On the Design of Complex *EM* Devices and Systems through the System-by-Design Paradigm - A Framework for Dealing with the Computational Complexity

A. Massa,[(1)(2)(3)(4)] *Fellow, IEEE*, and M. Salucci,[(2)(4)] *Member, IEEE*

[(1)] *ELEDIA Research Center* (*ELEDIA@UESTC* - UESTC)
School of Electronic Engineering, Chengdu 611731 - China
E-mail: *andrea.massa@uestc.edu.cn*
Website: *www.eledia.org/eledia-uestc*

[(2)] *ELEDIA Research Center* (*ELEDIA@UniTN* - University of Trento)
Via Sommarive 9, 38123 Trento - Italy
E-mail: {*andrea.massa*, *marco.salucci*}*@unitn.it*
Website: *www.eledia.org/eledia-unitn*

[(3)] *ELEDIA Research Center* (*ELEDIA@TSINGHUA* - Tsinghua University)
30 Shuangqing Rd, 100084 Haidian, Beijing - China
E-mail: *andrea.massa@tsinghua.edu.cn*
Website: *www.eledia.org/eledia-tsinghua*

[(4)] *ELEDIA Research Center* (*ELEDIA@L2S* - UMR 8506)
3 rue Joliot Curie, 91192 Gif-sur-Yvette - France
E-mail: {*andrea.massa*, *marco.salucci*}*@l2s.centralesupelec.fr*
Website: *www.eledia.org/eledia-l2s*

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

arXiv:2107.09521v1 [eess.SP] 16 Jul 2021

On the Design of Complex *EM* Devices and Systems through the System-by-Design Paradigm - A Framework for Dealing with the Computational Complexity

A. Massa and M. Salucci

Abstract

The System-by-Design (*SbD*) is an emerging engineering framework for the optimization-driven design of *complex* electromagnetic (*EM*) devices and systems. More specifically, the computational complexity of the design problem at hand is addressed by means of a suitable selection and integration of *functional blocks* comprising problem-dependent and computationally-efficient modeling and analysis tools as well as reliable prediction and optimization strategies. Thanks to the suitable re-formulation of the problem at hand as an optimization one, the profitable minimum-size coding of the degrees-of-freedom (*DoFs*), the "smart" replacement of expensive full-wave (*FW*) simulators with proper *surrogate models* (*SM*s), which yield fast yet accurate predictions starting from minimum size/reduced *CPU*-costs training sets, a favorable "environment" for an optimal exploitation of the features of global optimization tools in sampling wide/complex/nonlinear solution spaces is built. This research summary is then aimed at (*i*) providing a comprehensive description of the *SbD* framework and of its pillar concepts and strategies, (*ii*) giving useful guidelines for its successful customization and application to different *EM* design problems characterized by different levels of computational complexity, (*iii*) envisaging future trends and advances in this fascinating and high-interest (because of its relevant and topical industrial and commercial implications) topic. Representative benchmarks concerned with the synthesis of single antenna devices as well as complex array systems are presented to highlight advantages and potentialities as well as current limitations of the *SbD* paradigm.

Key words: Complex *EM* Problems, Optimization, Surrogate Modeling, Learning-by-Examples (*LBE*), System-by-Design (*SbD*).

1 Introduction

In the last years, there have been many and significant progresses in the development of numerical techniques - denoted as full-wave (*FW*) solvers - for the accurate analysis of complex electromagnetic (*EM*) devices and systems (see for examples [1]-[8] and the reference therein). Although highly-reliable, *FW* solvers are generally time-consuming [8] and their exploitation to solve *complex EM* synthesis problems[(1)] often implies "local" refinements of an initial/reference solution based on parametric sweeps and/or trial-and-error steps. Otherwise, design strategies involving analytic/semi-analytic methods (*AM*s) (e.g., [9][10]) are generally less computationally demanding, thus allowing the use of more effective and complex synthesis strategies (e.g., global optimization or gradient-based deterministic and iterative methods), but they may be unreliable when dealing with high-complexity systems since they typically approximate or even neglect nonlinear *EM* phenomena that require the *FW* solution of Maxwell's equations. Moreover, *AM*s cannot deal with whatever *EM* device or system, while they are generally suitable for canonical or rather "simple" structures [10]. Therefore the "holy-grail" in synthesizing complex EM systems is, on the one hand, to take advantage of the modeling accuracy of *FW* solvers, on the other, to exploit global optimization strategies for finding the global optimum (or the closest one) of the cost function that quantifies the mismatch between user-requirements and design outcomes. As a matter of fact, global optimization strategies, based for example on nature-inspired evolutionary algorithms (*EA*s) [11]-[15], have been widely applied in many *EM* engineering problems since they allow an efficient exploration of the whole solution space and, unlike deterministic algorithms, they require neither the analytic knowledge nor the differentiation of the cost function. Moreover, *a-priori* information (e.g., physical requirements or already available sub-optimal solutions) can be introduced in a straightforward manner as additional constraints on the iterative process of selecting trial solutions [13]. Of course, global optimizers require the evaluation of many solutions (typically hundreds or thousands) to ensure an effective sampling of the solution space and to find a solution fitting all user requirements, thus the "bare" integration of a *FW* solver in an iterative optimization tool will imply unreal-

[(1)] A problem is regarded as inherently *complex* if its solution requires significant resources, whatever the algorithm used. In other words, "complexity" is a "measure" of (*i*) the problem dimension, (*ii*) the adopted mathematical model, as well as (*iii*) the computational burden.

istic/unaffordable computational costs. In order to overcome those issues, different approaches have been proposed ranging from (*i*) the improvement of the convergence rate of *EA*s by (*i.1*) using a set of "good" (i.e., close to the global optimum) trial solutions at the initialization of the optimization process [16][17] and/or by (*i.2*) identifying a minimum set of representative solution parameters [18][19] up to (*iii*) the reduction of the time for evaluating (i.e., the computation of the mismatch cost function) a single solution [20]-[24] also integrating suitable coarse-to-fine *space mapping* strategies [25][26].

Within this context, the System-by-Design (*SbD*) recently emerged as an innovative paradigm able to exploit such strategies in a more integrated and seamless fashion [27]-[34]. As a matter of fact, the *SbD* enables an effective, reliable, and computationally-efficient use of global optimizers for addressing complex *EM* design problems, since it is aimed at the "*task-oriented design, definition, and integration of system components to yield EM devices with user-desired performance having the minim costs, the maximum scalability, and suitable reconfigurability properties*". Applications of the *SbD* to the synthesis of innovative meta-materials [27]-[29], fractal antennas [30], electrically-large airborne radomes [31], wide angle impedance matching layers (*WAIM*s) [32][33], and reflectarray antennas [34] have been recently documented.

The aim of this work is (*i*) to provide a comprehensive description of the *SbD* framework and of its pillar concepts and strategies, (*ii*) to give useful guidelines for its successful customization and application to different *EM* design problems that share the common issue of the computational complexity, (*iii*) to envisage future trends and advances in this fascinating and high-interest (because its relevant and topical industrial and commercial implications) topic.

The outline of the paper is as follows. The general principles of the *SbD* are pointed out in Sect. II, while the description of the *functional blocks* of the *SbD* is given in Sect. III. Two novel advanced *SbD*-based synthesis strategies are presented in Sect. IV. Representative synthesis benchmarks are illustrated (Sect. V) to show the *SbD* working as well as to give some proofs of the method effectiveness and efficiency when dealing with computational complexity issues. Some concluding remarks are finally drawn also envisaging future trends (Sect. VI).

2 The *SbD* Paradigm

By denoting with $\underline{\Omega}$ ($\underline{\Omega} = \{\Omega_k;\ k = 1,\ ...,\ K\}$) the set of K descriptive parameters [i.e., the degrees-of-freedom (*DoF*s)] of the design problem at hand, the goal of the *SbD* is to yield, in a *reasonable* time frame[(2)] , the setup of the *DoF*s, $\underline{\Omega}^{(end)}$, so that the corresponding cost function value, $\Phi\left\{\underline{\Omega}^{(end)}\right\}$, differs from that of the global optimum, $\Phi\left\{\underline{\Omega}^{(opt)}\right\}$, at most for a maximum deviation ξ

$$\underline{\Omega}^{(end)}:\ \left|\Phi\left\{\underline{\Omega}^{(end)}\right\} - \Phi\left\{\underline{\Omega}^{(opt)}\right\}\right| \leq \xi \tag{1}$$

Φ being a metric that quantifies the mismatch of a problem solution, $\underline{\Omega}$, from the project/user requirements, $\underline{\Gamma}^{th}$,

$$\Phi\left\{\underline{\Omega}\right\} \triangleq \left\|\underline{\Gamma}^{th} - \underline{\Upsilon}\left\{\underline{\Omega}\right\}\right\|^2 \tag{2}$$

where $\underline{\Upsilon}$ is the mapping function between the solution space, $\Re\left\{\underline{\Omega}\right\}$, and the requirements space, $\Re\left\{\underline{\Gamma}\right\}$ ($\underline{\Upsilon}:\ \Re\left\{\underline{\Omega}\right\} \rightarrow \Re\left\{\underline{\Gamma}\right\}$, being $\Re\left\{\underline{\Gamma}\right\} \equiv \Re\left\{\Phi\right\}$).

To find a computationally-efficient solution of such a design problem (1), the *SbD* exploits four interconnected *functional blocks* devoted to the following *sub-tasks* (Fig. 1):

1. *Requirements and Constraints Definition* (*RCD*) - Mathematical definition of the project requirements, $\underline{\Gamma}^{th}$, and of a set of physical-admissibility constraints, $G_a^{min} \leq G_a\left(\underline{\Omega}\right) \leq G_a^{max}$ ($a = 1,\ ...,\ A$), starting from specifications, guidelines, and objectives provided by the end-users in the statement of work (*SoW*);

2. *Problem Formulation* (*PF*) - Mathematical re-formulation of the synthesis problem as an optimization one by (*i*) selecting/defining a parametric model of the solution, (*ii*) identifying the corresponding *DoF*s, $\underline{\Omega}$, that is the minimum number of univocally representative model descriptors, and (*iii*) choosing the cost function, $\Phi\left\{\underline{\Omega}\right\}$, which mathematically codes the mismatch between the project requirements/constraints and the performance of the model, $\underline{\Upsilon}\left\{\underline{\Omega}\right\}$, whose global optimum corresponds to the best admissible physical solution of the problem at hand;

[(2)]Clearly, the expression is rather vague and intuitive. Of course, there is the need of quantify the meaning of "*reasonable time frame*" in a rigorous mathematical way for moving from a purely qualitative statement to a more engineer-oriented/quantitative one (see Sects. 3-4).

3. *Cost Function Computation* (*CFC*) - Evaluation, in the most efficient and accurate way, of the cost function value $\Phi\left\{\underline{\Omega}^{(p)}\right\}$ of a trial solution $\underline{\Omega}^{(p)}$ in order to "quantify" the optimality of the p-th ($p = 1, ..., P$) trial physical solution (i.e., the fitness of this solution to the problem at hand);

4. *Solution Space Exploration* (*SSE*) - Dealing with an optimization problem, the task to carry out is the sampling of the solution space, $\Re\{\underline{\Omega}\}$, bounded by $G_a^{min} \leq G_a(\underline{\Omega}) \leq G_a^{max}$ ($a = 1, ..., A$), to look for the global optimum of the cost function $\Phi\{\underline{\Omega}\}$ [i.e., $\underline{\Omega}^{(end)}$ in (1)]. Therefore, the SSE block is aimed at generating a succession of I trial solutions $\{\underline{\Omega}_i^{(p)};\ i = 1, ..., I\}$ ($p = 1, ..., P$), P and I being the number of agents and of iterations, respectively, by means of a suitable optimization algorithm so that $\underline{\Omega}^{(end)} = \arg\left\{\min_{i=1,...,I}\left[\Phi_i^{best}\right]\right\}$ ($\Phi_i^{best} \triangleq \Phi\left\{\underline{\Omega}_i^{best}\right\}$) being $\underline{\Omega}_i^{best} = \arg\left\{\min_{p=1,...,P}\left[\Phi_i^{(p)}\right]\right\}$ ($\Phi_i^{(p)} \triangleq \Phi\left\{\underline{\Omega}_i^{(p)}\right\}$). Generally speaking, $\underline{\Omega}_{i+1}^{(p)} = \underline{\Omega}_i^{(p)} + \underline{v}_i^{(p)}$ where $\underline{v}_i^{(p)} = \left\{v_{i,k}^{(p)};\ k = 1, ..., K\right\}$ is a vectorial increment defined on the basis of a suitable set of operators $\boldsymbol{\mathcal{L}}$, $\underline{v}_i^{(p)} = \boldsymbol{\mathcal{L}}\left[\left(\underline{\Omega}_j^{(q)},\ \Phi_j^{(q)}\right);\ j = 1, ..., (i-1);\ q = 1, ..., P\right]$ depending on the optimization strategy at hand.

