
Design Isophoric Phased Arrays Using an Irregular Two-size Square Tiling Approach

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

2019/09/15

Contents

1	Introduction	3
2	Mathematical Formulation	6
3	Tiling Theory, Theorems and Methods	10
3.1	Tiling Problem	10
3.2	Tileability	11
3.3	Problem Cardinality	13
3.4	Exhaustive Tiling Methods	14
3.4.1	Binary Matrix-Based Tiling Method - $(m \times m, 2m \times 2m) - BMTM$	14
3.4.2	Binary Matrix-Based Tiling Method - $(m \times m, sm \times sm) - BMTM$	21
3.5	Optimization-based Tiling Methods	25
3.5.1	Optimization Algorithms	25

1 Introduction

Since their introduction in the early 1900s, phased array antennas have played a central role in a huge number of applications. They are currently used in terrestrial wireless and satellite communications, RADAR systems, automotive, remote-sensing, biomedical-imaging, radiotherapy, radio-astronomy and in many other fields. The widespread use of phased arrays is due to the significant advantages that this technology provides with respect to others more conventional radiating systems (single elements antennas). The advantages include higher radiation performance, fast beam scanning, reconfigurability and reliability. Notwithstanding all these interesting features, the deployment of phased array antennas is strongly limited by several factors such as: implementation costs, architectural complexity, low power efficiency and the overall system weight. This statement is supported by the fact that almost the total cost of a phased array is due to the feeding network - RF boards/cabling - (45% of the total cost) and to the transmit/receive modules - TRM - (45% of the total cost) used for the beam forming. In the light of these facts, unconventional solutions e.g. clustered, thinned and sparse array architectures have been investigated.

Thinned arrays are obtained starting from a fully populated array, with the elements paced on a periodic lattice (Fig. 1), through the so called *thinning* operation. This operation requires to connect a sub-set of elements to a matched load (or alternatively the elements can be removed) such that they do not directly contribute to the pattern. The resulting aperiodic arrangement is called a thinned array. This kind of architectures are used in applications where the beam-width is of primary importance and the high antenna gain is not a concern. Is this the case of antennas that has to operate against a jamming environment, like ground-based high-frequency RADARs, interferometer arrays for radio astronomy and satellite receiving antennas. The non-periodic elements spacing allows to reduce the number of radiating elements while providing a spatial taper - distribution of active elements - yielding lower side-lobes and a pattern close to that of the fully populated array. The effective thinning of large linear and planar arrays has been addressed using analytically-defined binary sequences called the difference and the almost difference sets (DSs and ADSs).

Similarly to thinned arrays, sparse array architectures implement an aperiodic elements distribution able to create a spatial taper. But, unlike thinned, clustered and periodic arrangements, the elements of sparse array can be arbitrary placed on the antenna aperture. We have already see in thinned arrays, that the main advantages of aperiodic elements distribution are the reduction of the side-lobe level (*SLL*) without the need of any amplitude tapering and, compared to a periodic architecture, the reduction of the number of radiating elements - and thus of TRM modules - without a major impact on the beam-width. The synthesis procedure for a sparse array is indeed devoted at the retrieval of the unknown elements location. Unfortunately, these problem unknowns appears at the exponential of the array factor equation and the overall task turn out to be highly nonlinear. Only in recent years, also thanks to the improvements in the computation capabilities, effective synthesis methods have been proposed. Among these techniques, it is worth mentioning compressive sensing (CS) based and deterministic methodologies.

Clustered/Subarrayed antenna arrays are obtained through a partition of a uniform array into smaller subarrays/clusters (Fig. 5 and 6) having different size, shape or orientation. The main purpose of clustered arrays is the reduction of the number of control points (e.g. amplifiers, phase shifters and time-delay units) by moving some functionalities/features of element level at the subarray level. Over the last few years, subarrayed architectures have become popular since

they enables simplifications of the antenna design while preserving high aperture efficiency and good radiation properties. Furthermore, the reduced number of control units and the intrinsic *modularity* induced by the presence of subarrays makes these architectures an interesting low-cost solutions for many applications. For example, in the synthesis of mono-pulse radar tracking radar for surveillance application, several clustering techniques have proven their effectiveness.

The concept of *modularity* is recently gaining more and more attention due to the dramatic cost reduction that a simplified architecture can provide. In this direction, constrained-shape(s) physically contiguous (i.e. modular) subarrays are obtained partitioning the array aperture using elementary module, or *tiles*, of arbitrary fixed shape. Typically, elementary tiles are cluster of two or more contiguous radiating elements arranged in predefined geometry. As a matter of fact, the need to reduce costs and simplify hardware implementation (assembly, maintenance, etc..) is leading to a growing development of *tiled* arrays. Several sub-optimal covering methods using polyomino tiles of different shapes (e.g. L-shape octomino) have been successfully implemented. Whereas, in an optimal covering approach using domino tiles has been presented.

The aim of this thesis is the development of an innovative strategy for the design of modular antenna arrays, capable of compensate for the aforementioned drawbacks of the phased array technology as well as of satisfying the desired performance requirements. The proposed modular strategy make use of constrained-shapes physically contiguous non-overlapped subarrays to cluster the antenna aperture. For the specific case square tiles with two different edge size are used (See Fig. 6 for an illustrative example). Starting from the analysis of covering theorems derived in field of mathematics for square tiles of size 1×1 and 2×2 , an enumerative approach (Exhaustive Tiling Method, ETM) able to retrieve the optimal tiling providing the total aperture coverage and the best radiation performance with square tiles of generic size $m \times m$ and $n \times n$, has been developed to deal with the synthesis of low/medium-size rectangular arrays. Further, since the number of admissible clustering rapidly grows with the array size, an Optimization-based Tiling Method (OTM) benefiting from a “*smart*” coding strategy has been introduced for the design of large arrays using square tiles of size $m \times m$ and $sm \times sm$. As done for the ETM also the OTM take advantages from the analysis of mathematical covering theory. Needless to say that the resulting modular/tiled architecture can profit from a large-scale production to provide a low costs yet simple solution for a wide range of applications. In order to illustrate the applicability of the proposed strategy particular attention has been given to the satellite communications application. In such a framework, in order to provide downlink and uplink coverage, communications satellites require electrically large antenna apertures able to provide high gain multiple overlapped spot beams. Typically these apertures are implemented by reflectors, even though phased arrays would be the best solution from a theoretic point of view. In fact, despite the advantages, phased arrays have been rarely used for SATCOM applications due to their poor power efficiency, high costs and deployment complexity.

In this scenario, the proposed approach is able to generate several solutions characterized by and low costs, low complexity, high power efficiency, high aperture efficiency, good radiation performance and a possible high thermal efficiency (thanks to the regular placement of the radiating elements).

The thesis is structured as follows: In section 2 are presented the basics of antenna arrays analysis. The general clustered arrays design problem with both non-isophoric and isophoric excitations has been stated, as well. Section 3 introduces the

tiling theory, the theorems and the developed tiling methods. Section XX contains the results of the numerical analyses, and has been divided into 2 subsections. In the first subsection are presented the results of a comparison with the state of the art of tiled-arrays, while in the second those concerning the state of the art in the field of satellite communications (sparse arrays) including an analysis with real elements and mutual coupling. Finally, the conclusions drawn from the results are reported.

ELEDIA Research Center

2 Mathematical Formulation

Starting from the general definition the array factor of a uniform planar array of $J = M \times N$ elements placed on the xy -plane is defined as:

$$AF(\theta, \phi) = \sum_{m=1}^M \sum_{n=1}^N I_{mn} e^{jk(x_m \sin\theta \cos\phi + y_n \sin\theta \sin\phi)} \quad (1)$$

where $I_{mn} = \alpha_{mn} e^{j\beta_{mn}}$ is the complex excitations (amplitude α_{mn} and phase β_{mn}) of the (m, n) th element ($m = 1, \dots, M; n = 1, \dots, N$), λ is the working wavelength, (x_m, y_n) are the coordinates of the (m, n) th array element on the grid, and (θ, ϕ) are the polar variables used to identify the far-field radial direction.

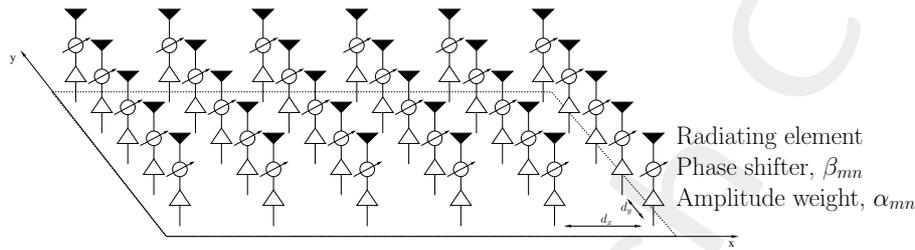


Figure 1: Sketch of the architecture of a regular phased array.

Let us consider a uniform planar array of $J = M \times N$ elements placed on a regular grid (Fig. 1). Clustered arrays are obtained by grouping the radiating elements into Q ($Q \leq M \times N$) subarrays/clusters having different size, shape or orientation. Similar considerations could be done also for linear arrays. The core idea of clustered arrays is to reduce the number of control points of the beam-forming network (e.g. amplifiers, phase shifters and time-delay units) by moving some functionalities/features of the element level at the subarray level. An example of transmit receive modules is reported in Fig. 2.

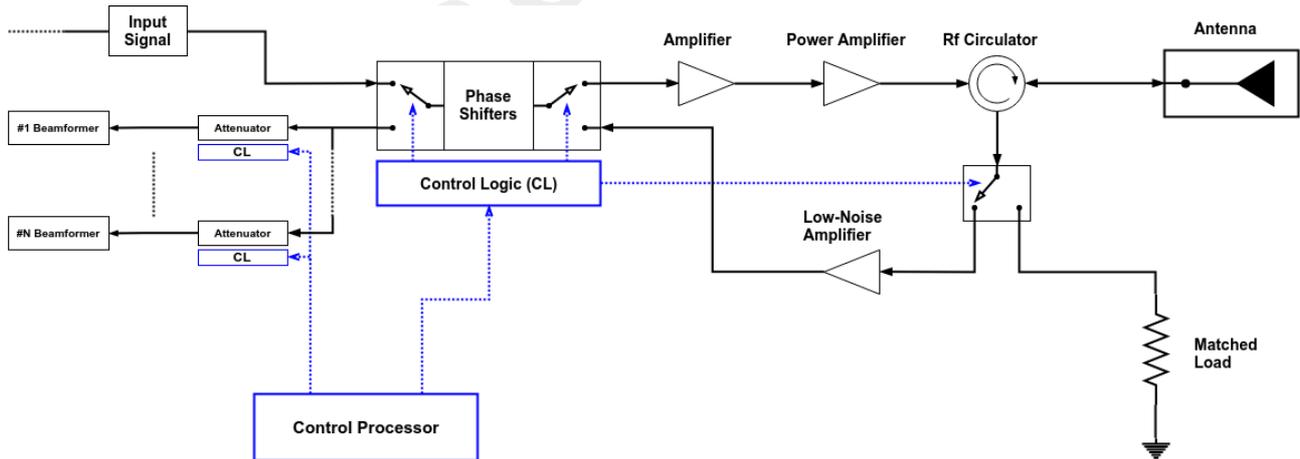


Figure 2: Simplified Transmit/Receive Module architecture;

Generally two main schemes are implemented, namely partial-subarrayed or fully-subarrayed. Partial-subarrayed schemes allow to move some element level control unit (e.g. amplifiers) at the subarray level while leaving other units (e.g. Phase shifters) at the element level (Fig. 3 (a)). While, in fully-subarrayed architectures all the control units are placed at the

subarray level, such that a single TRM modules is used to feed all the elements of the relative subarray (Fig. 3 (b)). In the following we will focus only on the fully-subarrayed scheme.

