 bits corresponds to one integer letter

The integer based method is similar to binary case [Sec.3.5.2]; also for the key-points: GA-Schemata and GA-Implicit Parallelism, the only difference is the coding type that permits to simplify the implementation, because is similar to binary case described in [Sec.3.5.2] but without *Step 2 - Binary Coding:*

- *Step 1: Population Initialization*
- *Step 2: Reproduction Cycle*

  there are differences between integer and binary case for crossover and mutation operator due to different coding type:

  - Integer Crossover: this operator cut two strings associated with two individuals and merge them to generate a best individual than the two before, with integer value we have more possibility to generate a feasible individual than binary case
  - Integer Mutation: this operator change a value to another in the string of individual, in a partially random way, this permits to add a new individual to the population, with integer value we have more possibility to generate a feasible individual than binary case
- *Step 3: Fitness Evaluation*
- *Step 4: Convergence Check*

More information on the topics of this document can be found in the following list of references.

References

[1] P. Rocca, M. Benedetti, M. Donelli, D. Franceschini, and A. Massa, "Evolutionary optimization as applied to inverse problems," *Inverse Problems - 25 th Year Special Issue of Inverse Problems, Invited Topical Review*, vol. 25, pp. 1-41, Dec. 2009

[2] P. Rocca, G. Oliveri, and A. Massa, "Differential Evolution as applied to electromagnetics," *IEEE Antennas Propag. Mag.*, vol. 53, no. 1, pp. 38-49, Feb. 2011

[3] P. Rocca, N. Anselmi, A. Polo, and A. Massa, "An irregular two-sizes square tiling method for the design of isophoric phased arrays," *IEEE Trans. Antennas Propag.*, vol. 68, no. 6, pp. 4437-4449, Jun. 2020

[4] P. Rocca, N. Anselmi, A. Polo, and A. Massa, "Modular design of hexagonal phased arrays through diamond tiles," *IEEE Trans. Antennas Propag.*, vol.68, no. 5, pp. 3598-3612, May 2020

[5] N. Anselmi, L. Poli, P. Rocca, and A. Massa, "Design of simplified array layouts for preliminary experimental testing and validation of large AESAs," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 6906-6920, Dec. 2018

[6] N. Anselmi, P. Rocca, M. Salucci, and A. Massa, "Contiguous phase-clustering in multibeam-on-receive scanning arrays," *IEEE Trans. Antennas Propag.*, vol. 66, no. 11, pp. 5879-5891, Nov. 2018

[7] G. Oliveri, G. Gottardi, F. Robol, A. Polo, L. Poli, M. Salucci, M. Chuan, C. Massagrande, P. Vinetti, M. Mattivi, R. Lombardi, and A. Massa, "Co-design of unconventional array architectures and antenna elements for 5G base station," *IEEE Trans. Antennas Propag.*, vol. 65, no. 12, pp. 6752-6767, Dec. 2017

[8] N. Anselmi, P. Rocca, M. Salucci, and A. Massa, "Irregular phased array tiling by means of analytic schemata-driven optimization," *IEEE Trans. Antennas Propag.*, vol. 65, no. 9, pp. 4495-4510, September 2017

[9] N. Anselmi, P. Rocca, M. Salucci, and A. Massa, "Optimization of excitation tolerances for robust beamforming in linear arrays," *IET Microwaves, Antennas & Propagation*, vol. 10, no. 2, pp. 208-214, 2016

[10] P. Rocca, R. J. Mailloux, and G. Toso, "GA-Based optimization of irregular sub-array layouts for wideband phased arrays desig," *IEEE Antennas and Wireless Propag. Lett.*, vol. 14, pp. 131-134, 2015

[11] P. Rocca, M. Donelli, G. Oliveri, F. Viani, and A. Massa, "Reconfigurable sum-difference pattern by means of parasitic elements for forward-looking monopulse radar," *IET Radar, Sonar & Navigation*, vol 7, no. 7, pp. 747-754, 2013

[12] P. Rocca, L. Manica, and A. Massa, "Ant colony based hybrid approach for optimal compromise sum-difference patterns synthesis," *Microwave Opt. Technol. Lett.*, vol. 52, no. 1, pp. 128-132, Jan. 2010

[13] P. Rocca, L. Manica, and A. Massa, "An improved excitation matching method based on an ant colony optimization for suboptimal-free clustering in sum-difference compromise synthesis," *IEEE Trans. Antennas Propag.*, vol. 57, no. 8, pp. 2297-2306, Aug. 2009

[14] M. Salucci, L. Poli, A. F. Morabito, and P. Rocca, "Adaptive nulling through subarray switching in planar antenna arrays," *Journal of Electromagnetic Waves and Applications*, vol. 30, no. 3, pp. 404-414, February 2016

[15] T. Moriyama, L. Poli, and P. Rocca, "Adaptive nulling in thinned planar arrays through genetic algorithms," *IEICE Electronics Express*, vol. 11, no. 21, pp. 1-9, Sep. 2014