3 *SbD* Functional Blocks Implementation

To provide the readers with a general description of the *SbD* framework and implementation strategies, let us now focus on the *PF*, *CFC*, and *SSE* functional blocks, postponing the details of the first block (*RCD*) to the illustrative examples discussed later on.

3.1 Problem Formulation (*PF*)

In order to synthesize satisfactory and reliable solutions, all project requirements/constraints defined by the end-user must be carefully "translated" into a proper mathematical framework, as detailed in the following:

- Solution Model and *DoF*s Identification

In global search strategies, such as *EA*s, the number P of agents/trial solutions evaluated at each i-th ($i = 1, ..., I$) iteration is proportional to the number of *DoF*s, K [13][14]. Accordingly, a suitable formulation of the problem at hand must be made so that (*i*) K is as low as possible to minimize the computational burden for a more efficient design, but at the same time, (*ii*) the choice of the K *DoF*s, $\underline{\Omega}$, guarantees the existence of a solution fitting the *SbD* goal (1) as well as its careful correspondence with a feasible physical solution. Towards this end, parametric studies are a valid approach to perform a *sensitivity analysis* and to identify which descriptors (varied within suitable bounds) have the highest impact on the performance indexes, $\underline{\Upsilon}\{\underline{\Omega}\}$, and are the most representative to define the minimum set of K descriptors. On the other hand, a key factor in the "representation" of the actual physical solution is the choice of the basis functions to be exploited for defining *smarter* solution models, which are characterized by a reduced dimensionality, K, while ensuring a high flexibility in the solution representation. This is the case, for instance, when synthesizing the shape/profile of an *EM* device as in [27] where linear arrays have been miniaturized by means of isotropic covering meta-lenses. More in detail, the goal has been that of determining the optimal transformation-optics 2*D* profile, $\gamma(x, y)$, such that the lens-enclosed antenna mimic a reference one with larger aperture. By describing $\gamma(x, y)$ with a *pixel-basis* representation

$$\gamma^{(pix)}(x, y) = \sum_{k=1}^{K^{(pix)}} \Omega_k^{(pix)} B_k^{(pix)}(x, y) \tag{3}$$

where $B_k^{(pix)}(x, y) = 1$ if $(x, y) \in \rho_k$, while $B_k^{(pix)}(x, y) = 0$ otherwise, ρ_k being the k-th discretization cell of the lens, the arising number of *DoF*s was equal to the number of pixels, $K^{(pix)}$ [Fig. 2(*a*)]. Differently, a *spline-based* representation has been adopted in [27] to describe the lens profile in terms of Bezier quadratic curves by means of second order polynomials, $B_k^{(spl)}\left(x, y, \underline{\Omega}^{(spl)}\right)$ ($k = 1, ..., K^{(spl)}$) [18][19][31] so that the *DoF*s coincide with the *control points* of the spline curve, $\underline{\Omega}^{(spl)} = \left\{\left(x_k^{(spl)}, y_k^{(spl)}\right); k = 1, ..., K^{(spl)}\right\}$ [Fig. 2(*b*)]. It is worth pointing out that while the pixel-based representation (3) needs a significantly larger number of descriptors, $K^{(pix)} \gg K^{(spl)}$, to yield a detailed model of $\gamma(x, y)$, the use of spline

bases allows one to model a continuous profile with a limited number of control points (e.g., $K^{(spl)} = 5$ [27]). Of course, using a spline-based representation is not always the best solution, but certainly a careful study on the representation properties of a set of basis functions is a "*golden*" rule for having a competitive *SbD*-based design approach;

- Cost Function Definition

The cost function $\Phi\left\{\underline{\Omega}\right\}$ is the unique link between the optimization strategy and the physics of the *EM* problem. Therefore, it must be carefully defined to guarantee the correct sampling of the solution space and the achievement of feasible solutions [13]. Moreover, the choice of the cost function determines the overall *complexity* of the K-dimensional *landscape* explored by the *SSE* block. To the best of authors' knowledge, there is not a general guideline to optimally select $\Phi\left\{\underline{\Omega}\right\}$ as well as to have the arising cost function with suitable properties, for instance, to limit the occurrence of local minima/false solutions. However, similarly to what is done in inverse scattering [35][36], its behavior can be roughly *estimated* by analyzing the functional cuts along some directions of the solution space

$$\Phi\left\{t,\, \underline{\Omega}^{(1)},\, \underline{\Omega}^{(2)}\right\} = \Phi\left\{(1-t)\times\underline{\Omega}^{(1)} + t\times\underline{\Omega}^{(2)}\right\} \tag{4}$$

$\underline{\Omega}^{(1)}$ and $\underline{\Omega}^{(2)}$ being two user-chosen positions within the solution space, while t is a real variable. From (4), it turns out that $\Phi\left\{t,\, \underline{\Omega}^{(1)},\, \underline{\Omega}^{(2)}\right\} = \Phi\left\{\underline{\Omega}^{(1)}\right\}$ if $t = 0$ and $\Phi\left\{t,\, \underline{\Omega}^{(1)},\, \underline{\Omega}^{(2)}\right\} = \Phi\left\{\underline{\Omega}^{(2)}\right\}$ if $t = 1$, while sweeping t within suitable bounds gives some insights on the behavior of the cost function along a one-dimensional (1-*D*) cut passing through $\underline{\Omega}^{(1)}$ and $\underline{\Omega}^{(2)}$. What is the reason for doing it? On the one hand, such evaluations allows one to have some indications (not analytic proofs) on the degree of complexity/nonlinearity of the functional space at hand [(3)]. On the other hand, they can provide a valid support for the optimal choice and implementation of the *CFC* and *SSE* blocks.

[(3)]The *feasibility* of such analyses clearly depends on the computational complexity and *CPU*-time of each evaluation of the cost function. Of course, careful *a-priori* analyses must be performed case-by-case to infer about their worthiness and proper set-up.

3.2 Cost Function Computation (*CFC*)

The computation of $\Phi\left\{\underline{\Omega}\right\}$ requires the evaluation of specific performance indexes of the synthesized *EM* device. Towards this aim, several numerical techniques are available for performing accurate *FW* analyses [1][2][3]. By formulating forward *EM* problems by means of properly discretized sets of integral and/or differential equations, the corresponding numerical solution needs the computation of thousands or millions of unknowns. Therefore, a repeated evaluation of the cost function $\Phi\left\{\underline{\Omega}\right\}$ is the real *bottleneck* of standard (*StD*) optimization-based designs whose computational burden is

$$\Delta t_{StD} = (P \times I) \times \Delta t_{FW} \tag{5}$$

Δt_{FW} being the *CPU* time for a single *FW* simulation. Unless reliable analytic model are available, learning by examples (*LBE*) techniques are exploited by the *SbD* to significantly reduce the computational burden, while keeping a reliable prediction of the performance of the synthesized device/system. In short, *LBE*s are devoted to build fast *surrogate models* (*SM*s) able to *predict*, in a computationally-efficient fashion, the outcome of high-fidelity *EM* simulations [37]. From an architectural viewpoint, *LBE* strategies are two-step implementations composed by (*i*) a training and (*ii*) a testing phase. The training phase is typically performed *off-line* and it is devoted to build an accurate and fast *surrogate* of the cost function $\Phi\left\{\underline{\Omega}\right\}$, $\widetilde{\Phi}\left\{\underline{\Omega}\right\}$, starting from a training set of S examples/observations of the input/output (*I/O*) relationship, $\mathcal{D}_S = \left[\left(\underline{\Omega}^{(s)};\ \Phi^{(s)}\right);\ \ s = 1,\ ...,\ S\right]$ where $\Phi^{(s)}$ stands for $\Phi^{(s)} \triangleq \Phi\left\{\underline{\Omega}^{(s)}\right\}$ (Fig. 3). During the test phase, *on-line* predictions of the cost function value are then outputted for previously-unseen inputs [37].

Among several *LBE* strategies, let us focus in the following on the most commonly-adopted ones in *EM* engineering [37]. Radial Basis Function Networks (*RBFN*s) are popular artificial neural networks (*ANN*s) computing the surrogate $\widetilde{\Phi}\left\{\underline{\Omega}\right\}$ as a linear combination, through suitable real expansion coefficients, $\{w^{(s)};\ s = 1, ..., S\}$, of S Gaussian functions [37], $\{\psi^{(s)}\left\{\underline{\Omega}\right\};\ s = 1, ..., S\}$,

$$\widetilde{\Phi}\left\{\underline{\Omega}\right\} = \sum_{s=1}^{S} \psi^{(s)}\left\{\underline{\Omega}\right\} w^{(s)}. \tag{6}$$

Otherwise, Support Vector Regressors (*SVR*s) define the surrogate model as follows

$$\widetilde{\Phi}\left\{\underline{\Omega}\right\} = \sum_{s=1}^{S} \left[\left(\alpha^{(s)} - \beta^{(s)}\right) \mathcal{K}\left\{\underline{\Omega}^{(s)}, \underline{\Omega}\right\} \right] + \varsigma \tag{7}$$

where $\alpha^{(s)}$ and $\beta^{(s)}(s = 1, ..., S)$ are the *SVR* weights, while $\mathcal{K}\left\{\underline{\Omega}^{(s)}, \underline{\Omega}\right\}$ is the kernel function, ς being a bias [37]. A main difference between *RBFN*s and *SVR*s is the intrinsic capability of *RBFN*s to exactly fit/interpolate the training samples (i.e., $\widetilde{\Phi}\left\{\underline{\Omega}^{(s)}\right\} = \Phi\left\{\underline{\Omega}^{(s)}\right\}$, $s = 1, ..., S$). Otherwise, the *SVR* tolerates/neglects deviations of the surrogate prediction $\widetilde{\Phi}\left\{\underline{\Omega}\right\}$ from the actual cost function $\Phi\left\{\underline{\Omega}\right\}$ smaller than a threshold ϵ by defining an "ϵ-insensitive tube" [37].

Of course, there is not an optimal and unique choice for the best prediction technique, but this depends on the design problem at hand as well as on the selection of the remaining *SbD* blocks. Indeed, exactly performing like high-fidelity *FW* simulators when processing previously-explored solutions may be a desirable feature since the *I/O* relationship, $\underline{\Upsilon}\left\{\underline{\Omega}\right\}$, is purely deterministic. However, the *SM* should not be regarded in the *SbD* framework as a highly-reliable computationally-efficient alternative to *FW* solvers, but rather as a sufficiently-accurate estimator of the behavior of the cost function to guide the solution-space sampling/exploration towards the *attraction basin* of $\underline{\Omega}^{(opt)}$. In order to better understand this latter concept, let us consider some simple yet intuitive examples on well-known 1*D* ($K = 1$) benchmark cost functions, $\Phi\left\{\underline{\Omega}\right\} \triangleq \Phi\left\{\Omega_1\right\}$. Figure 4(*a*) shows the 1*D* Levy's cost function within the range $\Omega_1 \in [-10,\ 10]$ [38] along with the predictions made by the *RBFN* and the *SVR* surrogates starting from $S = 6$ randomly-chosen training samples. As it can be observed, the *SVR* correctly identifies the presence of a *valley* centered at the global minimum of the actual cost function, $\Omega_{Levy}^{(opt)} = 1$, while a significantly worse prediction of the cost function behavior is given by the *RBFN* even though this latter perfectly fits all training observations. However, the *SVR* may lead to an over-smoothed surrogate of Φ failing to "understand" the overall trend of the actual cost function as shown in Fig. 4(*b*) for the Schwefel's function [38] ($\Omega \in [-500,\ 500]$, $\Omega_{Schwefel}^{(opt)} = 420.9687$). The two surrogates perform similarly when dealing with the Ackley's function [38] [$\Omega \in [-5,\ 5]$, $\Omega_{Ackley}^{(opt)} = 0$ - Fig. 4 (*c*)].