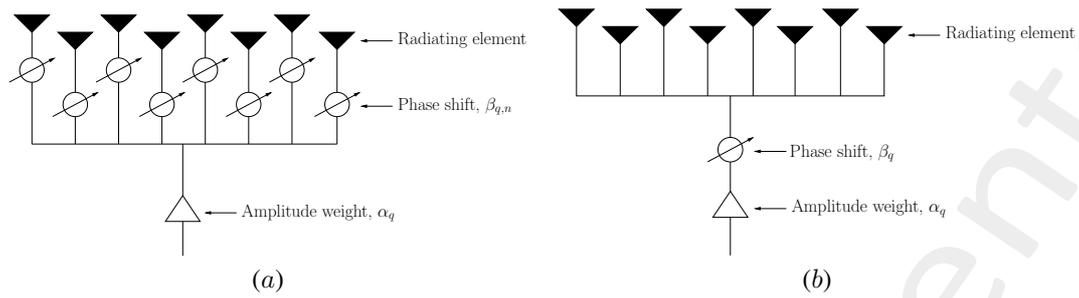


Figure 3: Partial (a) and fully (b) subarrayed scheme.

Concerning the shape of the clusters/subarrays it is possible to define two main classes of clustering algorithms: the first is based on unconstrained subarray shapes while the second based on constrained single-shape (Fig. 5) or multi-shape (Fig. 6). The clustering algorithms belonging to the unconstrained class do not impose any constraint on the clusters/subarrays shapes. In fact, the clusters could assume any kind of geometry.

Subarrays with unconstrained shape are not guarantee to be physically contiguous, thus possibly yielding a complex feeding network. Instead, physically contiguous clusters are of particular interest for modern practical application since they ensure a greater system modularity with respect to traditional unconstrained partition. Constrained-shape(s) physically contiguous (modular) subarrays are obtained partitioning the array aperture using elementary module, or *tiles*, of arbitrary fixed geometry. Typically the elementary tiles are subarrays of two or more contiguous radiating elements arranged in predefined geometry.

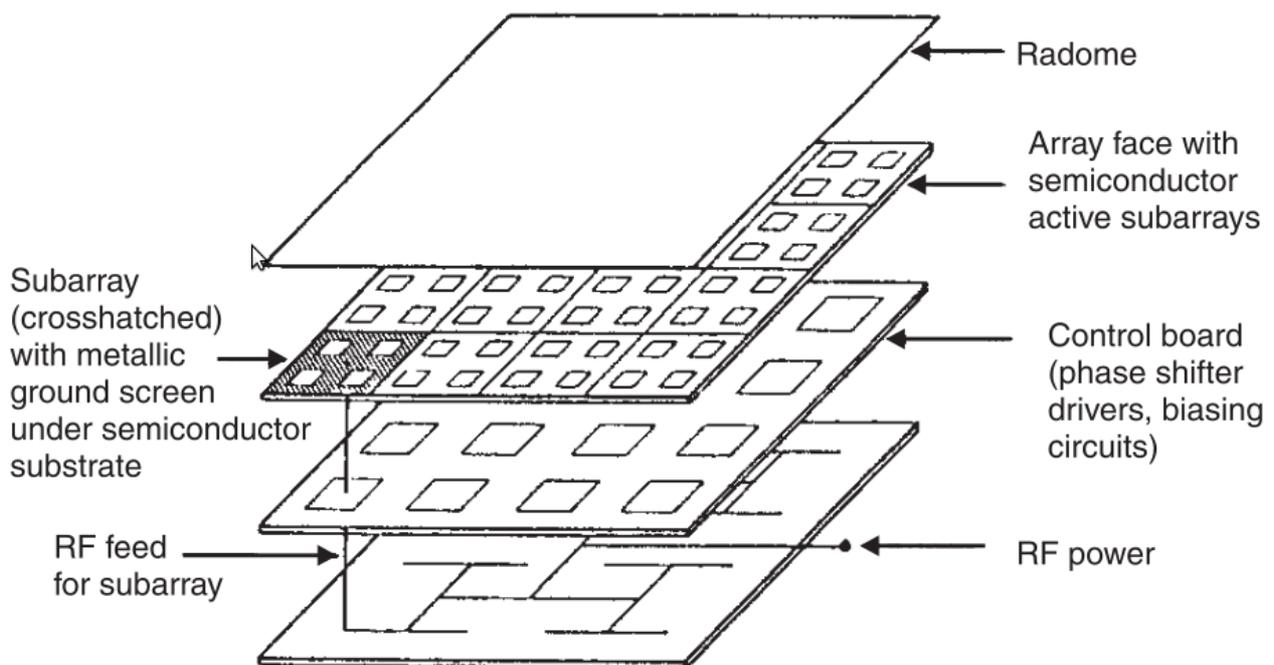


Figure 4: Tile Example;

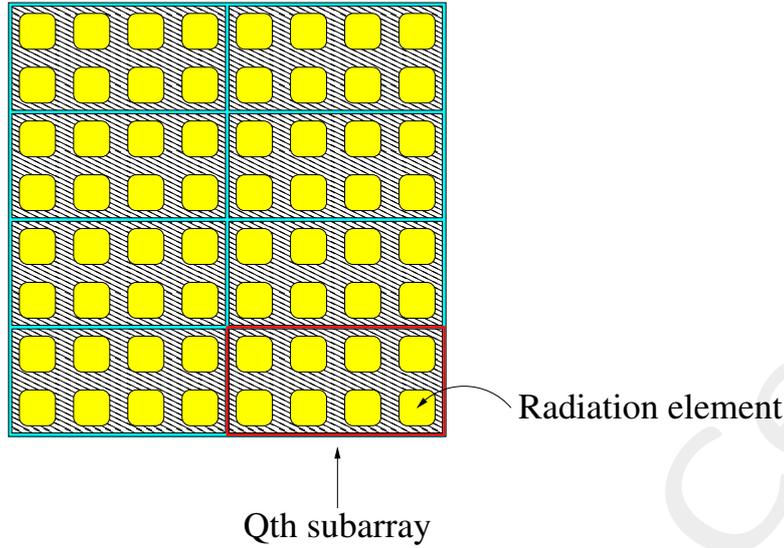


Figure 5: Subarrayed planar array (front view).

The general definition for the elements complex excitations are expressed in function of the subarray membership:

$$I_{mn} = I_{c_{mn}} = \alpha_{c_{mn}} e^{j\beta_{c_{mn}}}, \quad m = 1, \dots, M; \quad n = 1; \dots, N; \quad c_{mn} \in [1, Q] \quad (2)$$

where the integer index c_{mn} define the membership of each (m, n) th element of the array to one of the Q subarray, while $\alpha_{c_{mn}}$ and $\beta_{c_{mn}}$ are the q -th subarray amplitude and phase with $c_{mn} = q, m = 1, \dots, M; n = 1; \dots, N; q \in Q$.

The sub-array amplitude $\alpha_{c_{mn}}$ and phase $\beta_{c_{mn}}$ are usually defined by a non-isophoric excitation matching strategy. The *means/centroid* of the reference elements excitations of the q -th subarray is adopted for the subarray amplitude coefficient:

$$\alpha_{c_{mn}} = \frac{1}{N_q} \sum_{\alpha_{mn} \in q} \alpha_{mn}, \quad q \in Q \quad (3)$$

While the phase coefficients are computed according to:

$$\beta_{c_{mn}} = -kr_{c_{mn}} (\sin\theta\cos\phi + \sin\theta\sin\phi) \quad (4)$$

Where

$$r_{c_{mn}} \triangleq (x_{c_{mn}}, y_{c_{mn}}) = \frac{1}{N_q} \sum_{r_{mn} \triangleq (x_m, y_n) \in q} r_{mn}, \quad q \in Q$$

is the geometrical centroid of the q -th subarray, N_q is the number of elements in q , α_{mn} are the reference amplitude coefficients and $r_{mn} \triangleq (x_m, y_n)$ are the element coordinates on the lattice.

Excitation matching approaches are aimed to determine the subarray configuration and the relative weights that best approximate a reference/optimal pattern.

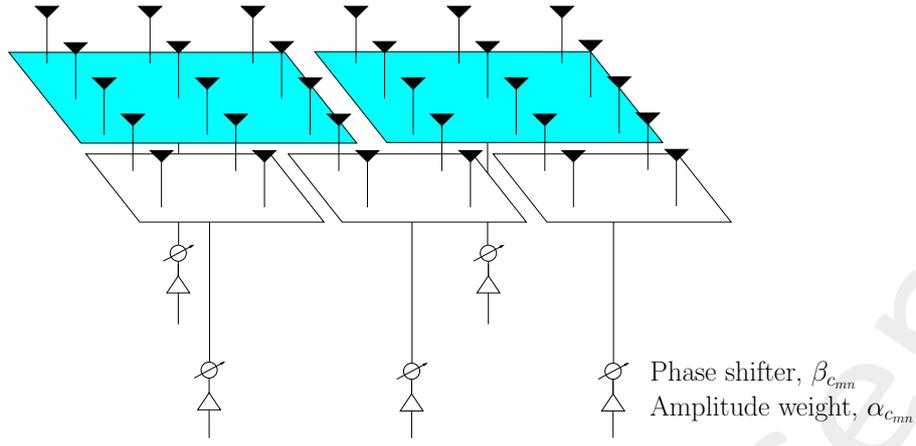


Figure 6: Sketch of the clustered architecture of a phased array.

In many modern applications (e.g. spacecraft antennas) a high power efficiency is required. This requirement is generally satisfied when the HPAs feeding the radiating elements (or the subarrays) are allowed to work at their maximum efficiency, i.e. providing equal-amplitude (isophoric) excitations. Hence considering a clustered array and assuming a single HPA to feed a subarray of N_q elements, the resulting amplitude weight of each element in the q th subarray is:

$$\alpha_{c_{mn}} = \sqrt{\frac{1}{N_q}} \quad (5)$$

The phase coefficients $\beta_{c_{mn}}$ are still computed according to (4).

Both the complex non-isophoric (excitation matching) and isophoric excitations defined according to (3),(5) and (3) will be considered in the results section.