[16] L. Poli, P. Rocca, M. Salucci, and A. Massa, "Reconfigurable thinning for the adaptive control of linear arrays," *IEEE Trans. Antennas Propag.*, vol. 61, no. 10, pp. 5068-5077, Oct. 2013

[17] P. Rocca, L. Poli, G. Oliveri, and A. Massa, "Adaptive nulling in time-varying scenarios through time-modulated linear arrays," *IEEE Antennas Wireless Propag. Lett.*, vol. 11, pp. 101-104, 2012

[18] M. Benedetti, G. Oliveri, P. Rocca, and A. Massa, "A fully-adaptive smart antenna prototype: ideal model and experimental validation in complex interference scenarios," *Progress in Electromagnetic Research*, PIER 96, pp. 173-191, 2009

[19] P. Rocca, L. Poli, A. Polo, and A. Massa, "Optimal excitation matching strategy for sub-arrayed phased linear arrays generating arbitrary shaped beams," *IEEE Trans. Antennas Propag.*, vol. 68, no. 6, pp. 4638-4647, Jun. 2020

[20] G. Oliveri, G. Gottardi and A. Massa, "A new meta-paradigm for the synthesis of antenna arrays for future wireless communications," *IEEE Trans. Antennas Propag.*, vol. 67, no. 6, pp. 3774-3788, Jun. 2019

[21] P. Rocca, M. H. Hannan, L. Poli, N. Anselmi, and A. Massa, "Optimal phase-matching strategy for beam scanning of sub-arrayed phased arrays," *IEEE Trans. Antennas and Propag.*, vol. 67, no. 2, pp. 951-959, Feb. 2019

[22] L. Poli, G. Oliveri, P. Rocca, M. Salucci, and A. Massa, "Long-Distance WPT Unconventional Arrays Synthesis," *Journal of Electromagnetic Waves and Applications*, vol. 31, no. 14, pp. 1399-1420, Jul. 2017

[23] G. Gottardi, L. Poli, P. Rocca, A. Montanari, A. Aprile, and A. Massa, "Optimal Monopulse Beamforming for Side-Looking Airborne Radars," *IEEE Antennas Wireless Propag. Lett.*, vol. 16, pp. 1221-1224, 2017

[24] G. Oliveri, M. Salucci, and A. Massa, "Synthesis of modular contiguously clustered linear arrays through a sparseness-regularized solver," *IEEE Trans. Antennas Propag.*, vol. 64, no. 10, pp. 4277-4287, Oct. 2016

[25] P. Rocca, G. Oliveri, R. J. Mailloux, and A. Massa, "Unconventional phased array architectures and design Methodologies - A review," *Proceedings of the IEEE = Special Issue on 'Phased Array Technologies', Invited Paper*, vol. 104, no. 3, pp. 544-560, March 2016

[26] P. Rocca, M. D'Urso, and L. Poli, "Advanced strategy for large antenna array design with subarray-only amplitude and phase contr," *IEEE Antennas and Wireless Propag. Lett.*, vol. 13, pp. 91-94, 2014

[27] L. Manica, P. Rocca, G. Oliveri, and A. Massa, "Synthesis of multi-beam sub-arrayed antennas through an excitation matching strategy," *IEEE Trans. Antennas Propag.*, vol. 59, no. 2, pp. 482-492, Feb. 2011

[28] G. Oliveri, "Multi-beam antenna arrays with common sub-array layouts," *IEEE Antennas Wireless Propag. Lett.*, vol. 9, pp. 1190-1193, 2010

[29] P. Rocca, R. Haupt, and A. Massa, "Sidelobe reduction through element phase control in sub-arrayed array antennas," *IEEE Antennas Wireless Propag. Lett.*, vol. 8, pp. 437-440, 2009

[30] P. Rocca, L. Manica, R. Azaro, and A. Massa, "A hybrid approach for the synthesis of sub-arrayed monopulse linear arrays," *IEEE Trans. Antennas Propag.*, vol. 57, no. 1, pp. 280-283, Jan. 2009

[31] 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 Trans. Antennas Propag.*, vol. 57, no. 3, pp. 652-664, Mar. 2009

[32] P. Rocca, L. Manica, A. Martini, and A. Massa, "Compromise sum-difference optimization through the iterative contiguous partition method," *IET Microwaves, Antennas & Propagation*, vol. 3, no. 2, pp. 348-361, 2009

[33] L. Manica, P. Rocca, and A. Massa, "An excitation matching procedure for sub-arrayed monopulse arrays with maximum directivity," *IET Radar, Sonar & Navigation*, vol. 3, no. 1, pp. 42-48, Feb. 2009

[34] L. Manica, P. Rocca, and A. Massa, "Design of subarrayed linear and planar array antennas with SLL control based on an excitation matching approach," *IEEE Trans. Antennas Propag.*, vol. 57, no. 6, pp. 1684-1691, Jun. 2009