Another widely-used *LBE* method is the Ordinary Kriging (*OK*) whose remarkable advantage

over the *RBFN* and the *SVR* is the straightforward capability of providing a measure, $\Psi\{\underline{\Omega}\}$, of the degree of *reliability/confidence* associated to any prediction $\widetilde{\Phi}\{\underline{\Omega}\}$ [37][40]. As it will be explained in the next Sections, such an additional output is a powerful source of information to be profitably exploited to enhance the effectiveness of the whole *SbD* synthesis. More in detail, the surrogate model generated by the *OK* is given by [39][40]

$$\widetilde{\Phi}\{\underline{\Omega}\} = \mu + \underline{\eta}^T \underline{\underline{\mathcal{R}}}^{-1} (\underline{\Phi} - \underline{\mathcal{I}}\mu) \tag{8}$$

where μ is a real constant, $\underline{\Phi} = \left[\Phi^{(s)};\ s = 1, ..., S\right]^T$, $.^T$ being the transpose operator, $\underline{\mathcal{I}}$ is the $(S \times 1)$ unit vector, $\underline{\underline{\mathcal{R}}} = \{\mathcal{R}_{pq};\ p,\ q = 1,\ ...,\ S\}$ is the $(S \times S)$ training correlation matrix, and $\underline{\eta} = \{\eta_s;\ s = 1, ..., S\}$ is the correlation vector of $\underline{\Omega}$ [40]. As for the prediction reliability metric, its meaning is quite intuitive since it is defined as the weighted distance between $\underline{\Omega}$ and the S training samples (i.e., $\Psi\{\underline{\Omega}\} \propto \frac{\sum_{s=1}^{S} \left\|\underline{\Omega} - \underline{\Omega}^{(s)}\right\|_2}{S}$) being $\Psi\{\underline{\Omega}\} = 0$ and $\widetilde{\Phi}\{\underline{\Omega}\} = \Phi\{\underline{\Omega}\}$ only if $\underline{\Omega} = \underline{\Omega}^{(s)}$ $(s = 1, ..., S)$ (Fig. 5). More specifically, the *OK* uncertainty is modeled starting from the assumption that the cost function value $\Phi\{\underline{\Omega}\}$ is the realization of a normally-distributed random variable with mean $\widetilde{\Phi}\{\underline{\Omega}\}$ and standard deviation equal to $\Psi\{\underline{\Omega}\}$ [39][40] (Fig. 5). For the sake of completeness, the predictions made by the *OK* for the Levy's, Schwefel's, and Ackley's functions are reported in Fig. 4.

- *SbD*-Driven Training Set Generation

In order to yield a reliable *SM,* the training set must be properly built. Hypothetically, only a very large (ideally infinite) number, S, of *I/O* pairs would allow to exactly predict the cost function value $\Phi\{\underline{\Omega}\}$ (i.e., $\widetilde{\Phi}\{\underline{\Omega}\} \rightarrow \Phi\{\underline{\Omega}\}$ when $S \rightarrow \infty$). Practically, strategies for the selection of the minimum number of *representative* samples guaranteeing the *SM* prediction error is below a desired threshold are needed. Towards this aim, the *SbD* toolkit deals with *LBE* strategies from a different perspective and in terms of a *three-step* approach where only the last one is finalized at defining the *SM* [41]. More in detail, the first step is concerned with the reduction of the dimension of the input (solution) space $\Re\{\underline{\Omega}\}$ to mitigate the negative effects of the *curse of dimensionality* [42]. It is worth pointing out that such a task is partially/preliminarily addressed by the *PF* block when choosing the smallest set of the most representative *DoF*s that

guarantees the existence of a physically-admissible solution (Sect. 3.1). However, whether K is still high (e.g., $K > 20$) lower-cardinality yet highly-informative training sets can be built by means of *space reduction* techniques. These latter determine a reduced set of K' ($K' \ll K$) *DoF*s, called *reduced features*, $\underline{\chi}^{(s)} = \left\{\chi_k^{(s)};\, k = 1, ..., K'\right\}$ by means of a linear/non-linear transformation operator Λ so that a *reduced* database, $\widehat{\mathcal{D}}_S = \left\{\left(\underline{\chi}^{(s)},\, \Phi^{(s)}\right);\, s = 1, ..., S\right\}$ (Fig. 3), is built. The Λ-based mapping can be *function-independent* (i.e., $\underline{\chi}^{(s)} = \Lambda\left\{\underline{\Omega}^{(s)}\right\}$), as in the Principal Component Analysis (*PCA*) [41] and in the Sammon Mapping (*SAM*) [23], or *function-dependent* (i.e., $\underline{\chi}^{(s)} = \Lambda\left\{\underline{\Omega}^{(s)},\, \Phi^{(s)}\right\}$) as for the Partial Least Squares (*PLS*) [41]. The second step is aimed at an "exhaustive" representation of the input space $\Re\left\{\underline{\Omega}\right\}$ by properly selecting the S *I/O* pairs. The available sampling strategies can be classified into two main categories: (*i*) one-shot/non-iterative and (*ii*) adaptive strategies [43][44][45][46][47]. The uniform grid sampling belongs to the first class (*i*) and it performs a *full-factorial* exploration of the input space by uniformly sampling each k-th ($k = 1, ..., K'$) dimension and considering all existing combinations. Clearly, it becomes rapidly *unfeasible* when increasing the solution space dimensionality K' and/or the number of quantization levels, Q (e.g., $S = Q^{K'} = 10^5$ when $K' = 5$ and $Q = 10$). To overcome such a drawback, other strategies such as the Latin Hypercube Sampling (*LHS*) [46] and the Orthogonal Arrays (*OA*s) method [16][30] can be exploited. These approaches, which belong to the class (*ii*), are based on the iterative selection of the S training samples to reach a profitable balancing between *exploration* (i.e., new samples in the regions of $\Re\left\{\underline{\chi}\right\}$ where the sampling rate is low) and *exploitation* [i.e., new samples where the cost function Φ is more nonlinear as it can be inferred from the cuts-analysis (4)] [46];

- *SbD* Time Saving

When using a *SM* instead of the *FW* solver, the total cost of the *SbD*-based synthesis turns out to be

$$\Delta t_{SbD} = \Delta t_{SM}\left(S\right) + \left(P \times I\right) \times \Delta t_{test} \tag{9}$$

where Δt_{test} is the *CPU* time to yield a single Φ-prediction and $\Delta t_{SM}\left(S\right) = \left(S \times \Delta t_{FW}\right) + \Delta t_{train}$ is the *CPU* time to perform the S training simulations and to generate the *SM* [(4)]. Ac-

[(4)]It should be noticed that also the training/test times are functions of S [i.e., $\Delta t_{train/test} = \Delta t_{train/test}\left(S\right)$]. However, S is kept quite small ($S \ll 10^3$) in practical applications to guarantee a significant time saving. There-

cordingly, the *SbD* becomes profitable and very competitive when $\Delta t_{SbD} \ll \Delta t_{StD}$ being

$$\Delta t_{StD} = (P \times I) \times \Delta t_{FW}. \tag{10}$$

To provide a simple, although rigorous, indication of the overall time saving of the *SbD* with respect to a trivial integration of a *FW* solver within an optimization loop, let us consider that in practical situations $(\Delta t_{train}, \Delta t_{test}) \ll \Delta t_{FW}$, thus the following approximation generally holds true: $\Delta t_{SbD} \approx S \times \Delta t_{FW}$. Accordingly, the time saving percentage thanks to the *SbD* (i.e., $\Delta t_{sav} \triangleq \frac{\Delta t_{StD} - \Delta t_{SbD}(S)}{\Delta t_{StD}}$) is equal to

$$\Delta t_{sav} \approx \frac{(P \times I) - S}{(P \times I)} \times 100, \tag{11}$$

thus the rule-of-thumb applicability condition for the *SbD* is $S < (P \times I)$.

The percentage saving in (11) can be even higher in case of massive parallel computing. With reference to the computational scenario where O ($O \geq P$) processors are available, while the effects of multiple and parallel computational capabilities can be exploited by a standard approach only to reduce the optimization time

$$\Delta t^{\|}_{StD} = \frac{\Delta t_{StD}}{P} \tag{12}$$

by sharing among the O *CPU*s the evaluation of the cost function of the P multiple-agents/trial-solutions at each i-th ($i = 1, ..., I$) iteration[(5)], the *SbD* benefits of this computational over-boost also for reducing the training time

$$\Delta t^{\|}_{SbD} = \frac{\Delta t_{SM}(S)}{O} + \frac{(P \times I) \times \Delta t_{test}}{P}. \tag{13}$$

Therefore, the percentage time saving is roughly around

$$\Delta t^{\|}_{sav} \approx \left(1 - \frac{S}{O \times I}\right) \times 100 \tag{14}$$

fore, the dependence on the training set size S can be neglected since $\Delta t_{train/test}(S) \ll (S \times \Delta t_{FW})$.

[(5)]Indeed, the number of *CPU*s that can be used in parallel in standard optimization algorithms is bounded by the number of trial solutions P since the generation of new solutions depends on the outcomes of the previous iteration.

[i.e., $\Delta t^{\parallel}_{sav} > 0$ if $S < (O \times I)$] and it further reduces to $\Delta t^{\parallel}_{sav} \approx \left(1 - \frac{1}{I}\right) \times 100$ [i.e., $\Delta t^{\parallel}_{sav} > 0$ always since $I > 1$ by definition of iterative optimization loop], if $O \geq S$ since $\Delta t^{\parallel}_{SM}(S) \approx \Delta t_{FW}$.

3.3 Solution Space Exploration (*SSE*)

Many global optimization methods exist and the choice of the most effective algorithm for the synthesis problem under study is a key issue and not trivial task at all. From a theoretical viewpoint, the "*no free-lunch theorem*" (*NFL*) of optimization [48] states that (*i*) "*the average performance of any pair of algorithms across all problems is identica*l" and (*ii*) "*no matter what the cost function, by simply observing how well the algorithm has done so far tells us nothing about how well it would do if we continue to use it on the same cost function*". This implies that "*whether an optimization algorithm performs better than random search on some class of problems, then it must perform worse than random search on another class*" [48]. Therefore, the application of an arbitrary algorithm to an optimization problem without understanding the nature and the properties/features of the cost function Φ as well as of $\underline{\Omega}$ is on average equivalent to perform a random search. Indeed, it is well proven that each optimizer has its own "optimal niche" of application where it outperforms other alternatives or vice-versa a suitable reformulation of the synthesis problem at hand allows one to use a particular optimization tool. By extension, the *NFL* principles hold true for the *SbD*, as well, since this latter formulates a synthesis problem as an optimization one. More specifically, the *NFL* rules apply to the SbD framework as follows "*it is not possible to a-priori identify the best combination of the functional blocks of the* SbD *able to perform well on every possible problem*".

To give some insights on this concept and its consequences, let us focus our attention to the integration of two representative blocks, namely the *SM* and the optimization tool. Towards this end, let us consider the optimization of the three benchmark functions in Fig. 4, but with $K = 6$, performed with the "bare" integration of the *RBFN*, the *SVR*, and the *OK* models with two state-of-the-art evolutionary optimizers, namely the Particle Swarm Optimizer (*PSO*) and the Differential Evolution (*DE*) [13][14]. Given the stochastic nature of both the *SM*s (i.e., the *LHS*) and the optimizers, the median realization over $R = 20$ executions with $P = 10$ agents

for $I = 200$ iterations is reported to yield statistically-meaningful results. Figure 6 reports the cost function values at the last iteration ($i = I$) of the different *BARE-SbD* algorithms versus the $\frac{S}{K}$ ratio. Dealing with the Levy's function [Fig. 6(a)], the choice of the *SVR* to generate the *SM* turns out to be successful for the arising *SbD* implementation since both the *PSO-SVR* and the *DE-SVR* integrations yield the best results provided that a sufficient number of training samples is available (i.e., $\frac{S}{K} \geq 20 \rightarrow S \geq 120$ being $K = 6$). Otherwise, the *SVR*-based methods are the worse ones for the minimization of the Schwefel's function [Fig. 6(b)] because of the *over-smoothing* in approximating the actual cost function [Fig. 4(b)]. On the contrary, the *OK* enables a proper exploration of the solution space when integrated with the *PSO* [Fig. 6(b)]. This latter choice is sub-optimal when dealing with the Ackley's function, while both *SVR*-based strategies perform very well whatever the cardinality S of the training set [Fig. 6(c)].

4 Advanced *SbD* Strategies

Totally replacing the *FW* solver with a *SM*, as done in *BARE-SbD* approaches (Sect.3.3), may lead to sub-optimal designs [49]. This holds true especially when non-negligible time savings are required to permit the synthesis of high-complexity and computationally demanding devices/systems, thus setting a low size, S, for the training set because of the very limited number of *affordable* simulations within a reasonable amount of time. As a matter of fact, treating the *SM* as a "magic black box" could produce undesired effects such as (i) the convergence towards false solutions/local minima and/or (ii) the prediction of *unfeasible* cost function values (e.g., negative values, $\widetilde{\Phi}\left\{\underline{\Omega}\right\} < 0$). Moreover, simply increasing S to *globally* improve the prediction accuracy could be not enough to prevent such issues and advanced *SbD* strategies are necessary. They are mainly based on the following basic recipes: (i) the *local refinement* of the *SM* within the *attraction basin* of $\underline{\Omega}^{(opt)}$ and/or (ii) the *interactive collaboration* between the optimizer and the predictor. By following those guidelines, two advanced *SbD* implementations are described in the following.