Problem Statement

The design of planar clustered phased arrays require the definition of the DoFs of the system that are the clustering configuration $\underline{C} = \{c_{1,1}, c_{1,2}, \dots, c_{M,N}\}$, $c_{mn} \in [1, Q]$ and the complex excitation weights $I_{c_{mn}} = \alpha_{c_{mn}} e^{j\beta_{c_{mn}}}$, $m = 1, \dots, M$; $n = 1, \dots, N$; $c_{mn} \in [1, Q]$ for each radiating element of the antenna. Accordingly the following minimization problem must be solved:

$$(\underline{C}^{opt}, \underline{\alpha}^{opt}, \underline{\beta}^{opt}) = \min_{\{\underline{C}, \underline{\alpha}, \underline{\beta}\}} \left\{ \Phi \left[|AF(\theta_s, \phi_s, \underline{C}, \underline{\alpha}, \underline{\beta})|^2 \right] \right\} \quad (6)$$

Φ being the a cost function accounting for the desired objectives. Typically *SLL* (Fig. 7 (a)) and Mask Matching (Fig. 7 (b)) are considered, but nothing prevents designers from considering other objectives (e.g. directivity, HPBW, number of TRMs, etc ..).

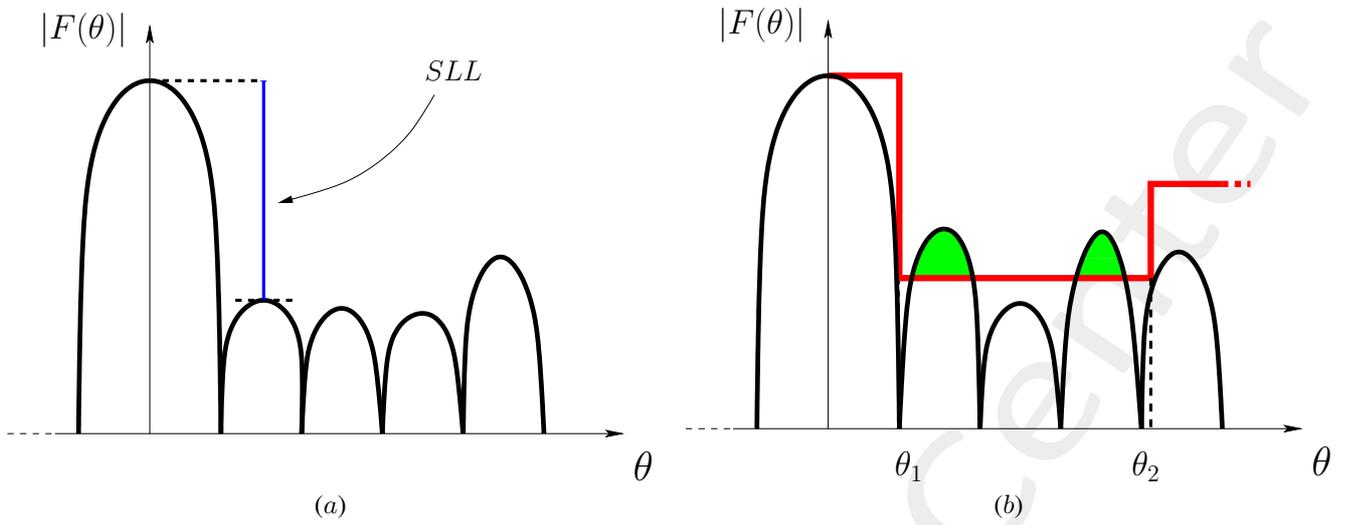


Figure 7: SLL (a) and Mask Matching (b) computation illustration.

3 Tiling Theory, Theorems and Methods

Let us consider the tilings of a closed simply connected region R of \mathbb{Z}^2 by two squares of different sizes m and n , where $m, n \in \mathbb{N}$.

3.1 Tiling Problem

Given a pair of squares $\{S_m, S_n\}$ with sides of fixed integer dimensions m and n , where $m, n \in \mathbb{N}$, any translated copy of the square S_m is commonly defined as an m -tile (respectively n -tile if the square of side n is considered). The set \underline{v} of non-overlapping tiles that completely covers (with no gaps) a figure R defines what it is called a *tiling* of R , see Fig. 8. Therefore, the tiling problem aims to find the set $\underline{\Upsilon} = \{\underline{v}^{(\gamma)}, \gamma, \dots, \Gamma\}$ of all the configuration/tilings that satisfy the above definition of tiling (Γ denotes the cardinality of the problem, i.e. the number of allowed tilings).

At this point two questions arise:

1. Given S_m and S_n , is it possible to determine whether a generic region R is tileable (i.e. can be completely covered with no gaps) or non-tileable?
2. If the answer to 1) is positive, is it possible to a priori know the cardinality Γ ?

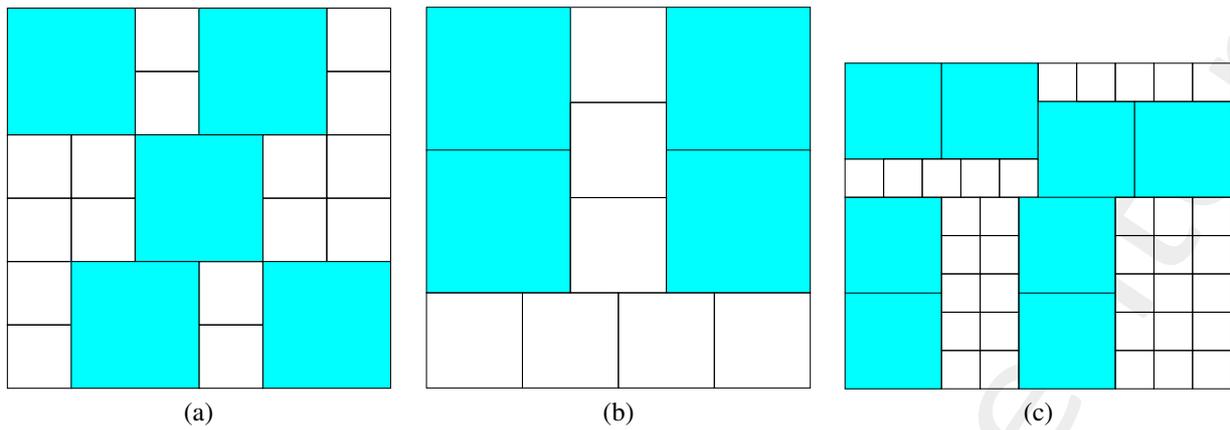


Figure 8: Examples of Tilings - $R = 6 \times 6$ with $(1 \times 1, 2 \times 2)$ -tiling (a), $R = 8 \times 8$ with $(2 \times 2, 3 \times 3)$ -tiling (b) and $R = 15 \times 20$ with $(2 \times 2, 5 \times 5)$ -tiling (c).

3.2 Tileability

Theorem 1 (Dehn 1903) An $R = A \times B$ can be tiled by squares (not necessarily equal) if and only if the ratio A/B is rational.

Theorem 2 Let us consider a rectangular region $R = A \times B$ and two square tiles $S_m = m \times m$ and $S_n = n \times n$. R is tilable by S_m and S_n if and only if the following conditions are satisfied:

- (i) either: A and B are divisible by m ;
- (ii) or: A and B are divisible by n ;
- (iii) or: A is an integer multiple of $m \times n$ and B is a positive linear combination of m and n ;
- (iv) or: B is an integer multiple of $m \times n$ and A is a positive linear combination of m and n .

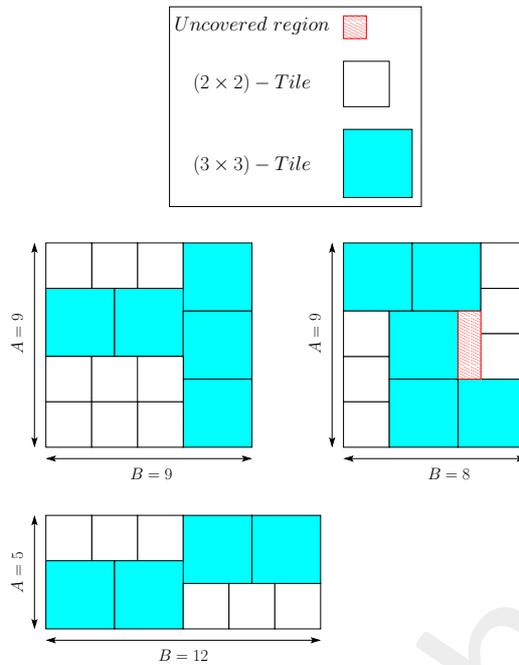


Figure 9: $(2 \times 2, 3 \times 3)$ – tiling Tileable vs. Untileable region.

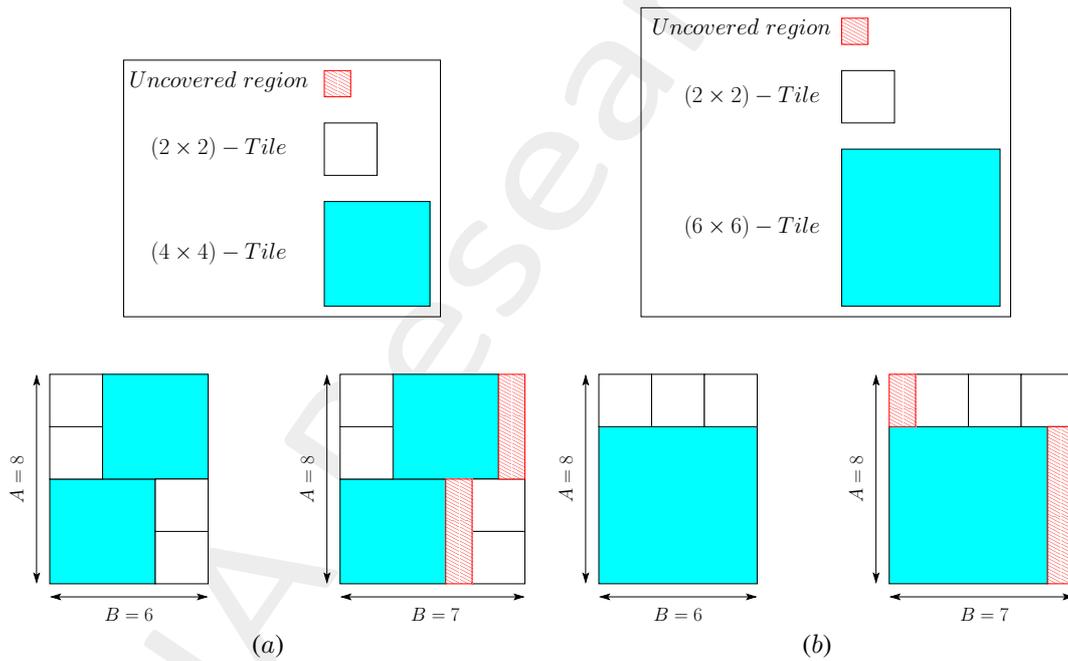


Figure 10: $(2 \times 2, s \cdot 2 \times s \cdot 2)$ – tiling Tileable vs. Untileable region, $s = 2$ (a) and $s = 3$ (b).

PROOF

(i) If A and B are divisible by m there exists always a tiling of R using only the tile S_m that can be obtained as follows: we first place a row of $\frac{B}{m}$ S_m tiles in order to completely cover the bottom edge of the uncovered part of the rectangle and repeat for $\frac{A}{m} - 1$ times until R is completely tiled.