4.1 Optimization-Driven "Smart" *SM* Generation

Unlike state-of-the-art adaptive sampling strategies (Sect. 3.2), the accuracy of the *SM* is *locally* enhanced in the neighborhood of $\underline{\Omega}^{(opt)}$. Towards this end, an *optimization-driven* adaptive sampling strategy can be adopted to build the training set $\mathcal{D}_S$ by iteratively adding new samples to an initial dataset $\mathcal{D}_0$ with S_0 ($S_0 < S$) *I/O* pairs. Such a strategy belongs to the class of the "output space filling" (*OSF*) techniques [50] and it is aimed at uniformly exploring the output/cost function space for which $\Phi\left\{\underline{\Omega}\right\} < \Phi_{th}$, Φ_{th} being a user-specified threshold. It is performed within the *CFC* block (Fig. 1) and it consists of the following procedural steps:

1. *Initialization* - Generate $\mathcal{D}_0 = \left[\left(\underline{\Omega}^{(s)};\, \Phi^{(s)}\right);\, s = 1, ..., S_0\right]$ by sampling the input space via *LHS* and initialize the loop index ($j = 1$);

2. *SbD-OSF Loop* [$j = 1, ..., (S - S_0)$]

   (a) Train a *SM* using the S_{j-1} samples of the dataset $\mathcal{D}_{j-1}$;

   (b) Sample the input space via *LHS* to generate C *candidates*, $\{\underline{\Omega}^{(c)};\, c = 1, ..., C\}$[6], and predict the corresponding cost function values, $\{\widetilde{\Phi}^{(c)} \triangleq \widetilde{\Phi}\left\{\underline{\Omega}^{(c)}\right\};\, c = 1, ..., C\}$;

   (c) Select the "best" candidate $\underline{\Omega}^*$ as $\underline{\Omega}^* = \arg\left\{\max_{c=1,...,C}\left[\min_{s=1,...,S_{j-1}}\left(d_{c,\,s}\right)\right]\right\}$ subject to $\widetilde{\Phi}^{(c)} \leq \Phi_{th}$ being $d_{c,\,s} \triangleq \left|\widetilde{\Phi}^{(c)} - \Phi^{(s)}\right|$;

   (d) Compute the actual cost function value of $\underline{\Omega}^*$, $\Phi^* \triangleq \Phi\left\{\underline{\Omega}^*\right\}$, and update the training size, $S_j \leftarrow (S_{j-1} + 1)$, and dataset, $\mathcal{D}_j \leftarrow \mathcal{D}_{j-1} \bigcup (\underline{\Omega}^*,\, \Phi^*)$ along with the loop index, $j \leftarrow (j + 1)$. Stop the procedure if $S_j = S$ or go to Step 2(a).

Figure 7 illustrates the results of such a procedure when applied to the 1-*D* Ackley's function. More specifically, the *SbD-OSF* has been run by setting $S_0 = 5$ and $S = 50$, while the *OK*-based *SM* model has been chosen to predict $C = 200$ candidates at each j-th [$j = 1, ..., (S - S_0)$] loop. As it can be observed, there is an adaptive exploration of the region where $\Phi\left\{\underline{\Omega}\right\} < \Phi_{th}$ [$\Phi_{th} = 6.0$ - Fig. 7(*a*)] and, as a by-product, the accuracy of the *OK* has been enhanced only close to $\underline{\Omega}^{(opt)}$ [Fig. 7(*b*)].

[6] According to the reference literature [47], the number of candidates is set to a value in the range $C \in [50,\, 200] \times K$.

4.2 "Confidence-Enhanced" *SbD* Optimization

Another strategy to improve the effectiveness of the *SbD* is to implement a more *interactive* framework by enforcing a *bilateral* exchange of information between the *CFC* and the *SSE* blocks (Fig. 1). Let us first notice that, on the one hand, the *SM* is a computationally-cheap alternative to the *FW* solver, on the other hand, the optimizer progressively localizes "*promising*" regions of the search space (i.e., the attraction basins of the landscape of the cost function) where the prediction accuracy should be enhanced to facilitate the convergence towards the global minimum, $\underline{\Omega}^{(opt)}$. In order to fully exploit these features, a novel *SbD* strategy is proposed hereinafter by profitably combining the global search/hill-climbing features of the *PSO* [13] with the capability of the *OK* to provide a *reliability* index (Sect. 3.2) of the *SM* predictions. The arising "confidence-enhanced" *PSO-OK* (*PSO-OK/C*) method is based on a "reinforced learning" (*RL*) strategy that updates the *SM* during the optimization by adaptively selecting trial solutions to be evaluated with the *FW* solver. It works as follows:

1. *Initialization* ($i = 0$) - Train an *OK*-based *SM* starting from an initial training set of S_0 *I/O* pairs, $\mathcal{D}_{S_0}$. Compute the best cost function value of the solutions in $\mathcal{D}_{S_0}$,$\Phi_{best}^{train} = \min_{s=1,...,S_0}\left[\Phi^{(s)}\right]$. Given the desired time saving Δt_{sav}, set the maximum number of *affordable* simulations, S, accordingly [i.e., $S = P \times I \times \left(1 - \frac{\Delta t_{sav}}{100}\right)$]. Define an initial swarm of P particles with random positions $\{\underline{\Omega}_0^{(p)};\ p = 1, ..., P\}$ and velocities $\{\underline{v}_0^{(p)};\ p = 1, ..., P\}$;

2. *PSO-OK/C Optimization Loop* ($i = 0, ..., I$)

   (a) For each p-th ($p = 1, ..., P$) particle, predict the cost function value, $\widetilde{\Phi}_i^{(p)}$ ($\widetilde{\Phi}_i^{(p)} \triangleq \widetilde{\Phi}\left\{\underline{\Omega}_i^{(p)}\right\}$), and compute the associated *confidence* index, $\Psi_i^{(p)}$ ($\Psi_i^{(p)} \triangleq \Psi\left\{\underline{\Omega}_i^{(p)}\right\}$). If $S_i < S$ then go to Step 2(b), otherwise go to Step 2(c);

   (b) Select the "most promising" particle as $\underline{\Omega}_i^* = \arg\left[\min_{p=1,...,P}\left(\mathcal{F}^-\left\{\underline{\Omega}_i^{(p)}\right\}\right)\right]$, where $\mathcal{F}^-\left\{\underline{\Omega}_i^{(p)}\right\} = \widetilde{\Phi}_i^{(p)} - \zeta\Psi_i^{(p)}$ is the "lower confidence bound" (*LCB*) of the p-th ($p = 1, ..., P$) trial solution, $1 \leq \zeta \leq 3$ being a real coefficient [23]. If $\mathcal{F}^-\left\{\underline{\Omega}_i^{(p)}\right\} > \Phi_{best}^{train}$ then jump to Step 2(c), otherwise perform the following *RL* operations:

i. Use the *FW* solver to compute the actual cost function of $\underline{\Omega}_i^*$, Φ_i^* ($\Phi_i^* \triangleq \Phi\left\{\underline{\Omega}_i^*\right\}$);

ii. Update the training set, $\mathcal{D}_{S_i} \leftarrow \mathcal{D}_{S_{i-1}} \bigcup (\underline{\Omega}^*,\ \Phi_i^*)$, and its size, $S_i \leftarrow (S_{i-1}+1)$. If $\Phi_i^* < \Phi_{best}^{train}$, then update Φ_{best}^{train} ($\Phi_{best}^{train} \leftarrow \Phi_i^*$);

iii. Re-train the *OK* model with $\mathcal{D}_{S_i}$;

(c) Update the personal best position of each p-th ($p = 1, ..., P$) particle, $\underline{b}_i^{(p)}$ ($\underline{b}_i^{(p)} = \arg\left\{\min_{j=1,...,i}\left[\Phi_j^{(p)}\right]\right\}$), according to the rules summarized in Tab. I and sketched in Fig. 8. Such an updating process is based on the degree of *reliability* of each p-th ($p = 1, ..., P$) trial solution, $\Psi_i^{(p)}$, and of its previous best position, $\Psi\left\{\underline{b}_{i-1}^{(p)}\right\}$. For instance, let us consider the case illustrated in Fig. 8(*d*) where the cost function value assigned to $\underline{\Omega}_i^{(p)}$, $\widetilde{\Phi}_i^{(p)}$, and $\underline{b}_{i-1}^{(p)}$, $\widetilde{\Phi}\left\{\underline{b}_{i-1}^{(p)}\right\}$, is affected by some uncertainty, both being predicted. Although $\widetilde{\Phi}_i^{(p)} > \widetilde{\Phi}\left\{\underline{b}_{i-1}^{(p)}\right\}$, it is profitable to update the p-th ($p = 1, ..., P$) personal best (i.e., $\underline{b}_i^{(p)} \leftarrow \underline{\Omega}_i^{(p)}$) since $\underline{\Omega}_i^{(p)}$ has a higher *probability* to have a smaller cost function value than $\underline{b}_{i-1}^{(p)}$ [i.e., $\mathcal{F}^-\left\{\underline{\Omega}_i^{(p)}\right\} < \mathcal{F}^-\left\{\underline{b}_{i-1}^{(p)}\right\}$ - Tab I and Fig. 8(*d*)]. Otherwise, whether the previous personal best has been simulated [i.e., it is 100% reliable - Fig. 8(*b*)], it can be updated with the current particle position only if this latter has no chance to have a worse/higher cost function value (i.e., $\mathcal{F}^+\left\{\underline{\Omega}_i^{(p)}\right\} < \Phi\left\{\underline{b}_{i-1}^{(p)}\right\}$, $\mathcal{F}^+\left\{\underline{\Omega}_i^{(p)}\right\}$ being the "upper confidence level" (*UCB*) associated to $\underline{\Omega}_i^{(p)}$ given by $\mathcal{F}^+\left\{\underline{\Omega}_i^{(p)}\right\} \triangleq \widetilde{\Phi}_i^{(p)} + \zeta\Psi_i^{(p)}$ [Tab. I - Fig. 8(*b*)];

(d) Update the global best position, $\underline{g}_i$ ($\underline{g}_i \equiv \underline{\Omega}_i^{best}$) according to the *confidence-based* rules in Tab. II and illustrated in Fig. 9.

(e) If $i = I$, then stop the optimization and output $\underline{g}_I$ as the final design (i.e., $\underline{\Omega}^{(end)} = \underline{g}_I$), else go to Step 2(*f*);

(f) Use the standard *PSO* governing equations [13] to generate a new set of particles positions and velocities, update the iteration index, $i \leftarrow (i+1)$, and go to Step 2(a).

As it can be inferred, the *PSO-OK/C SbD* method implements a surrogate-assisted evolutionary optimization by exploiting an "individual-based" *model management* strategy [49], but unlike state-of-the-art techniques, it gives the user a full control of the time saving Δt_{sav} by letting him specify the total amount S of feasible *FW* simulations [i.e., $S = P \times I \times \left(1 - \frac{\Delta t_{sav}}{100}\right)$] in order

to comply with specific computational/time constraints for the synthesis problem at hand.

5 *SbD* as Applied to the Synthesis of Complex *EM* Systems

This Section is aimed at assessing the effectiveness, the potentialities, and the current limitations of the *SbD* as applied to the design of complex *EM* systems. Towards this end, two representative benchmarks, concerned with the synthesis of (*i*) time modulated arrays (*TMA*s) (Sect. 5.1) and of (*ii*) microstrip patch arrays for 5*G* applications (Sect. 5.2), will be discussed.

5.1 *Benchmark 1* - Synthesis of *TMA*s

The first benchmark problem deals with the synthesis of *TMA*s comprising real radiators and non-linear switching beam-forming networks (*BFN*s). To model the mutual coupling effects among the N antennas as well as the complex nonlinear/dynamic behavior of the *BFN*, the Harmonic-Balance (*HB*) technique [51][52] has been used. The goal is to determine the optimal setup of the switch-on instants, $\underline{t}^{on} = \{t_n^{on};\, n = 1, ..., N\}$, t_n^{on} being the T-normalized rise time of the n-th ($n = 1, ..., N$) element subject to the physical-admissibility constraint $0 \leq t_n^{on} < 1$ (*RCD* Block), that minimizes the fluctuations of the instantaneous directivity D within the modulation period T [51]. Following the guidelines in Sect. 3.1 (*PF* Block), the cardinality of the solution space (i.e., the number of *DoF*s) has been reduced to $K = \frac{N}{2}$ by considering symmetric excitation sequences [i.e., $t_n^{on} = t_{(N-n+1)}^{on};\, n = \left(\frac{N}{2} + 1\right), ..., N$] so that $\underline{\Omega} = \left\{t_n^{on};\, n = 1, ..., \frac{N}{2}\right\}$, while the cost function has been defined as

$$\Phi\left\{\underline{\Omega}\right\} = \frac{1}{\overline{D}\left\{\underline{\Omega}\right\} T} \int_0^T \left|\overline{D}\left\{\underline{\Omega}\right\} - D\left\{\underline{\Omega},\, t_p\right\}\right| dt_p \tag{15}$$

$\overline{D}$ being the average value of the instantaneous directivity over T.

As a representative numerical test, a *TMA* with $N = 16$ monopoles resonating at $f_0 = 2.45$ [GHz] printed over a *RF60-A* Taconic substrate with relative permittivity $\varepsilon_r = 6.15$, loss tangent $\tan\delta = 0.0028$, and thickness $h = 0.635$ [mm] [51] has been considered. Moreover, the duration of the modulation period has been set to $T = 10$ [μsec], while the durations of the driving pulses, $\underline{\tau} = \{\tau_n;\, n = 1,\, ...,\, N\}$, have been chosen to afford a Dolph-Chebyshev pattern

with a side-lobe level of $SLL = -30$ [dB] at f_0.

In order to define the most suitable implementation of the *CFC* and *SSE* blocks, a preliminary study has been carried out to (*i*) estimate the nature of the cost function (15) as well as to (*ii*) assess the accuracy of different *SM*s. More specifically, the plots of the 1-*D* cuts of Φ (4) have been evaluated and an example is reported in Fig. 10(*a*). In this latter, the trial solution $\underline{\Omega} = \underline{\Omega}^{(1)}$ $(t = 0)$ corresponds to a randomly-chosen position within the search space $\Re\{\underline{\Omega}\}$, while $\underline{\Omega} = \underline{\Omega}^{(2)}$ $(t = 1)$ is the solution found by a state-of-the-art *StD* approach based on a *PSO* run of $I = 200$ iterations with $P = 10$ particles [51] [$\Phi_{StD} = 3.2 \times 10^{-2}$ ($\Phi_{StD} \triangleq \Phi\left\{\underline{\Omega}_{StD}^{(end)}\right\}$) - Fig. 10(*a*)]. As it can be observed, the cost function (15) has a highly-oscillating multi-modal behavior with the occurrence of many local minima and the presence of steep/non-symmetric valleys [Fig. 10(*a*)]. Those features imply that the *SSE* block cannot be a deterministic optimizer, but a stochastic hill-climbing techniques is mandatory.

Next, the accuracy of different *SM*s (namely, the *RBFN*, the *SVR*, and the *OK* models) has been preliminary analyzed by checking the dependence of the prediction accuracy on the number of training samples, S. Towards this aim, a set of $U = (P \times I) = 2000$ *HB-FW* simulations has been performed by *LHS*-sampling the solution space, then the *SM*s have been trained with a subset of $S < U$ samples and their accuracy has been evaluated on the remaining $M = (U - S)$ samples. Figure 10(*b*) reports the values of the *SM* error

$$\Xi(S) = \frac{\sum_{m=1}^{M=(U-S)} \left|\widetilde{\Phi}\left\{\underline{\Omega}^{(m)} \middle| S\right\} - \Phi\left\{\underline{\Omega}^{(m)}\right\}\right|^2}{\sum_{m=1}^{M=(U-S)} \left|\Phi\left\{\underline{\Omega}^{(m)}\right\}\right|^2} \tag{16}$$

versus S along with the time saving Δt_{sav} of the *SbD* [Fig. 10(*b*)]. As expected, Δt_{sav} (linearly) decreases with S independently on the adopted *SM*, while the *OK*-based surrogate always outperforms the other *SM*s in terms of prediction performance, the *SVR* generally providing the worse values of Ξ [Fig. 10(*b*)]. It is also worth pointing out that, no matter what the training set size within the range $0.1 \leq \frac{S}{U} \leq 0.75$, the value of Ξ of each method is almost constant and smaller than 10^{-1}.

Moving to the solution of the *TMA* design problem with standard *SbD* approaches using *LHS* training sets of different sizes, let us analyze the case of very high time saving (i.e., $\Delta t_{sav} \geq$

$90\% \rightarrow S \leq 200$). By computing the "design quality index", $\Delta\Phi \triangleq \frac{(\Phi_{SbD}-\Phi_{StD})}{\Phi_{StD}}$, as the normalized difference between the cost function of the *StD* solution, Φ_{StD}, and that from the *SbD*, $\Phi_{SbD} \triangleq \Phi\left\{\underline{\Omega}_{SbD}^{(end)}\right\}$, it turns out that both the *DE-OK* and the *PSO-OK* always yield the best results [Fig. 10(*c*)]. The performance of the *advanced SbD* approaches presented in Sect. 4 have been assessed, as well. They allows one remarkable improvements with respect to all bare *SbD* implementations. Indeed, Figure 11 indicates that the "smart" sampling performed by the *SbD-OSF* technique (here initialized with $S_0 = 40$ samples) results in a significant reduction of $\Delta\Phi$ when applying, later on, the *PSO-OK*. Even more significant is the positive effect of the *PSO-OK/C* (still initialized with $S_0 = 40$ samples). For instance, the *PSO-OK/C* obtains a solution very close to that of the *StD* approach when $S = 100$ (i.e., $\Delta\Phi|_{PSO-OK/C}^{S=100} = 6.2\%$ - Fig. 11), but with an impressive time saving of $\Delta t_{sav} = 95\%$. In order to better appreciate such an outcome, let us consider that $\Delta t_{FW} = 13$ [sec] on a desktop PC with 3.6 GHz CPU and 32 GB of RAM memory, thus the time required by the *StD* approach and the *PSO-OK/C* to complete the optimization ($i = I$) was $\Delta t_{StD} = 7.2$ [hours] and $\Delta t|_{PSO-OK/C}^{S=100} = 21.2$ [min], respectively.

The effectiveness of the *PSO-OK/C* in exploring the solution space are confirmed by the evolution of the optimal value of the cost function during the iterative optimization process ($i = 1, ..., I$) [Φ_i^{best} - Fig. 12(*a*)] as well as by the behavior of the optimized instantaneous directivities [Fig. 12(*b*)], which appears very similar to that of the *StD* approach in both plots. It is also interesting to note the progressive refinement of the *OK* surrogate during the *PSO-OK/C* optimization as it can be inferred from the plot of the functional cut passing through the global best solutions at $i = 0$ (i.e., $\underline{g}_{i=0}$ - $t = 0$) and $i = I$ ($\underline{g}_{i=I}$ - $t = 1$) (Fig. 13). As a matter of fact, there is a non-negligible improvement of the *SM* accuracy in mapping the actual behavior of the cost function from the initial ($i = 0$) to the final iteration ($i = I$) thanks to the adaptive selection, during the optimization, of new training samples (Fig. 13).

As for the improvement of the *PSO-OK/C* over the *PSO-OK*, Figure 14 compares the predicted values of the cost function, $\widetilde{\Phi}\left\{\underline{g}_j\right\}$, with the corresponding actual ones, $\Phi\left\{\underline{g}_j\right\}$, at some intermediate "control-points" ($j = 1, ..., J$; J being the maximum number of selected iterations). Unlike the *PSO-OK* (although the same number of $S = 100$ *FW* simulations has been used),

there is a perfect matching between the *PSO-OK/C* predictions and the actual cost function values thanks to the adaptive addition of training samples during the minimization. Quantitatively, it turns out that $\Phi_{SbD\,(PSO-OK)} = 9.4 \times 10^{-2}$ vs. $\Phi_{SbD\,(PSO-OK/C)} = 3.4 \times 10^{-2}$ (Fig. 14). Finally, the *SbD* has been compared with competitive state-of-the-art approaches. First, the surrogate-assisted *DE* (*SADE*) algorithm [23] has been taken into account and Figure 11 proves that the *PSO-OK/C* positively compares with such an optimization approach always providing good trade-offs between design-quality and time saving. Moreover, a comparison between the *PSO-OK/C* performance and those from a space mapping technique [25] has been carried out, as well. Towards this end, the additive-input/multiplicative-output (*AIMO*) space mapping approach [26] has been implemented by considering an analytic model for the *TMA* composed by isotropic radiators and ideal switches ($\Delta t_{coarse} = 0.02$ [sec]). It turns out that, despite a lightly better time saving with just $S = 6$ *HB-FW* simulations and $S_{coarse} = 2200$ coarse-model evaluations (i.e., $\Delta t_{sav}|_{AIMO} = 97.9\%$), the final solution is slightly less accurate than the *PSO-OK/C* one (i.e., $\Phi_{AIMO} = 5.9 \times 10^{-2}$).

5.2 *Benchmark 2* - Design of a Microstrip Array for 5*G* Applications

This second test case is devoted to further assess the advantages of the *SbD* when dealing with *high-complexity EM* design problems. Let us consider the synthesis of a planar microstrip array for 5*G* applications working at $f_0 = 3.5$ [GHz] (Fig. 15). The antenna has been supposed to lie on the $(y,\ z)$-plane and composed by $N = (4 \times 6)$ cavity-backed slot-fed square patches of side $L_p = 2.73 \times 10^{-1}\ \lambda$. To enable ± 45 [deg] dual-polarization operation ($J = 2$ being the number of operating modes), each element has been rotated by 45 [deg] and fed by two rectangular slots of dimensions $(W_h,\ L_h) = (2.65 \times 10^{-2},\ 1.31 \times 10^{-1})\ \lambda$ [Fig. 15(*b*)]. The dimensions of the feeding lines have been set to $(W_f,\ L_f) = (1.69 \times 10^{-2},\ 1.17 \times 10^{-1})\ \lambda$, with an additional stub of length $L_s = 5.25 \times 10^{-2}\ \lambda$ to reach a proper impedance matching. The following material and dielectric properties have been assumed: Arlon DiClad527 ($\varepsilon_r = 2.5$, $\tan\delta = 0.0018$) with a total thickness of $H = (h_1 + h_2 + h_c) = 1.24 \times 10^{-1}\ \lambda$ [Fig. 15(*a*)], $h_1 = 1.77 \times 10^{-2}\ \lambda$, $h_2 = 5.93 \times 10^{-3}\ \lambda$, and $h_c = 1.0 \times 10^{-1}\ \lambda$ being the height of the patch substrate, of the feeding substrate, and of the cavity, respectively. Starting from this reference setup, the set of

*DoF*s $\underline{\Omega} = \{d_y,\, d_z,\, l_1,\, l_2\}$ ($K = 4$), d_y and d_z being the inter-element spacing along y and z, respectively, while l_1 and l_2 locate the position of the feeding slot and microstrip line associated to polarization/port $j = 2$ [Fig. 15(*b*)], have been optimized to maximize the realized gain along the main beam direction $(\theta_o,\, \varphi_o)$, $G_{j,o}\left(\theta_o,\, \varphi_o\right)$, for both polarizations and also towards $O = 5$ steering angles ($o = 1, ..., O$) within the visible cone of the array [(7)]. Mathematically, such a synthesis problem has been coded (*PF* Block) into the following cost function

$$\Phi\left\{\underline{\Omega}\right\} = \left(\frac{1}{2 \times O} \sum_{j=1}^{J} \sum_{o=1}^{O} G_{j,o}\left(\theta_o,\, \varphi_o \middle| \underline{\Omega}_{5G}\right)\right)^{-2} \tag{17}$$

As for the computation of the actual value of the cost function Φ, the *FW* problem at hand has been solved with the Ansys *HFSS EM* simulator [53] by considering exactly the *finite* structure of the antenna (i.e., no periodic-infinite hypotheses) to take into account both the mutual coupling effects among the array elements as well as the fringing effects, the average simulation time being equal to $\Delta t_{FW} = 5.64 \times 10^3$ [sec].