(ii) If A and B are divisible by n there exists always a tiling of R using only the tile S_n that can be obtained as follows: we first place a row of $\frac{B}{n}$ S_n tiles in order to completely cover the bottom edge of the uncovered part of the rectangle and repeat for $\frac{A}{n} - 1$ times until R is completely tiled.

(iii) if A is an integer multiple of $m \times n$ and B is a positive linear combination of m and n there exists always a tiling of R using the tiles S_n and S_m that can be obtained as follows: being A an integer multiple of m we can place a column of S_m tiles, completely covering the dimension of R parallel to the edge having A length. Let us refer to this group of placed tiles as an m -column. Similarly, being A an integer multiple also of n we can place a column of S_n tiles, completely covering the dimension of R parallel to the edge having A length. Let us refer to this group of placed tiles as an n -column. The tiling of R can be obtained as a linear combination of m -columns and n -columns completely covering also the dimension of R parallel to the edge having B length.

(iv) for symmetry, can be obtained as in (iii) exchanging A with B .

Finally in order to complete the proof, let us prove the following statement directly derived from the theorem above:

If A is not divisible by n , and B is not divisible by m , (or A is not divisible by m , and B is not divisible by n), then R is not tilable by S_m and S_n .

Let us color the cells of the region R with black and white colors, according to the pattern shown in Fig. 11 in which the pattern composed by n black columns and $m - n$ white columns is periodically repeated from left to right. Accordingly every tile placed within R will always cover a number of black cells which is a multiple of n , independently by their position in R . If B is not divisible by m and A is not divisible by n , the total number of black cells turns out to be non divisible by n . Consequently the total number of black cells is not a multiple of n and R cannot be tiled.

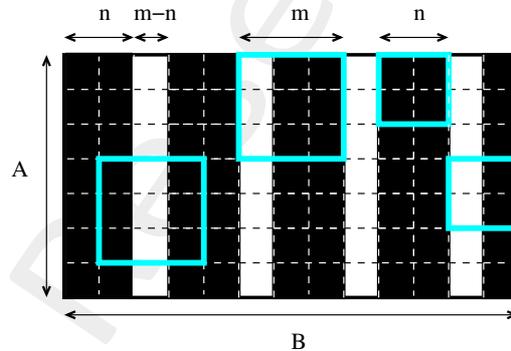


Figure 11: Proof of the tilability considering $A = 7$, $B = 13$, $m = 3$ and $n = 2$.

3.3 Problem Cardinality

The problem of *counting* the number of tiling of a rectangle has been discussed. The two problems are equal if the task of “*counting kings*”⁽¹⁾ is performed on a 2D board. Unfortunately, only in the specific case where the tiles have dimensions $m = 1$ and $n = 2$ it is possible to a priori calculate Γ .

Theorem 3 - $n = 2$:

Given $R = A \times B$ the cardinality Γ for the $(1 \times 1, 2 \times 2)$ -tiling of R is defined:

$$\Gamma = \mathbf{1}^T \underline{\underline{G}}^{A-2} \mathbf{1} \quad (7)$$

(1)

Where $\mathbf{1}$ is a column vector of ones and G is the adjacency matrix (The definition of matrix G is detailed in 3.4.1). Since G is symmetric the above equation can be rewritten as:

$$\Gamma = \mathbf{1}^T \underline{G}^{\alpha+\beta} \mathbf{1} = (\underline{G}^{\alpha} \mathbf{1})^T \cdot (\underline{G}^{\beta} \mathbf{1}) \quad (8)$$

Furthermore, the number of tilings Γ for a region $R = A \times B$ is exactly the same as for a region $R = B \times A$. If the region R is square ($A = B$) the number of tilings is provided by the sequence A063443 (Table I).

$R = A \times B$	Number of Tilings
(3×3)	5
(4×4)	35
(5×5)	314
(6×6)	6427
(7×7)	20.28×10^4
(8×8)	12.72×10^6

Table I: Number of Tilings of a square region (A063443), $R = A \times B$.

It is worth mentioning that (7) can be straightforwardly generalized for the $(m \times m, 2m \times 2m)$ -tiling of a region $R = (Am) \times (Bm)$.

3.4 Exhaustive Tiling Methods

3.4.1 Binary Matrix-Based Tiling Method - $(m \times m, 2m \times 2m)$ - BMTM

Let us consider a rectangular region R of \mathbb{Z}^2 with size $A \times B$ and a set of square tiles $m \times m$ and $2m \times 2m$ where $A, B, m \in \mathbb{N}$. Proven that such a region is coverable/tileable with such tiles (Theorem 1), we are interested in exploring the whole set of allowed configurations/tilings $\underline{\Upsilon}$. In order to do that a suitable encoding for each tiling \underline{v} of R is needed.

Let us start considering the case of tiling using squares of size $m = 1$ and $n = 2m = 2$, the generic $(m \times m, 2m \times 2m)$ -tiling case will be introduced as an extension of the current one.

Matrix Representation

The authors state that it is possible to introduce a binary matrix \underline{M} to represent one admissible tiling of R . The matrix \underline{M} is constructed by placing ones in the matrix cell corresponding to the index associated with the lower left corner of the n -tile (i.e. the "big" tile) on the square lattice of R , and zeros elsewhere. In accordance with the definition of \underline{M} , the first row and last column are always zero vectors; therefore, it is sufficient to consider a $k \times q$ matrix instead of an $A \times B$ matrix. For the case of interest since $n = 2$ and $k = A - (n - 1) = A - 1$ and $q = B - (n - 1) = B - 1$. See Fig. 12 for an illustrative example with $R = 5 \times 4$.

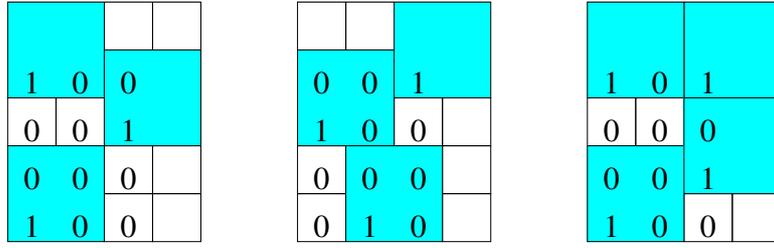


Figure 12: Binary Matrix Representation, $R = 5 \times 4$ - Three $(1 \times 1, 2 \times 2)$ -tilings.

The associated binary matrix are:

$$\underline{M}_{4,3}^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \underline{M}_{4,3}^{(2)} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \underline{M}_{4,3}^{(3)} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Binary Words

Let us now consider a generic row of a matrix $\underline{M}_{k,q}$ defined as above. A row of such matrix is a binary word of q letters. From to the definition of tiling (Par. 3.1) the overlap between tiles is not allowed. Therefore, in the exhaustive generation of candidate binary words (rows) the following constraint must be considered:

- **Inter-Letters Dependency:** Each letter in a word is dependent from its neighbours, i.e. in binary word it is not admissible to have two consecutive ones.

The resulting set of (all) the allowed words $\underline{T} = \{\underline{t}^{(i)}, i = 1, \dots, |\underline{T}|\}$ with

$$\underline{t}^{(ith)} = \{t_j \in \{0, 1\}, j = 1, \dots, q : t_j t_{j+1} \neq 11, t = t_1, \dots, t_q\} \quad (9)$$

is generated using a binary counter combined with the inter-letter dependency check defined above. The cardinality of the set \underline{T} :

$$|\underline{T}| = F_{q+2} \quad (10)$$

where F_q is the q th Fibonacci number (see A000045).

Graph Representation

Once defined \underline{T} according to inter-letter dependency criterion, each tiling \underline{v} of R can be represented through the matrix \underline{M} . In fact, \underline{M} is generated using the words $\underline{t} \in \underline{T}$. This concept has been discussed as the generalization/extension of the 1D “strip” case to the 2D “board” case. In order to do that it is necessary to define a criterion for the words “aggregation” (i.e. define how to generate \underline{M} using $\underline{t} \in \underline{T}$) such that the definition of tiling (sec. 3.1) is respected. For this purpose the following constraint has been imposed:

- **Inter-Words Dependency:** Each word (row) in $\underline{M}_{k,q}$ is dependent from its neighboring words. The row $\underline{r}_x = \{r_{x,1}, r_{x,2}, \dots, r_{x,q}\}$, x is the row index, can be placed on top of $\underline{r}_{x+1} = \{r_{x+1,1}, r_{x+1,2}, \dots, r_{x+1,q}\}$ iff $r_{x+1,j} = 0$ for $j \in \{i-1, i, i+1\} \cap \{1, \dots, q\}$ for all i such that $r_{x,i} = 1$ and $x \in \{1, \dots, k\}$.

The admissible words $\underline{t} \in \underline{T}$ are now represented as nodes in a finite graph Λ , where there is an edge between two nodes, $\underline{t}^{(g)}$ and $\underline{t}^{(h)}$ if and only if $\underline{t}^{(g)}$ can be placed next to $\underline{t}^{(h)}$ such that there are no inter-words dependencies and $g, h \in \{1, \dots, k\}$.

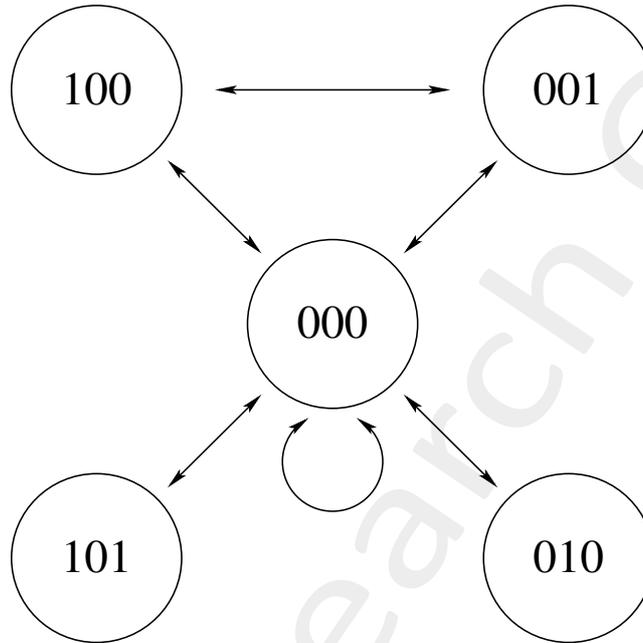


Figure 13: Graph Representation of the Words: Λ , $R = 5 \times 4$ - $(1 \times 1, 2 \times 2)$ -Tiling.

The resulting graph Λ encode all the possible combination of words, which means that it encode all the information needed to explore all the tilings $\underline{v} \in \underline{Y}$ of R . Straightforwardly any path of length k construct on skeleton of Λ produce a sequence of binary words (or matrix $\underline{M}_{k,q}$) representing a tiling \underline{v} for R .