Figure 16 shows the evolution of the optimal value of the cost function for a *StD* optimization based on the *PSO* executed with $P = 4$ agents for $I = 50$ iterations ($P \times I = 200 \rightarrow \Delta t_{StD} = 1.13 \times 10^6$ [sec] ~ 13 [days]), $\Phi\left\{\underline{g}_i\right\}$ ($i = 1, ..., I$), as well as the predicted curve outputted by the *PSO-OK/C*, Φ_i, along with the intermediate "control points". The *SbD* converges to a solution whose cost function value is only $\Delta\Phi = 2.64\%$ greater than that of the *StD*, but with a remarkable advantage in terms of computational efficiency. As a matter of fact, the *PSO-OK/C* performed only $S = 40$ *FW* simulations (comprising $S_0 = 20$ initial training samples before the optimization) enabling a time saving of $\Delta t_{sav} = 80\%$. For completeness, the plot of the arising realized gain patterns are shown in Fig. 17 when steering the main beam towards $(\theta_o,\, \varphi_o) = (90,\, 0)$ [deg] [Fig. 17(*a*)] and $(\theta_o,\, \varphi_o) = (105,\, 60)$ [deg] [Fig. 17(*b*)]. For completeness, the average realized gains along the steering direction are very close [i.e., $\overline{D}_{StD}\big\rfloor_{j=1} = 15.43$ [dB] vs. $\overline{D}_{SbD}\big\rfloor_{j=1} = 15.22$ [dB] and $\overline{D}_{StD}\big\rfloor_{j=2} = 15.53$ [dB] vs. $\overline{D}_{SbD}\big\rfloor_{j=2} = 15.36$ [dB], being $\overline{D}_j \triangleq \frac{1}{2\times O} \sum_{o=1}^{O} G_{j,o}\left(\theta_o,\, \varphi_o \middle| \underline{\Omega}\right)$], further confirming the similarity of the solutions at the convergence.

[(7)]More in detail, the following steering directions have been considered: $(\theta_1,\, \varphi_1) = (90,\, 0)$ [deg], $(\theta_2,\, \varphi_2) = (-60,\, 75)$ [deg], $(\theta_3,\, \varphi_3) = (60,\, 75)$ [deg], $(\theta_4,\, \varphi_4) = (-60,\, 105)$ [deg], and $(\theta_5,\, \varphi_5) = (60,\, 105)$ [deg].

6 Conclusions, Final Remarks, and Future Trends

The *SbD* is an innovative paradigm for the computationally-efficient solution of complex *EM* synthesis problems mainly devoted to properly deal with the "high complexity" curse. Towards this purpose, the synthesis problems at hand are addressed through a suitable problem-driven selection and interconnection of *functional blocks*, each one implementing a rather "simple" and well-identified task, but all integrated to jointly fit, in an easier fashion, the required system functionality/performance. From a theoretical point of view, after summarizing the key-features and the building blocks of the *SbD*-based synthesis framework, two innovative *SbD* implementations have been presented and applied to two challenging problems concerned with the design of realistic *TMA*s (Sect. 5.1) and the synthesis of planar dual-polarization microstrip arrays for 5*G* applications (Sect. 5.2).

The main outcomes from these analyses are as follows:

- the *SbD* enables the design of complex *EM* devices/systems in a suitable time-frame thanks to the possibility to select the best (i.e., problem-oriented) trade-off between time saving and prediction accuracy;
- the accurate selection of the solution descriptors (i.e., the *DoF*s) allows one to reduce the number of required training samples and therefore the computational burden of the training phase;
- the role of the *SM*s in the *SbD* is not only that of a reliable (i.e., accurate prediction of the cost function values) and computationally-cheap alternative to the *FW* solver, but they mainly devoted to map the landscape of the cost function to reliably drive the search for the global optimum of the *SSE* algorithm.

Future activities, out of the scope of this work, will be aimed at customizing and applying the *SbD* to further challenging and high-complexity designs including unconventional phased arrays with real elements [54], and innovative meta-materials for smart *EM* environments applications [55]. Moreover, the development of *SbD* strategies, which exploit deep learning [56] and/or multi-objective optimization algorithms [57], is under investigation. Finally, innovative

optimization-driven methodologies based on the Compressive Sensing (*CS*) paradigm will be studied to build optimal training sets overcoming the Nyquist's theoretical limit so that it would possible to faithfully predict the cost function values using far fewer samples than traditional approaches [58].

Acknowledgements

This work has been partially supported by the Italian Ministry of Education, University, and Research within the Program "Smart cities and communities and Social Innovation" (CUP: E44G14000060008) for the Project "WATERTECH - Smart Community per lo Sviluppo e l'Applicazione di Tecnologie di Monitoraggio Innovative per le Reti di Distribuzione Idrica negli usi idropotabili ed agricoli" (Grant no. SCN_00489) and by the Ministry of Education of China within the Chang-Jiang Visiting Professor chair. A. Massa wishes to thank E. Vico for her never-ending inspiration, support, guidance, and help.

References

[1] J.-M. Jin, *The Finite Element Method in Electromagnetics*, Wiley-IEEE Press, 2014.

[2] R. F. Harrington, *Field Computation by Moment Methods.* Wiley-IEEE Press, 1993.

[3] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, 2005.

[4] S. V. Savic, M. M. Ilic, and B. M. Kolundzija, "Maximally orthogonalized higher order basis functions in large-domain finite element modeling in electromagnetics," *IEEE Trans. Antennas Propag.*, in press (DOI: 10.1109/TAP.2020.2970038).

[5] F. Verni, M. Righero, and G. Vecchi, "On the use of entire-domain basis functions and fast factorizations for the design of modulated metasurface," *IEEE Trans. Antennas Propag.*, vol. 68, no. 5, pp. 3824-3833, May 2020.

[6] T. Wan, B. Tang, and M. Li, “An iteration-free domain decomposition method for the fast finite element analysis of electromagnetic problems,” *IEEE Trans. Antennas Propag.*, vol. 68, no. 1, pp. 400-410, Jan. 2020.

[7] X. Jia, F. Yang, X. Liu, M. Li, and S. Xu, “Fast nonuniform metasurface analysis in FDTD using surface susceptibility model,” *IEEE Trans. Antennas Propag.*, in press (DOI: 10.1109/TAP.2019.2957317).

[8] D. Davidson, *Computational electromagnetics for RF and microwave engineering*. Cambridge: Cambridge University Press, 2011.

[9] J. H. Kim, H. J. Chun, I. C. Hong, Y. J. Kim, and Y. B. Park, “Analysis of FSS radomes based on physical optics method and ray tracing technique,” *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 868-871, 2014.

[10] D. R. Jackson and N. G. Alexopoulos, “Simple approximate formulas for input resistance, bandwidth, and efficiency of a resonant rectangular patch,” *IEEE Trans. Antennas Propag.*, vol. 39, no. 3, pp. 407-410, Mar. 1991.

[11] S. K. Goudos, T. V. Yioultsis, A. D. Boursianis, K. E. Psannis, and K. Siakavara, “Application of new hybrid jaya grey wolf optimizer to antenna design for 5G communications systems,” *IEEE Access*, vol. 7, pp. 71061-71071, 2019.

[12] S. K. Goudos, K. Siakavara, T. Samaras, E. E. Vafiadis, and J. N. Sahalos, "Self-adaptive differential evolution applied to real-valued antenna and microwave design problems,” *IEEE Trans. Antennas Propag.*, vol. 59, no. 4, pp. 1286-1298, Apr. 2011.

[13] P. Rocca, M. Benedetti, M. Donelli, D. Franceschini, and A. Massa, “Evolutionary optimization as applied to inverse scattering problems,” *Inv. Prob.*, Invited Paper, vol. 24, pp. 1-41, 2009.

[14] 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.

[15] Z. Bayraktar, M. Komurcu, J. A. Bossard, and D. H. Werner, "The wind driven optimization technique and its application in electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 61, no. 5, pp. 2745-2757, May 2013.

[16] Z. Bayraktar, D. H. Werner, and P. L. Werner, "Miniature Meander-line dipole antenna arrays, designed via an Orthogonal-Array-initialized hybrid Particle-Swarm Optimizer," *IEEE Antennas Propag. Mag.*, vol. 53, no. 3, pp. 42-59, Jun. 2011.

[17] G. Oliveri and A. Massa, "GA-Enhanced ADS-based approach for array thinning," *IET Microw. Antennas Propag.*,vol. 5, no. 3, pp. 305-315, 2011.

[18] 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 Trans. Antennas Propag.*, vol. 66, no. 8, pp. 4292-4297, Aug. 2018.

[19] L. Lizzi, F. Viani, R. Azaro, and A. Massa, "A PSO-driven spline-based shaping approach for ultra-wideband (UWB) antenna synthesis," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2613-2621, Aug. 2008.

[20] H. Aliakbari, A. Abdipour, A. Costanzo, D. Masotti, R. Mirzavand, and P. Mousavi, "ANN-based design of a versatile millimetre-wave slotted patch multi-antenna configuration for 5G scenarios," *IET Microw. Antennas Propag.*, vol. 11, no. 9, pp. 1288-1295, Jul. 2017.

[21] S. Koziel and A. Bekasiewicz, "Fast surrogate-assisted frequency scaling of planar antennas with circular polarisation," *IET Microw. Antennas Propag.*, vol. 13, no. 5, pp. 602-607, Apr. 2019.

[22] J. A. Easum, J. Nagar, P. L. Werner, and D. H. Werner, "Efficient multiobjective antenna optimization with tolerance analysis through the use of surrogate models," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 6706-6715, Dec. 2018.

[23] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180-192, Apr. 2014.

[24] D. R. Prado, J. A. Lopez-Fernandez, M. Arrebola, and G. Goussetis, “Support vector regression to accelerate design and crosspolar optimization of shaped-beam reflectarray antennas for space applications,” *IEEE Trans. Antennas Propag.*, vol. 67, no. 3, pp. 1659-1668, Mar. 2019.

[25] J. W. Bandler et *al.*, "Space mapping: the state of the art,” *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 337-361, Jan. 2004.

[26] S. Koziel and J. W. Bandler, “Distributed fine model evaluation for rapid space-mapping optimisation of microwave structures,” *IET Microw. Antennas Propag.*, vol. 3, no. 5, pp. 798-807, Aug. 2009.

[27] M. Salucci, L. Tenuti, G. Gottardi, A. Hannan, and A. Massa, ”System-by-design method for efficient linear array miniaturisation through low- complexity isotropic lenses,” *Electron. Lett.*, vol. 55, no. 8, pp. 433-434, May 2019.

[28] G. Oliveri, L. Tenuti, E. Bekele, M. Carlin, and A. Massa, “An SbD-QCTO approach to the synthesis of isotropic metamaterial lenses,” *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 1783-1786, 2014.

[29] J. Nagar, S. D. Campbell, Q. Ren, J. A. Easum, R. P. Jenkins, and D. H. Werner, “Multi-objective optimization-aided metamaterials-by-design with application to highly directive nanodevices,” *IEEE J. Multiscale Multiphys. Comput. Tech.*, vol. 2, pp. 147-158, 2017.

[30] M. Salucci, N. Anselmi, S. Goudos, and A. Massa, “Fast design of multiband fractal antennas through a system-by-design approach for NB-IoT applications,” *EURASIP J. Wirel. Comm.,* vol. 2019, no. 1, pp. 1-15, Mar. 2019.

[31] A. Massa and M. Salucci, “Dealing with complexity in electromagnetics through the system-by-design paradigm - New strategies and applications to the design of airborne radomes,” *Proc. 2018 IEEE AP-S International Symposium and USNC-URSI Radio Science Meeting*, Boston, Massachussets, USA, pp. 529-530, Jul. 8-13, 2018.

[32] G. Oliveri, F. Viani, N. Anselmi, and A. Massa, “Synthesis of multi-layer WAIM coatings for planar phased arrays within the system-by-design framework,” *IEEE Trans. Antennas Propag.*, vol. 63, no. 6, pp. 2482-2496, Jun. 2015.

[33] G. Oliveri, M. Salucci, N. Anselmi, and A. Massa, “Multiscale system-by-design synthesis of printed WAIMs for waveguide array enhancement,” *IEEE J. Multiscale Multiphys. Comput. Tech.*, vol. 2, pp. 84-96, 2017.

[34] G. Oliveri, A. Gelmini, A. Polo, N. Anselmi, and A. Massa, "System-by-Design multiscale synthesis of task-oriented reflectarrays,” *IEEE Trans. Antennas Propag*., vol. 68, no. 4, pp. 2867-2882, Apr. 2020.

[35] T. Isernia, V. Pascazio, and R. Pierri, “On the local minima in a tomographic imaging technique,” *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1596-1607, Jul. 2001.

[36] M. Salucci, L. Poli, N. Anselmi, and A. Massa, “Multifrequency particle swarm optimization for enhanced multiresolution GPR microwave imaging,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1305-1317, Mar. 2017.

[37] A. Massa, G. Oliveri, M. Salucci, N. Anselmi, and P. Rocca, ”Learning-by-examples techniques as applied to electromagnetics,” *J. Electromagn. Waves Appl.*, vol. 32, no. 4, pp. 516-541, 2018.

[38] M. Laguna and R. Marti, “Experimental testing of advanced scatter search designs for global optimization of multimodal functions,” Journal Global Optimization, vol. 33, no. 2, pp. 235-255, 2005.

[39] A. I. J. Forrester, A. Sobester, and A. J. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*. Hoboken, N.J.: John Wiley & Sons, 2008.

[40] D. R. Jones, M. Schonlau, and W.J. Welch, “Efficient global optimization of expensive black-box functions,” *J. Global Opt*., vol. 13, pp. 455-492, 1998.

[41] M. Salucci, N. Anselmi, G. Oliveri, P. Calmon, R. Miorelli, C. Reboud, and A. Massa, "Real-time NDT-NDE through an innovative adaptive partial least squares SVR inversion approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 11, pp. 6818-6832, Nov. 2016.

[42] L. O. Jimenez-Rodriguez, E. Arzuaga-Cruz, and M. Velez-Reyes, "Unsupervised linear feature-extraction methods and their effects in the classification of high-dimensional data," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 2, pp. 469-483, Feb. 2007.

[43] C. Q. Lam and W. I. Notz, "Sequential adaptive designs in computer experiments for response surface model fit," *Statistics App.*, vol. 6, no. 1-2, pp. 207-233, 2008.

[44] K. Crombecq, D. Gorissen, D. Deschrijver, and T. Dhaene, "A novel hybrid sequential design strategy for global surrogate modeling of computer experiments," *SIAM J. Scientific Comput.*, vol. 33, no. 4, pp. 1948-1974, Jan. 2011.

[45] S. S. Garud, I. A. Karimi, and M. Kraft, "Design of computer experiments: a review," *Comput. Chem. Eng.*, vol. 106, pp. 71-95, May 2017.

[46] H. Liu, Y.-S. Ong, and J. Cai, "A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design," *Struct. Multidiscip. Optim.*, vol. 57, pp. 393-416, Jun. 2017.

[47] D. Maljovec, B. Wang, A. Kupresanin, G. Johannesson, V. Pascucci, and P. Bremer, "Adaptive sampling with topological scores," *Int. J. Uncertain. Quant.*, vol. 3, no. 2, pp. 119-141, Jan. 2013.

[48] D. H. Wolper and W. G. Mcready, "No free lunch theorem for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67-82, Apr. 1997.

[49] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 61-70, 2011.

[50] S. Bilicz, M. Lambert, and S. Gyimothy, "Kriging-based generation of optimal databases as forward and inverse surrogate models," *Inverse Prob.*, vol. 26, no. 7, p. 074012, Jun. 2010.

[51] D. Masotti, L. Poli, M. Salucci, P. Rocca, and A. Costanzo, “An effective procedure for nonlinear dynamic optimization of time-modulated arrays,” *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 10, pp. 2204-2208, Oct. 2019.

[52] V. Rizzoli, D. Masotti, F. Mastri, and E. Montanari, “System-oriented harmonic-balance algorithms for circuit-level simulation,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 2, pp. 256-269, Feb. 2011.

[53] Ansys Corporation, “HFSS,” 2019, Suite v19, Pittsburg (PA), USA.

[54] P. Rocca, G. Oliveri, R. J. Mailloux, and A. Massa, “Unconventional phased array architectures and design methodologies - A review,” *Proc. IEEE*, vol. 104, no. 3, pp. 544-560, Mar. 2016.

[55] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M. Alouini, and R. Zhang, "Wireless communications through reconfigurable intelligent surfaces,” *IEEE Access*, vol. 7, pp. 116753-116773, Aug. 2019.

[56] A. Massa, D. Marcantonio, X. Chen, M. Li, and M. Salucci, “DNNs as applied to electromagnetics, antennas, and propagation - A review,” *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2225-2229, Nov. 2019.

[57] P. Kadlec and Z. Raida, “Multi-objective self-organizing migrating algorithm applied to the design of electromagnetic components,” *IEEE Antennas Propag. Mag.*, vol. 55, no. 6, pp. 50-68, Dec. 2013.

[58] A. Massa, P. Rocca, and G. Oliveri, “Compressive sensing in electromagnetics - A review,” *IEEE Antennas Propag. Mag.*, pp. 224-238, vol. 57, no. 1, Feb. 2015.

FIGURE CAPTIONS

- **Figure 1.** *SbD Paradigm* - Functional scheme of the *SbD*.

- **Figure 2.** *Problem Formulation* - Sketch of the (a) pixel-based and (b) spline-based representation of the *TO* contour $\gamma\left(x,\, y\right)$ [27].

- **Figure 3.** *Training Set Generation* - Pictorial representation of the *DoF*s, reduced features, and cost function spaces and their interconnections for the generation of the training sets $\mathcal{D}_S = \left\{\left(\underline{\Omega}^{(s)};\, \Phi^{(s)}\right);\, s = 1, ..., S\right\}$ and/or $\widehat{\mathcal{D}}_S = \left\{\left(\underline{\chi}^{(s)};\, \Phi^{(s)}\right);\, s = 1, ..., S\right\}$.

- **Figure 4.** *Cost Function Computation* - Actual function value and predictions made by the *RBFN*, *SVR*, and *OK* surrogate models for the (a) Levy's, (b) the Schwefel's, and (c) the Ackley's 1-*D* ($K = 1$) benchmark functions when using $S = 6$ training samples.

- **Figure 5.** *Cost Function Computation* - Actual, $\Phi\left\{\underline{\Omega}\right\}$, predicted, $\widetilde{\Phi}\left\{\underline{\Omega}\right\}$, and confidence bounds, $\widetilde{\Phi}\left\{\underline{\Omega}\right\} \pm \Psi\left\{\underline{\Omega}\right\}$, of the *OK* surrogate model.

- **Figure 6.** *Solution Space Exploration* - Values of the cost function at the convergence, Φ_{StD}, for different "bare" *SbD* algorithms versus the ratio between the training cardinality and the number of variables, S/K: (a) Levy's, (b) Schwefel's, and (c) Ackley's functions with $K = 6$.

- **Figure 7.** *Advanced SbD Strategies* (*SbD-OSF*; Ackley's function) - (a) Training samples generated by the initial *LHS* sampling ($S_0 = 5$) along with those iteratively added by the *SbD-OSF* ($S = 50$); (b) *OK* predictions with the initial (*LHS*) or the *SbD-OSF* training set.

- **Figure 8.** *Advanced SbD Strategies* (*SbD PSO-OK/C*) - Pictorial sketch of the updating rules at the i-th ($i = 1, ..., I$) iteration for the personal best position of each p-th ($p = 1, ..., P$) particle, $\underline{b}_i^{(p)}$ ($\underline{b}_i^{(p)} = \arg\left\{\min_{j=1,...,i}\left[\Phi_j^{(p)}\right]\right\}$).

- **Figure 9.** *Advanced SbD Strategies* (*SbD PSO-OK/C*) - Pictorial sketch of the *confidence-based* updating rules at the i-th ($i = 1, ..., I$) iteration for the global best position, $\underline{g}_i$ ($\underline{g}_i \equiv \underline{\Omega}_i^{best}$).

- **Figure 10.** *SbD-Synthesis* (*Benchmark 1*: $K = 8$, $P = 10$, $I = 200$) - (*a*) 1*D* cut of Φ; (*b*) prediction error, Ξ, and time saving, Δt_{sav}, versus the size of the training set, S; (*c*) performance indexes (i.e., $\Delta\Phi$ and Δt_{sav}) versus S.

- **Figure 11.** *SbD-Synthesis* (*Benchmark 1*: $K = 8$, $P = 10$, $I = 200$) - Design quality index, $\Delta\Phi$, and SbD time saving, Δt_{sav}, versus S.

- **Figure 12.** *SbD-Synthesis* (*Benchmark 1*: $K = 8$, $P = 10$, $I = 200$, $S_0 = 40$, $S = 100$) - (*a*) Evolution of the best value of the cost function, Φ_i^{best}, versus the iteration index, i ($i = 1, ..., I$) and (*b*) plot of the instantaneous directivity over one modulation period T.

- **Figure 13.** *SbD-Synthesis* (*Benchmark 1*: $K = 8$, $P = 10$, $I = 200$, $S_0 = 40$, $S = 100$) - One-dimensional cut of $\Phi\left\{\underline{\Omega}\right\}$ and *OK* predictions when applying the *PSO-OK/C*.

- **Figure 14.** *SbD-Synthesis* (*Benchmark 1*: $K = 8$, $P = 10$, $I = 200$, $S_0 = 40$, $S = 100$) - Evolution of $\widetilde{\Phi}\left\{\underline{g}_j\right\}$ versus the iteration index, i ($i = 1, ..., I$) and cost function values at the J control-iterations, $\{\Phi\left\{\underline{g}_j\right\}; j = 1, ..., J\}$.

- **Figure 15.** *SbD-Synthesis* (*Benchmark 2*: $K = 4$) - Geometry of (*a*) the complete antenna array system modeled with *HFSS* and (*b*) details of the elementary radiator, namely a cavity-backed slot-fed square patch.

- **Figure 16** *SbD-Synthesis* (*Benchmark 2*: $K = 4$, $P = 4$, $I = 50$, $S_0 = 20$, $S = 40$) - Evolution of the best value of the cost function, Φ_i^{best}, versus the iteration index, i ($i = 1, ..., I$).

- **Figure 17.** *SbD-Synthesis* (*Benchmark 2*: $K = 4$, $P = 4$, $I = 50$, $S_0 = 20$, $S = 40$) - Optimized realized gain patterns for the j-th ($j = 1, 2$) polarization when steering the main beam towards (*a*) $(\theta_0, \varphi_0) = (90, 0)$ [deg] and (*b*) $(\theta_0, \varphi_0) = (105, 60)$ [deg].

TABLE CAPTIONS

- **Table I.** *Advanced SbD Strategies* (*SbD PSO-OK/C*) - Personal best update rules.

- **Table II.** *Advanced SbD Strategies* (*SbD PSO-OK/C*) - Global best update rules.

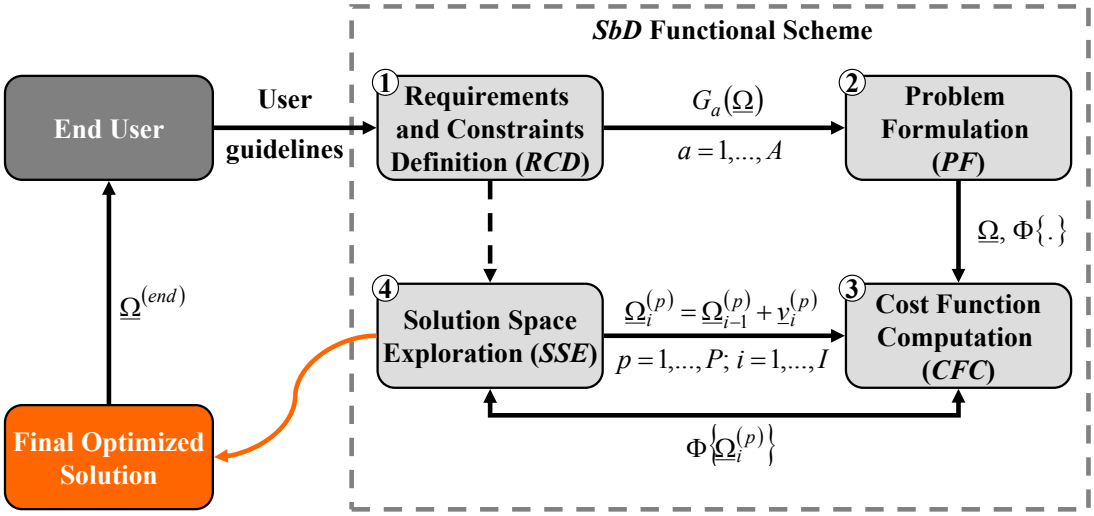
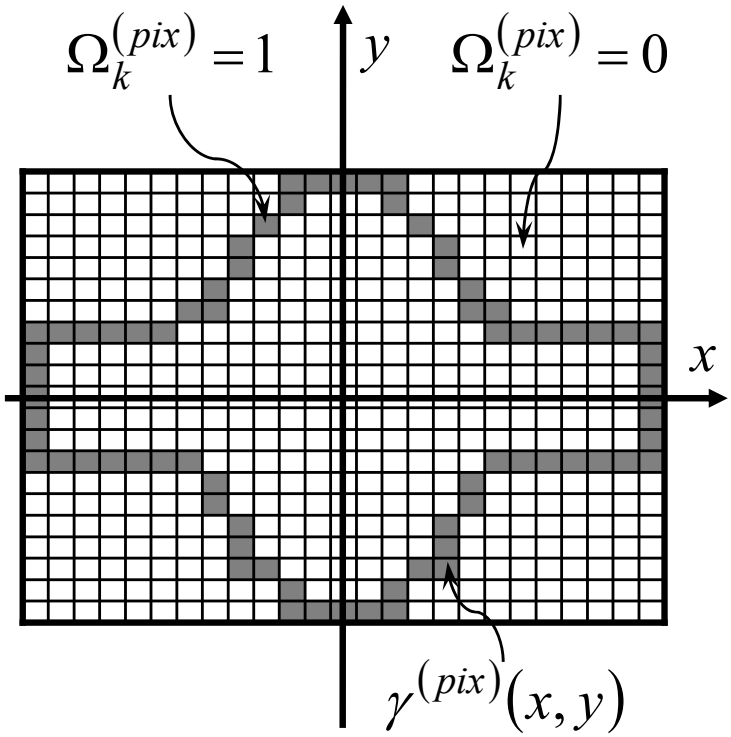
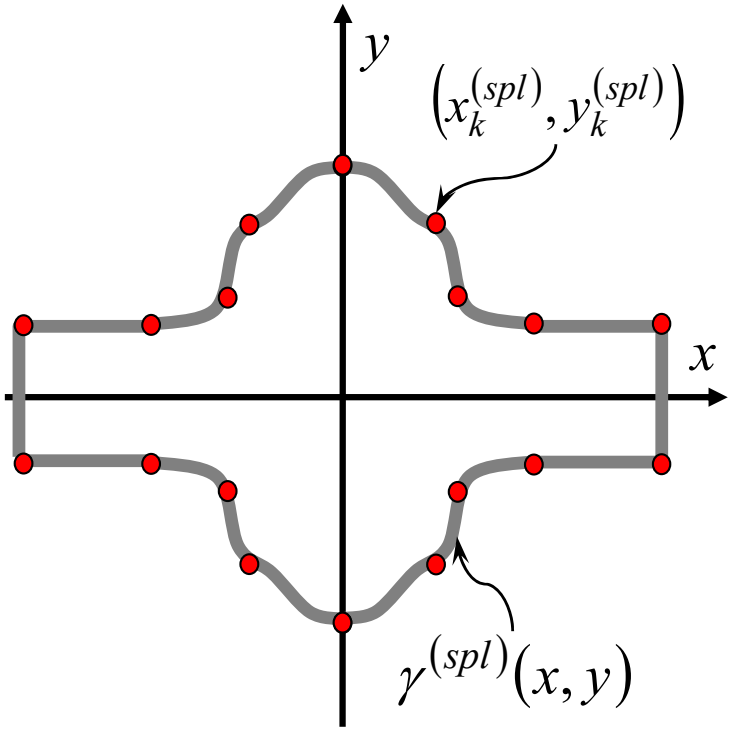