In the example in Fig. 12 the paths are respectively:

$$\underline{M}_{4,3}^{(1)} = \begin{pmatrix} 100 \\ \downarrow \\ 001 \\ \downarrow \\ 000 \\ \downarrow \\ 100 \end{pmatrix}, \quad \underline{M}_{4,3}^{(2)} = \begin{pmatrix} 100 \\ \downarrow \\ 001 \\ \downarrow \\ 000 \\ \downarrow \\ 010 \end{pmatrix}, \quad \underline{M}_{4,3}^{(3)} = \begin{pmatrix} 101 \\ \downarrow \\ 000 \\ \downarrow \\ 001 \\ \downarrow \\ 100 \end{pmatrix}$$

Adjacency Matrix

The structure of a finite graph is usually represent by its adjacency matrix \underline{G} . The adjacency matrix is a binary symmetric matrix. Each entry of such matrix are $g_{x,y} = 1$ ($g_{y,x} = 1$) iff there is an edge between node x and node y , and is zero

otherwise.

In the case of interest the adjacency matrix $\underline{\underline{G}}$ encode the structure of the graph Λ whose nodes are $\underline{t} \in \underline{\underline{T}}$. Hence, the number of rows (and columns) of $\underline{\underline{G}}$ is given by the cardinality of $\underline{\underline{T}}$:

$$Rows(\underline{\underline{G}}) = |\underline{\underline{T}}| = F_{q+2}$$

Where $Rows(\cdot)$ indicate the number of rows of $\underline{\underline{G}}$.

It is possible to give a recursive definition of the matrix $\underline{\underline{G}}$.

$$\underline{\underline{G}}_q = \begin{pmatrix} \underline{\underline{G}}_{q-1} & \underline{\underline{G}}_{q-2} \\ \underline{\underline{G}}_{q-2} & 0 \end{pmatrix}, \quad \underline{\underline{G}}_0 = (1), \quad \underline{\underline{G}}_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

where the missing entries in $\underline{\underline{G}}_q$ is filled with zeros.

The number of edges in the graph Λ is proportional to the number of ones in the adjacency matrix. More precisely it is the number of ones in the upper (or lower) triangular adjacency matrix.

$$Ones(\underline{\underline{G}}_q) = \frac{4}{3}2^q - \frac{1}{3}(-1)^q$$

Where $Ones(\cdot)$ indicates the number of ones in $\underline{\underline{G}}_q$. The above equation reflects the Jacobsthal sequence (A001045).

The following figure reports two examples of adjacency matrix, $\underline{\underline{G}}_5$ (Fig. 14 (a)) and $\underline{\underline{G}}_6$ (Fig. 14 (b)). The white cells refers to the "1" (the black for the "0") of the associated matrix. These images provide a visual illustration of the recursive nature of $\underline{\underline{G}}$.

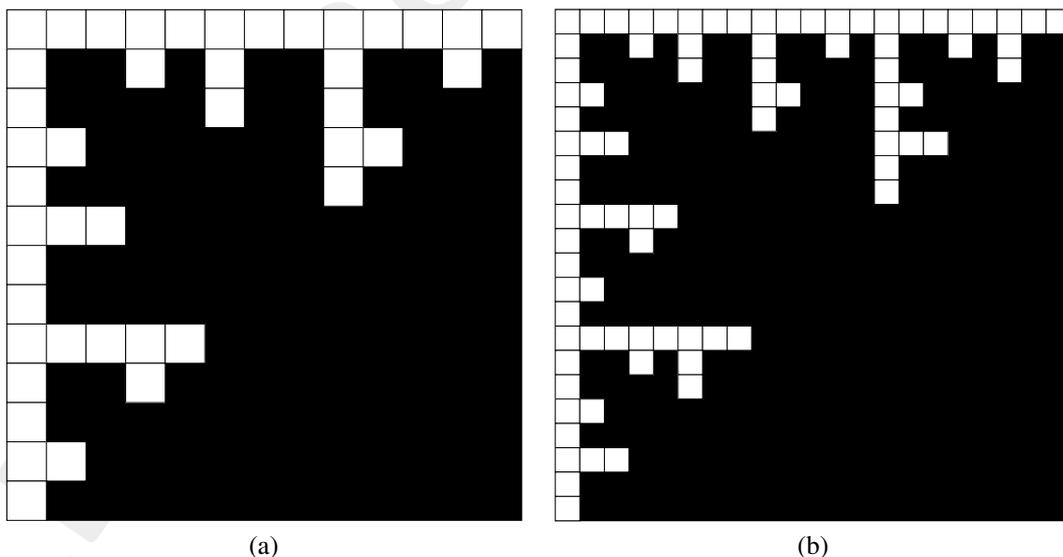


Figure 14: Adjacency Matrix - $\underline{\underline{G}}_5$ ($R = A \times 6$) (a) and $\underline{\underline{G}}_6$ ($R = A \times 7$) (b).

The structure of the graph Λ (and also $\underline{\underline{G}}_q$) does not change if the vertical side of the rectangle (A) increases or decreases. Indeed, the Λ (and also $\underline{\underline{G}}_q$) depends only on the horizontal side of the rectangle (B). However, increasing the vertical

side will increase the number of possible tilings Γ according to (7).

The General Case: $(m \times m, 2m \times 2m)$ -Tiling

The binary matrix representation can be extended to the case of a generic tiling \underline{v} of a rectangular region $R = (Am) \times (Bm)$ iff it is performed by means of square tiles of size m and $n = 2m$ with $m \in \mathbb{N}$. Thus, the $(m \times m, 2m \times 2m)$ -tiling problem is addressed through a recast of the $(1 \times 1, 2 \times 2)$ -tiling on proportionally scaled the region $R' = A \times B$.

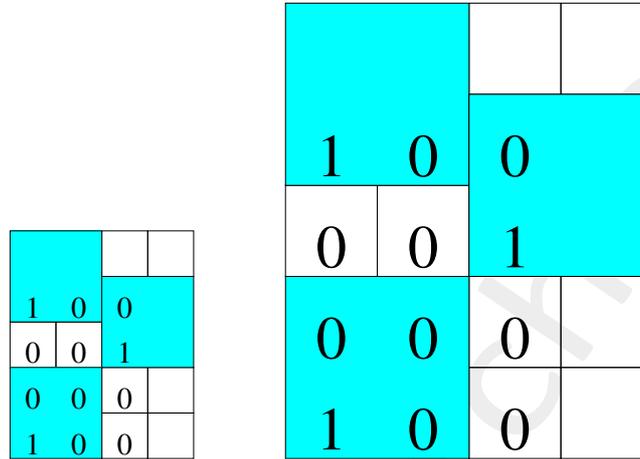


Figure 15: Binary Matrix Representation - $(1 \times 1, 2 \times 2)$ -Tiling $R = 5 \times 4$ and $(2 \times 2, 4 \times 4)$ -Tiling $R = 10 \times 8$.

Resume

Here are reported the steps that must be followed in order to exhaustively retrieve all the $(1 \times 1, 2 \times 2)$ -tilings $\underline{\Upsilon}$ of a rectangular region $R = A \times B$.

- Step 0: Define k and q as, $k = A - (2 - 1)$ and $q = B - (2 - 1)$;
- Step 1: Generate $\underline{T} = \{\underline{t}^{(i)}, i = 1, \dots, |\underline{T}|\}$ according to the **Inter-Letter** dependency;
- Step 2: Construct the graph $\Lambda(\underline{G})$ according to the **Inter-Words** dependency;
- Step 4: For each node of Λ (i.e. $\underline{t} \in \underline{T}$) find all the path on the skeleton of Λ having a length of k . In agreement with the theory, we know that each path is a tiling \underline{v} of R . Hence, the exploration of the whole set of these paths provides the exhaustive generation of $\underline{\Upsilon}$.

The pseudo-code for the function used to explore the whole set of fixed length path on Λ is reported below along with some illustrative pictures.

Algorithm 1 Pseudo-code of the iterative backtracking algorithm

```

depth ← (k - 1) ▷ Initialize depth
for  $\underline{t} \in \underline{T} = \{\underline{t}^{(i)}, i = 1, \dots, |\underline{T}|\}$  do ▷ Iterate on the nodes of  $\Lambda$ 
    FindFixedLengthPath ( $\underline{t}^{(i)}, \underline{G}, \underline{H}, \underline{S}, \text{depth}$ )
end

void FindFixedLengthPath ( $\underline{t}^{(i)}, \underline{G}, \underline{H}, \underline{S}, \text{depth}$ )
     $\underline{S} \leftarrow i$  ▷ Store the node id  $i$  to the vector  $\underline{S}$ 
    for  $j$  in  $G(i, j) == 1$  do ▷ Iterate on the neighboring nodes
        if (depth == 0) ▷ Path found!
             $\underline{H} \leftarrow \underline{S}$  ▷ Store the path  $\underline{S}$  in the matrix  $\underline{H}$ 
        if (depth - 1 < 0) ▷ Check: if true backtrack to the previous node
            return or exit if there are no other path to explore
        else
            FindFixedLengthPath ( $\underline{t}^{(j)}, \underline{G}, \underline{H}, \underline{S}, \text{depth} - 1$ ) ▷ Recursion
    end

```

At the end of the process the matrix $\underline{H} = \{\underline{S}^{(i)}, i = 1, \dots, \Gamma\}$ contains the whole set of paths. Then, since each path is represented by a vector $\underline{S} = \{s^{(j)}, j = 1, \dots, k\}$ of id number (i.e. the node id), the matrix \underline{Y} is obtained through to the conversion between node id and the associated binary vector \underline{t} .

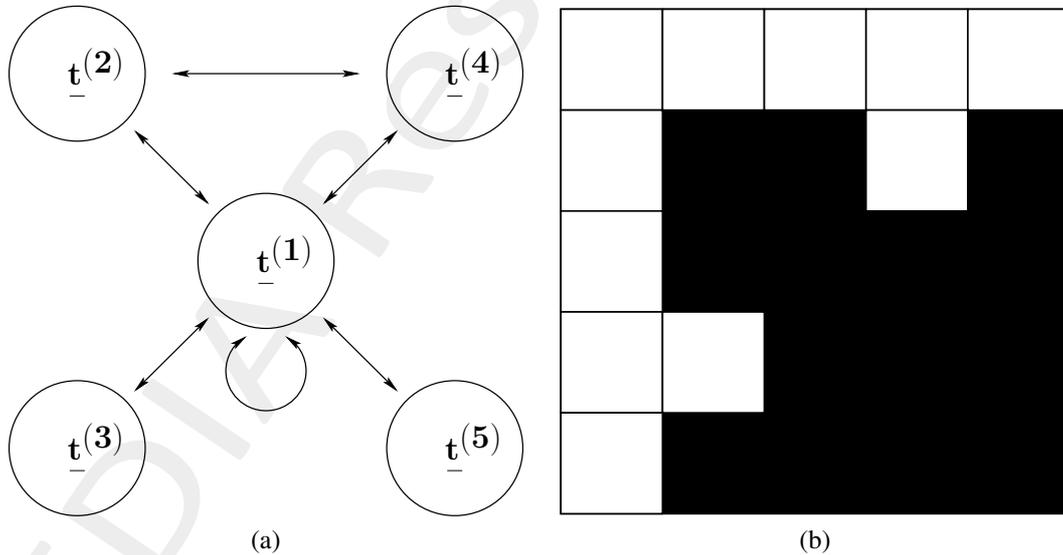


Figure 16: $R = A \times 5$ - Graph Λ (a) and Adjacency Matrix \underline{G} (b).

$k = 4$

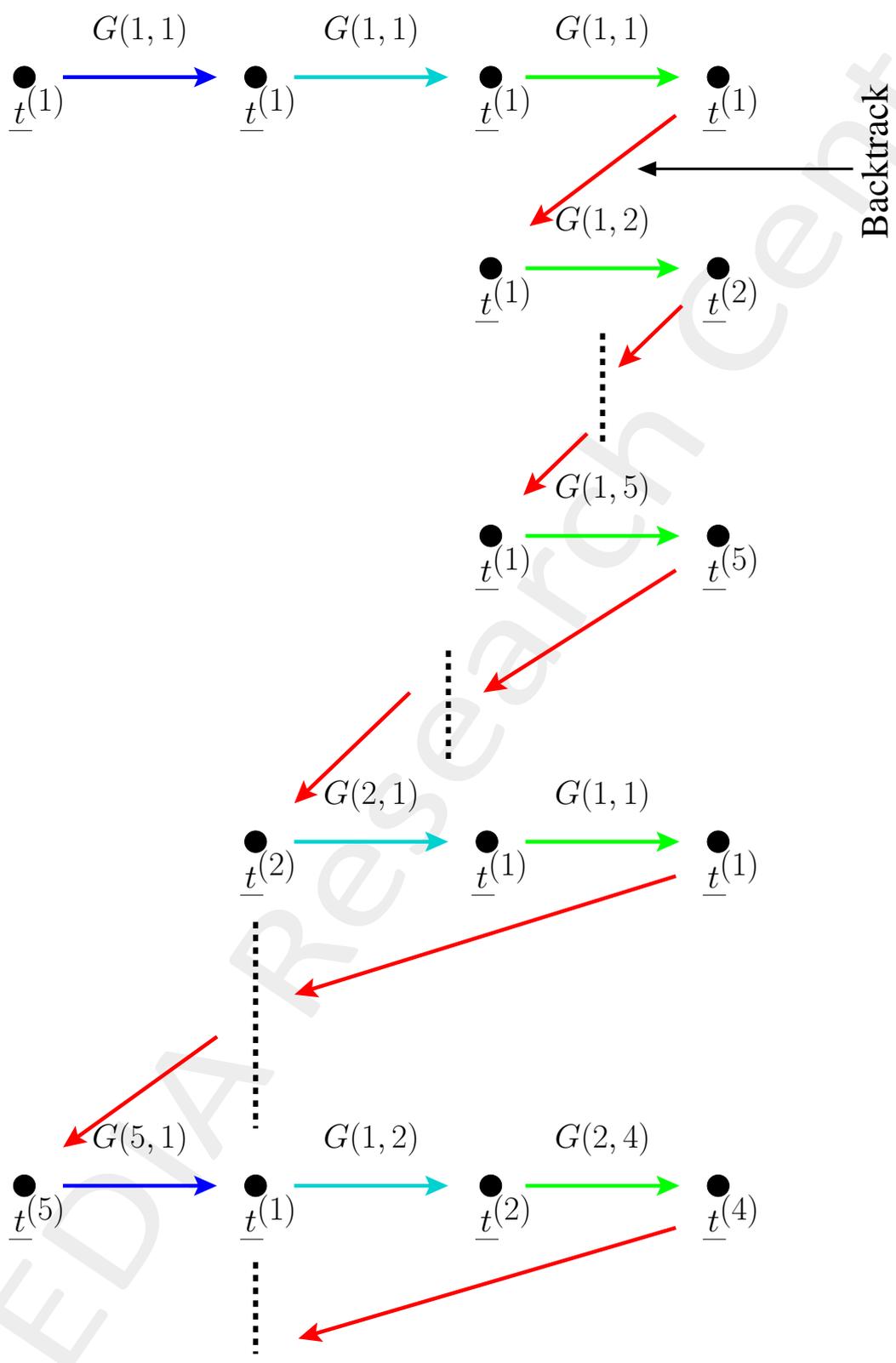


Figure 17: $R = A \times 5$ - Iterative Backtracking Algorithm Illustration.

3.4.2 Binary Matrix-Based Tiling Method - $(m \times m, sm \times sm) - BMTM$

Let us consider a rectangular region R of \mathbb{Z}^2 with size $A \times B$ and a set of square tiles $m \times m$ and $sm \times sm$, $A, B, m, s \in \mathbb{N}$. Proven that such a region is coverable/tileable (Theorem 1) with such tiles, we are interested in explore whole set of allowed configurations/tilings $\underline{\mathcal{T}}$. In order to do that a suitable encoding for each tiling \underline{v} of R is needed. Broadly speaking, the $(m \times m, sm \times sm)$ -tiling method is a generalization of the $(m \times m, 2m \times 2m)$ -tiling method introduced in 3.4.1.

The specific case in which the tilings of R is performed using squares of size 1 and s will be investigated, the generalization to the $(m \times m, sm \times sm)$ -tiling will be introduced after.

Matrix Representation

Similarly to the $(m \times m, 2m \times 2m) - BMTM$ it is possible to introduce binary matrix \underline{M} to represent one admissible tiling of R . This matrix is constructed by placing ones in the matrix cell corresponding to the index associated the lower left corner of the n -tile (the “big” tile) on the square lattice of R , and zeros elsewhere. For the case of interest since $n = s$ and $k = A - (s - 1)$ and $q = B - (s - 1)$, the first $s - 1$ rows and last $s - 1$ columns of \underline{M} are always zero vectors, thus it is sufficient to consider a $k \times q$ matrix $\underline{M}_{k,q}$ instead of a $A \times B$ matrix. See Fig. 18 for an illustrative example with $R = 6 \times 5$.

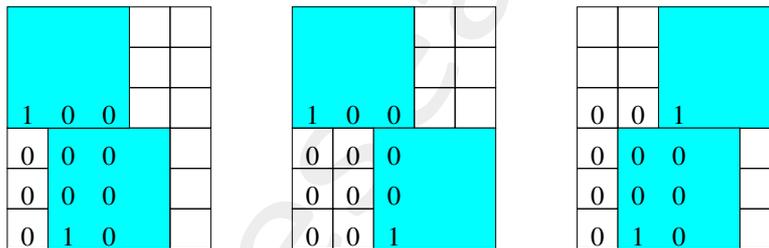


Figure 18: Binary Matrix Representation, $R = 6 \times 5$ - Three $(1 \times 1, 3 \times 3)$ -tilings.

The associated binary matrix are:

$$\underline{M}_{4,3}^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \underline{M}_{4,3}^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \underline{M}_{4,3}^{(3)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Binary Words

Let now consider a generic row (binary word of q letters) of $\underline{M}_{k,q}$. From to the definition of tiling (Par. 3.1) the overlap between tiles is not allowed. Therefore, in the exhaustive generation of candidate binary words (rows) the following constraint must be considered:

- **Inter-Letters Dependency:** Each letter in a word is dependent from its neighbours, i.e. considering a generic

binary word $\underline{r}_x = \{r_{x,1}, r_{x,2}, \dots, r_{x,q}\}$ with $r_{x,i} = 1, i \in \{1, \dots, q\}$ it is not admissible to have ones in the range $[i - (s - 1), i + (s - 1)] \cap \{1, \dots, q\}$.

The resulting set of allowed words $\underline{T} = \{\underline{t}^{(i)}, i = 1, \dots, |\underline{T}|\}$ is generated using a binary counter combined with the inter-letter dependency check defined above. If admissible, the binary string is stored in \underline{T} otherwise it will be discarded. This operation is similar to that presented for sec. 3.4.1. The only difference is in the Inter-Letters dependency check which has been customized in order to deal with the (possibly) bigger n -tile ($n = sm$).

Graph Representation

Once the set \underline{T} has been defined according to inter-letter dependency criterion, a tiling \underline{v} of R can be constructed using the words in \underline{T} . As done for the $(m \times m, 2m \times 2m)$ -tiling, also in the $(m \times m, sm \times sm)$ -tiling method in order to perform the generalization (i.e. generate \underline{Y} starting from \underline{T}) it is necessary to define a criterion for the words aggregation which satisfies the tiling definition (Par. 3.1). For this purpose the following constraint has been imposed:

- **Inter-Words Dependency:** Each word (row) in $\underline{M}_{k,q}$ is dependent from its neighboring words. The row $\underline{r}_x = \{r_{x,1}, r_{x,2}, \dots, r_{x,q}\}$ can be placed on top of $\underline{r}_{x+1} = \{r_{x+1,1}, r_{x+1,2}, \dots, r_{x+1,q}\}$ iff $r_{x+n,j} = 0$ for $j \in \{i - (s - 1), \dots, i + (s - 1)\} \cap \{1, \dots, q\}$; $n \in \{1, \dots, (s - 1)\} \cap \{1, \dots, k\}$ for all i such that $r_{x,i} = 1$ and $x \in \{1, \dots, k\}$. Note that n define the degree/depth of the inter-words check.

The admissible words $\underline{t} \in \underline{T}$ are now represented as nodes in a finite graph Λ , where there is an edge between two nodes, $\underline{t}^{(g)}$ and $\underline{t}^{(h)}$ if and only if $\underline{t}^{(g)}$ can be placed next to $\underline{t}^{(h)}$ such that there are no inter-words dependencies and $g, h \in \{1, \dots, k\}$.

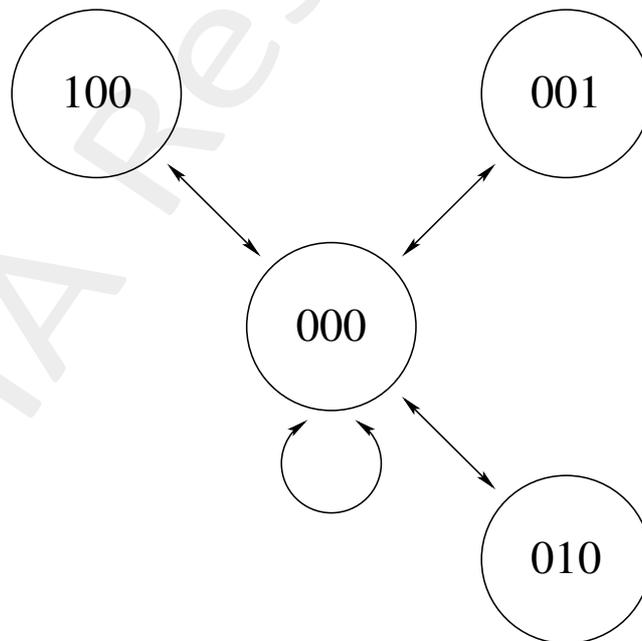


Figure 19: Graph Representation of the Words: $\Lambda, R = 5 \times 4 - (1 \times 1, 2 \times 2)$ -Tiling.

The resulting graph Λ encode all the possible combination of words, which means that it encode all the information needed to explore all the tilings $\underline{v} \in \underline{Y}$ of R . Straightforwardly any path of length k construct on skeleton of Λ produce a sequence

of binary words (or matrix $\underline{M}_{k,q}$) representing a tiling \underline{v} for R . In the example of Fig. 18 the paths are respectively:

$$\underline{M}_{4,3}^{(1)} = \begin{pmatrix} 100 \\ \downarrow \\ 000 \\ \downarrow \\ 000 \\ \downarrow \\ 010 \end{pmatrix}, \quad \underline{M}_{4,3}^{(2)} = \begin{pmatrix} 100 \\ \downarrow \\ 000 \\ \downarrow \\ 000 \\ \downarrow \\ 001 \end{pmatrix}, \quad \underline{M}_{4,3}^{(3)} = \begin{pmatrix} 001 \\ \downarrow \\ 000 \\ \downarrow \\ 000 \\ \downarrow \\ 010 \end{pmatrix}$$

The General Case: $(m \times m, sm \times sm)$ – Tiling

The coding implemented by the binary matrix can be extended to the case of a generic tiling \underline{v} of a rectangular region $R = (Am) \times (Bm)$ iff it is performed by means of square tiles of size m and $n = sm$ with $m, s \in \mathbb{N}$. Thus, the $(m \times m, sm \times sm)$ –tiling problem is addressed through a recast of the $(1 \times 1, s \times s)$ –tiling on proportionally scaled the region $R' = A \times B$.

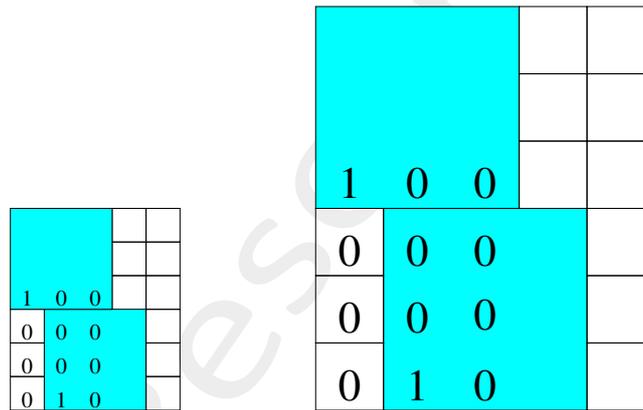


Figure 20: Binary Matrix Representation - $(1 \times 1, 3 \times 3)$ –Tiling $R = 6 \times 5$ and $(2 \times 2, 6 \times 6)$ –Tiling $R = 12 \times 10$.

Resume

Here are reported the steps that must be followed In order to exhaustively retrieve all the $(1 \times 1, s \times s)$ -tilings $\underline{\Upsilon}$ of a rectangular region $R = A \times B$.

- Step 0: Define k and q as, $k = A - (s - 1)$ and $q = B - (s - 1)$;
- Step 1: Generate $\underline{T} = \{\underline{t}^{(i)}, i = 1, \dots, |\underline{T}|\}$ according to the **Inter-Letter** dependency;
- Step 2: Construct the graph Λ (\underline{G}) according to the **Inter-Words** dependency;
- Step 4: For each node of Λ (i.e. $\underline{t} \in \underline{T}$) find all the path on the skeleton of Λ having a length of k . In agreement with the theory, we know that each paths is a tiling \underline{v} of R . Hence, the exploration of the whole set of these paths provides the exhaustive generation of $\underline{\Upsilon}$.

The procedure is identical to that proposed in sec. 3.4.2, except for the Inter-Letter and Inter-Words checks which have been customized. Hence, without loss of generality, the reader can refer to Algorithm 1.

ELEDIA Research Center

3.5 Optimization-based Tiling Methods

When the dimension of the aperture and the cardinality of the corresponding solution space does not allow a computationally feasible application of the enumerative approach (Section 3), the aperture tiling is obtained through an optimization-based method. Let us consider a generic rectangular region R of \mathbb{Z}^2 with size $A \times B$ ($A, B \in \mathbb{N}$) and a set of square tiles $m \times m$ and $n \times n$ where $m, n \in \mathbb{N}$ subject to $n = sm$ and $s \in \mathbb{N}$. Proven that such a region is coverable/tileable with the available tiles (Theorem 1), we are interested in the tiling/configuration \underline{v}^{opt} (or set of configurations) the best fulfills the user defined requirements/objectives. Indeed, the overall search process is aimed at the optimization of a suitable cost function $\Phi(\underline{f})$ or functions $\underline{\Phi}(\underline{f}) = \{\Phi_1(\underline{f}), \dots, \Phi_T(\underline{f})\}$.

3.5.1 Optimization Algorithms

The choice of the optimization algorithm is guided by the problem itself as well as the complexity of $\Phi(\underline{f}^{(p)})$. Local/deterministic search algorithms are usually adopted when the functional presents a single global minima/maxima (i.e. $\Phi(\underline{f}^{(p)})$ convex). In presence of non-convex functional this deterministic strategies suffer from the presence of local minima/maxim, in which they risk to be trapped without a prior domain knowledge. On the contrary global/stochastic algorithms are potentially able of escaping from local minima and of finding the global optimum without the need of additional information.

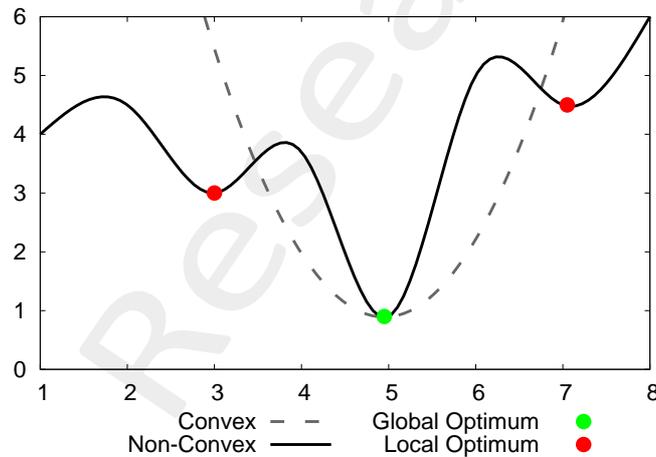


Figure 21: Convex vs. Non-convex function.

The problem of array synthesis is usually characterized by highly nonlinear functional which are likely to exhibit multiple local minima/maxima (non-convex functional). Therefore, the implementation of a stochastic/global optimizer is in general the best choice. Among the possible candidates, the Genetic Algorithm (GA) is certainly one of the most suitable optimizer for the synthesis of large antenna array. Besides the explicit parallelism guaranteed by its multiple-agent nature, the GAs are also related to the concept of *schemata* and the *implicit* parallelism. It has been shown that the effective number of schemata processed by the GA at each generation is greater than the number of individuals of the population. Such property guarantees that multiple characteristics of the solution are processed in parallel. This features assures that the GAs are effective also in presence of an high-dimensional solution space.

Single Objective Optimization: GA

The GAs are Evolutionary Algorithms (EAs) based on the Darwinian theory of evolution. The algorithm reproduce the natural selection process where the individuals compete with each others and only the fittest are able to reproduce and generate offsprings. Using the notation usually adopted in genetics, let us define:

- Individual \underline{f} : Trial solution;
- Chromosome \underline{c} : Coded individual/solution;
- Population $F = \{\underline{f}^{(p)}, p = 1, \dots, P\}$: A set of P agents/individuals;

The evolution process is implemented through a set of genetic operators \mathcal{L}^{GA} (Selection, \mathcal{S} , crossover, \mathcal{C} and mutation, \mathcal{M}) which are applied to a population. For instance at the k th iteration/generation the offsprings (F_{k+1}) of the population $F_k = \{\underline{f}_k^{(p)}, p = 1, \dots, P\}$ are generated according to the following reproduction cycle:

1. Selection: The *mating pool* is chosen applying the selection procedure (tournament, roulette wheel , etc...) on F_k

$$F_{k(S)} = \mathcal{S}\{F_k\} \quad (11)$$

2. Crossover:

$$F_{k(C)} = \mathcal{C}\{F_{k(S)}\} \quad (12)$$

3. Mutation:

$$F_{k(\mathcal{M})} = \mathcal{M}\{F_{k(C)}\} \quad (13)$$

The genetic operators are iteratively applied on the mating pool. The new population F_{k+1} is defined as:

$$F_{k+1} = F_{k(C)} \cup F_{k(\mathcal{M})} \quad (14)$$

Sometimes in order to enhance the convergence of behaviour of the algorithm the best individuals of F_k are injected in the new population F_{k+1} in place of the worst solutions. This is what is called *elitism*. The general flowchart of a simple GA implementation is reported in the figure below.

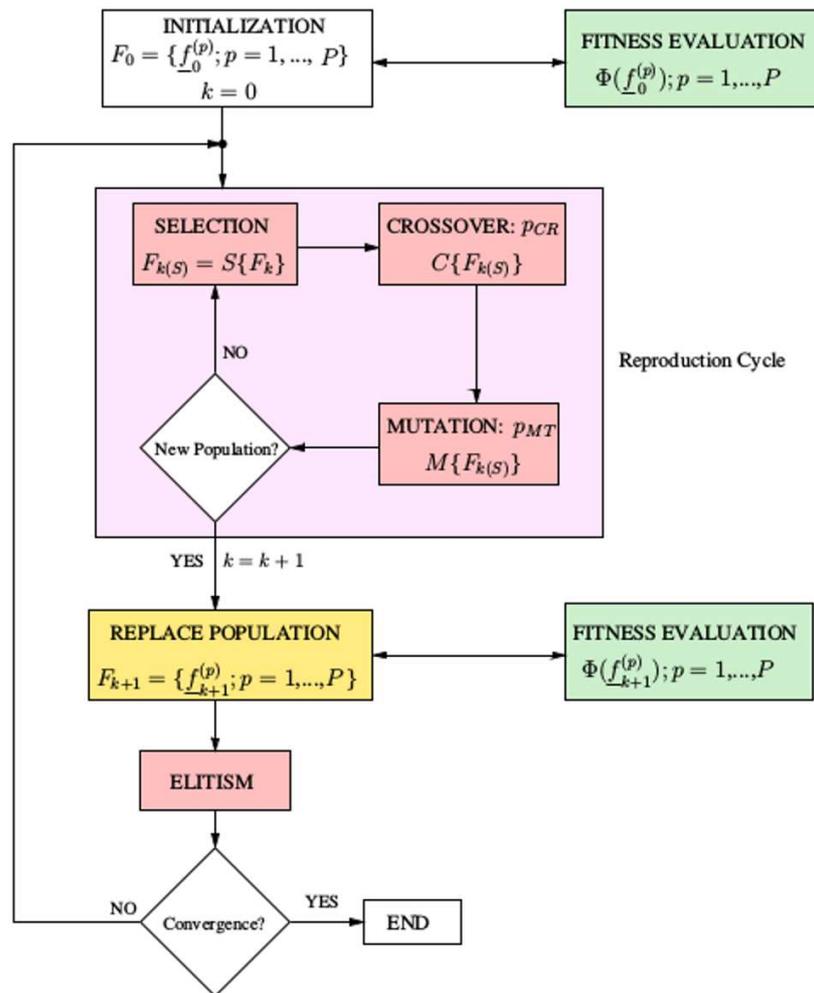


Figure 22: GA Flowgraph.

Multiple Objective Optimization: NSGA-III

Evolutionary Algorithms have been largely adopted in MOPs since the 90s. The extension of standard single-objective EAs to a multi-objective version requires the integration of additional operators. Two of the most common operators introduced in Multi-objective Evolutionary Algorithms (MOEAs) are aimed at providing:

- Non-dominated sorting: This operator is used to identify solutions that in absence of further information can not be said to be better than others. This non-dominated solutions (or Pareto-optimal solutions) compose the Pareto front;
- Diversity among solutions: In order to ensure a satisfactory exploration of the Pareto-optimal front it is necessary to preserve *diversity* between the solutions;

Furthermore, EAs are multi-agent based algorithms which, even in their simplest version, make them suitable to explore several solutions in a single run.

The Non-dominated-Sorting Genetic Algorithm (NSGA) has been one of the first and most widespread MOEAs. The NSGA have been evolving to meet the requirements imposed by new challenging application scenarios.

The main concerns moved on where:

1. The computational complexity of the non-dominated sorting: The naive algorithm implementations presents an high computational complexity;
2. Elitism: The lack of elitism was one of the main concerns of the first MOEAs. Elitism allows the speed up of the algorithm as well the preservation of good solutions;
3. Sharing parameter specification: The diversity preservation operator of NSGA requires the user to define the parameter θ . Usually parameter-less operator are preferred;

The second version of the NSGA (NSGA-II) provides a fast elitist multi-objective optimization algorithm which is based on a fast non-dominated sorting approach and a crowding-distance diversity operator able to overcome the above limitations.

In the recent years, due to the growing demand of handling more than just two or three objectives, the researchers have been focused on the development of many-objective⁽²⁾ optimization algorithms. Neglecting the problem arising when dealing with high-dimensional functional, increasing the number of objectives above four involves the following problems:

1. Non-domination of a large fraction of the population: It is clear that the fraction of non-dominated solution gets larger as the number of objectives grows. This drawback could prevent the generation of new solutions, leading to a poor exploration of the solutions space;
2. Computing diversity measure in high-dimensional space became expensive: The computation of the crowding distance in a high dimensional space results to be computationally expensive;
3. Crossover operator may be inefficient: In a high-dimensional space the solutions tend to be very distant from each other. If the standard crossover operation has to be used on two distant parents, the resulting offsprings will be probably distant from the parents;

The latest NSGA (NSGA-III) optimizer tries to solve the above limitations through the use of a predefined multiple targeted search. In the following the algorithm will be depicted.

NSGA-III

- Initialization: The initial population F_0 of P individuals is created and sorted in different non-domination levels (L_1, L_2 , etc) according to a fast non-domination method. The offsprings Q_0 of size P is generated using the classical GA reproduction cycle (selection, crossover and mutation);
- Main loop (k th iter.):
 1. Merge the parents F_k and the offsprings Q_k to obtain a combined population $R_k = F_k \cup Q_k$ of size $2P$;
 2. Sort R_k into non-domination levels (L_1, L_2 , etc);
 3. Generate the new population F_{k+1} choosing the individuals starting from the first non-domination level L_1 until $\sum_{i=1}^l |L_i| < P$, l is the index of the last non-domination level L_l that can be fully or partially accommodate in F_{k+1} . The individuals from L_{l+1} onward are rejected;

⁽²⁾Multi-objective problems implies two or three objective, many-objective implies four or more objectives.

-
- (a) If $\sum_{i=1}^l |L_i| = P$ go to point 4;
- (b) If $\sum_{i=1}^l |L_i| > P$ only a sub-set of solutions in L_l can be accommodated in F_{k+1} such that the size of the population do not exceed P , the others are rejected. The diversity operator is applied to select the best individuals in L_l to be inserted in F_{k+1} ;

4. Generate the offsprings Q_{k+1} using the classical GA reproduction cycle (selection, crossover and mutation);

The core of the NSGA-III algorithm is almost the same of the NSGA-II except for the diversity preservation operation. The expansive crowding distance measure has been replaced by a multiple targeted search strategy. The elitism is still guaranteed through a comparison (point 2 of the main loop) between the offsprings and the parents, which actually involves the previous best non-dominated solutions. Moreover, the use of a predefined multiple targeted search allows to mitigate the problems related to the many-objective optimization problems. The algorithm flowthrough has been resumed in the following figure.

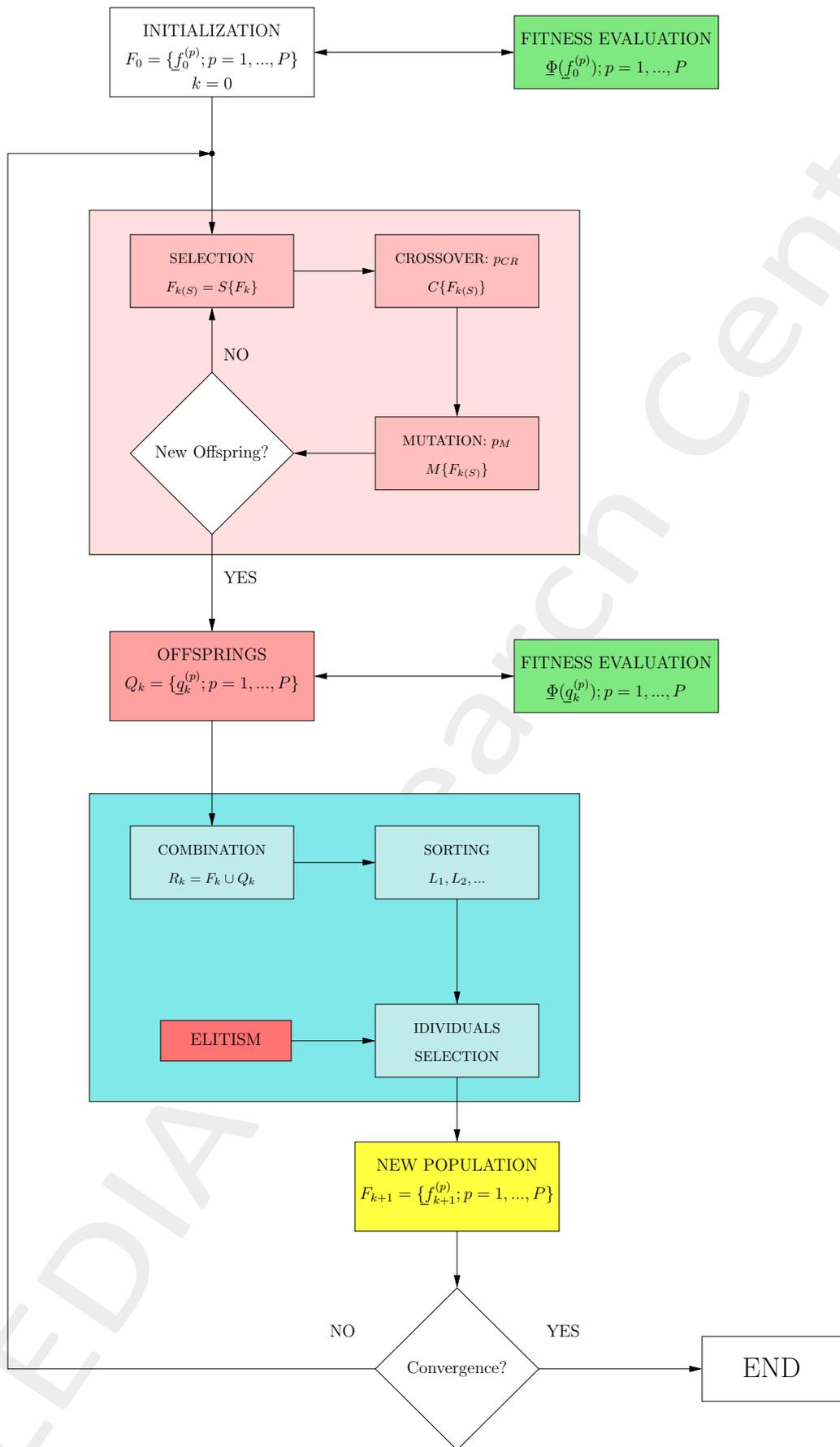


Figure 23: NSGA-III Flowgraph

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

References

- [1] 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.
- [2] M. Salucci, G. Gottardi, N. Anselmi, and G. Oliveri, "Planar thinned array design by hybrid analytical-stochastic optimization," *IET Microwaves, Antennas & Propagation*, vol. 11, no. 13, pp. 1841-1845, Oct. 2017
- [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 design," *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] G. Oliveri, A. Gelmini, A. Polo, N. Anselmi, and A. Massa, "System-by-design multi-scale synthesis of task-oriented reflectarrays," *IEEE Trans. Antennas Propag.*, vol. 68, no. 4, pp. 2867-2882, Apr. 2020.

-
- [14] 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.
- [15] M. Salucci, F. Robol, N. Anselmi, M. A. Hannan, P. Rocca, G. Oliveri, M. Donelli, and A. Massa, "S-Band spline-shaped aperture-stacked patch antenna for air traffic control applications," *IEEE Tran. Antennas Propag.*, vol. 66, no. 8, pp. 4292-4297, Aug. 2018.
- [16] 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
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.

-
- [29] 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.
- [30] G. Oliveri, "Multi-beam antenna arrays with common sub-array layouts," *IEEE Antennas Wireless Propag. Lett.*, vol. 9, pp. 1190-1193, 2010.
- [31] L. T. P. Bui, N. Anselmi, T. Isernia, P. Rocca, and A. F. Morabito, "On bandwidth maximization of fixed-geometry arrays through convex programming," *Journal of Electromagnetic Waves and Applications*, vol. 34, no. 5, pp. 581-600, 2020.
- [32] 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.
- [33] 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.
- [34] G. Oliveri and T. Moriyama, "Hybrid PS-CP technique for the synthesis of n-uniform linear arrays with maximum directivity," *Journal of Electromagnetic Waves and Applications*, vol. 29, no. 1, pp. 113-123, Jan. 2015.
- [35] P. Rocca and A. Morabito, "Optimal synthesis of reconfigurable planar arrays with simplified architectures for monopulse radar applications," *IEEE Trans. Antennas Propag.*, vol. 63, no. 3, pp. 1048-1058, Mar. 2015.
- [36] A. F. Morabito and P. Rocca, "Reducing the number of elements in phase-only reconfigurable arrays generating sum and difference patterns," *IEEE Antennas and Wireless Propagation Letters*, vol. 14, pp. 1338-1341, 2015.
- [37] P. Rocca, N. Anselmi, and A. Massa, "Optimal synthesis of robust beamformer weights exploiting interval analysis and convex optimization," *IEEE Trans. Antennas Propag.*, vol. 62, no. 7, pp. 3603-3612, Jul. 2014.
- [38] A. Morabito and P. Rocca, "Optimal synthesis of sum and difference patterns with arbitrary sidelobes subject to common excitations constraints," *IEEE Antennas Wireless Propag. Lett.*, vol. 9, pp. 623-626, 2010.
- [39] D. Sartori, G. Oliveri, L. Manica, and A. Massa, "Hybrid design of no-regular linear arrays with accurate control of the pattern sidelobe," *IEEE Trans. Antennas Propag.*, vol. 61, no. 12, pp. 6237-6242, Dec. 2013.