Fig. 1 - A. Massa et *al.*, "On the Design of Complex ..."



(*a*)



(*b*)

Fig. 2 - A. Massa et *al.*, "On the Design of Complex ..."

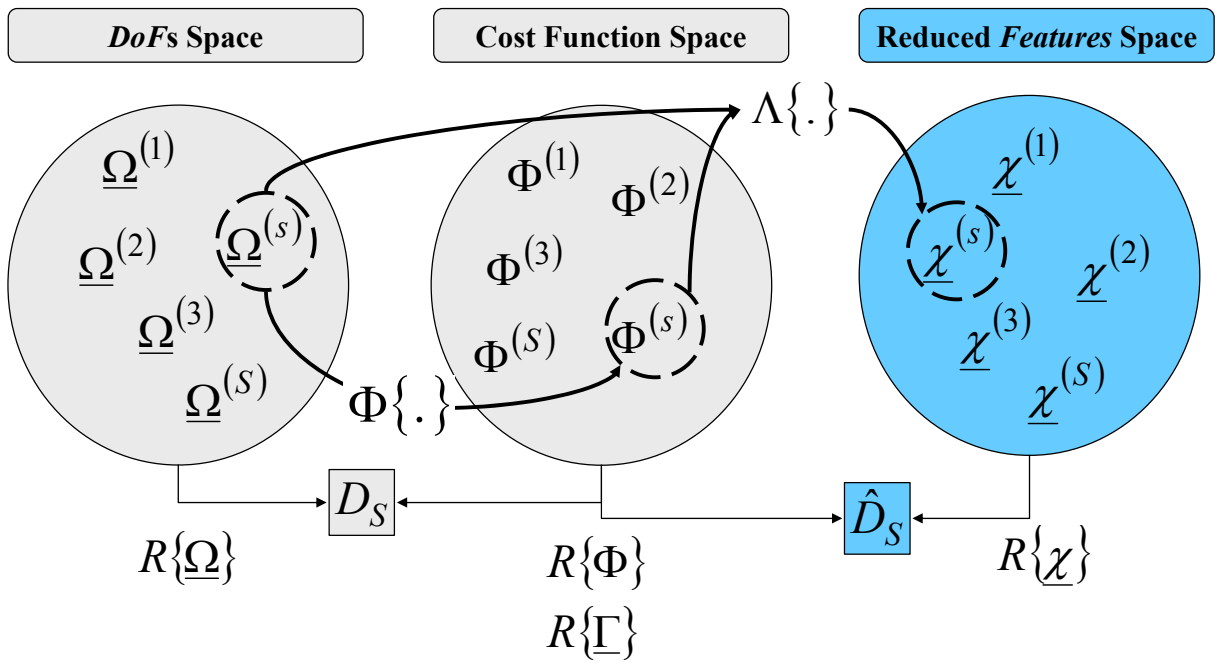
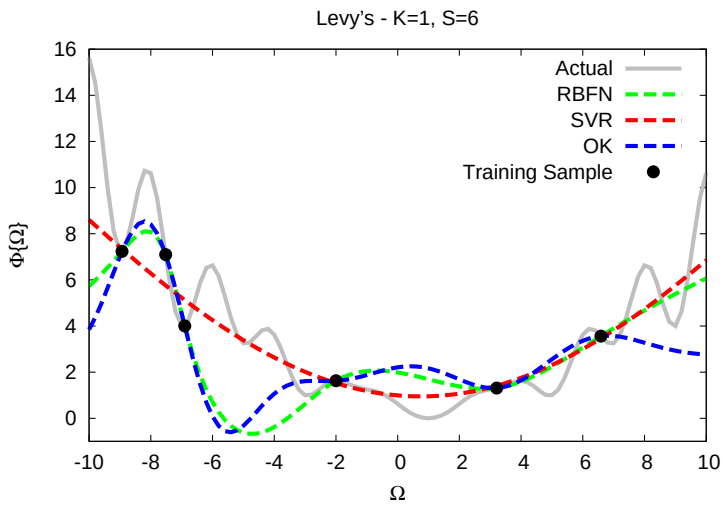
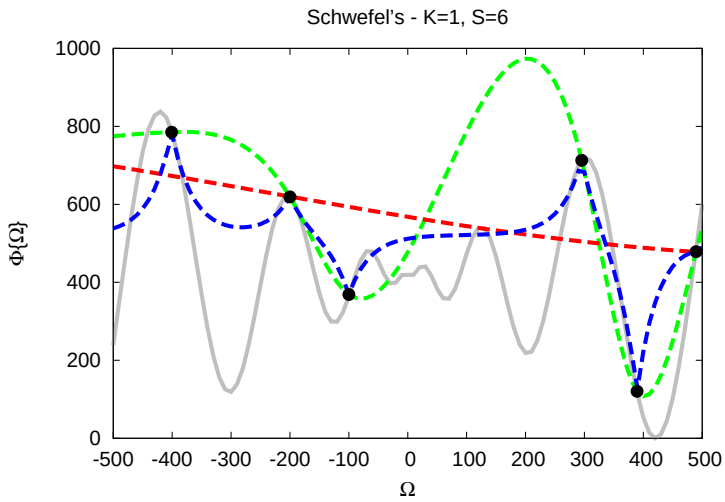


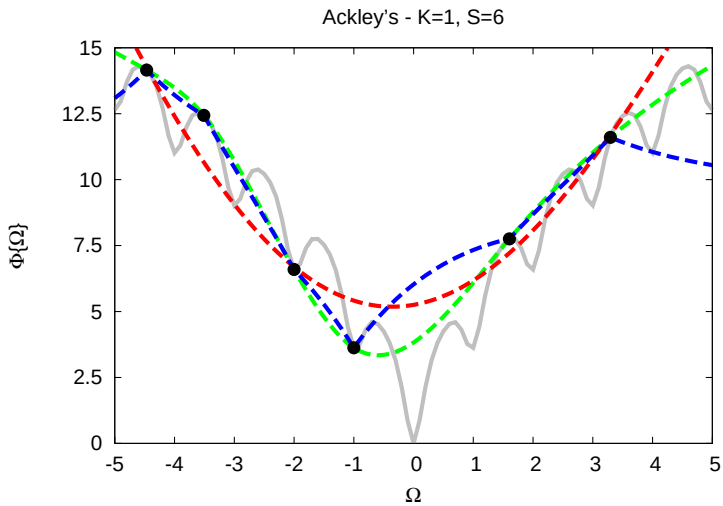
Fig. 3 - A. Massa et *al.*, "On the Design of Complex ..."



(*a*)



(*b*)



(*c*)

Fig. 4 - A. Massa et *al.*, "On the Design of Complex ..."

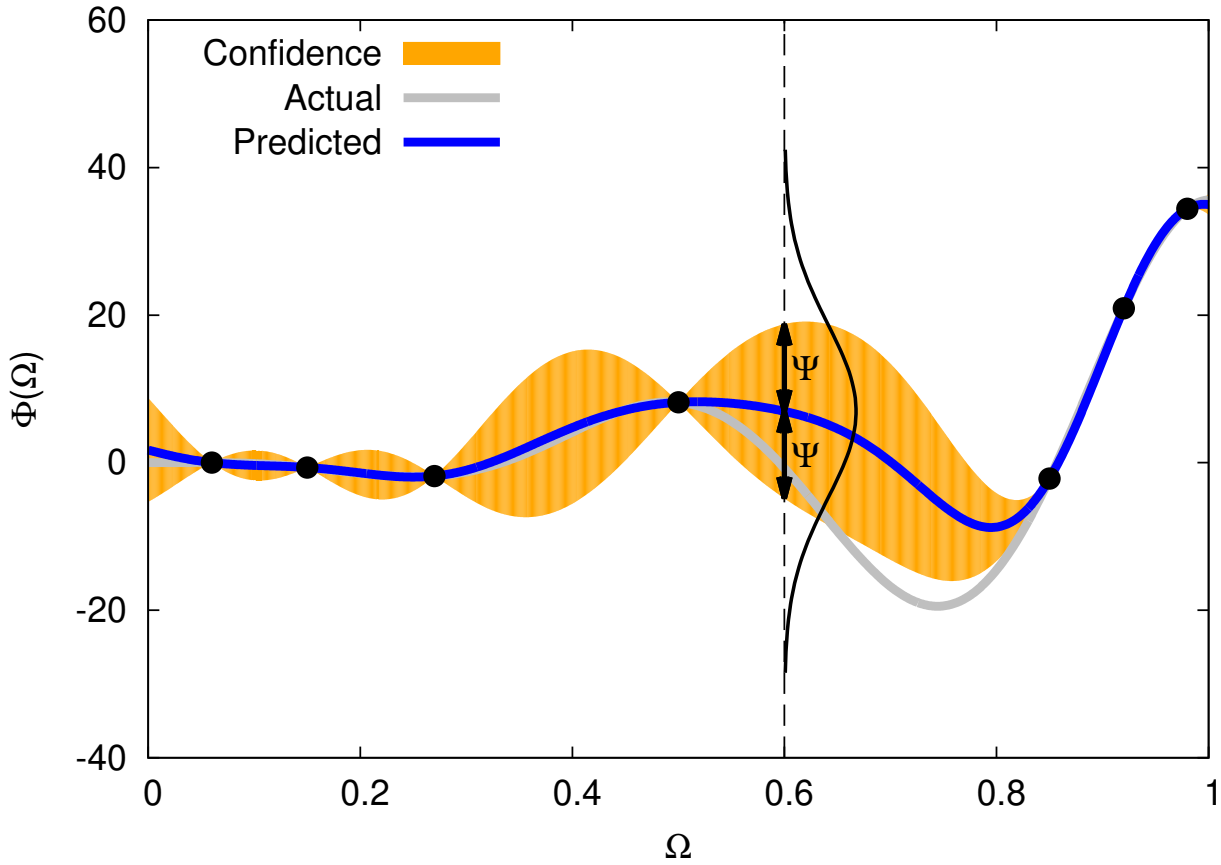
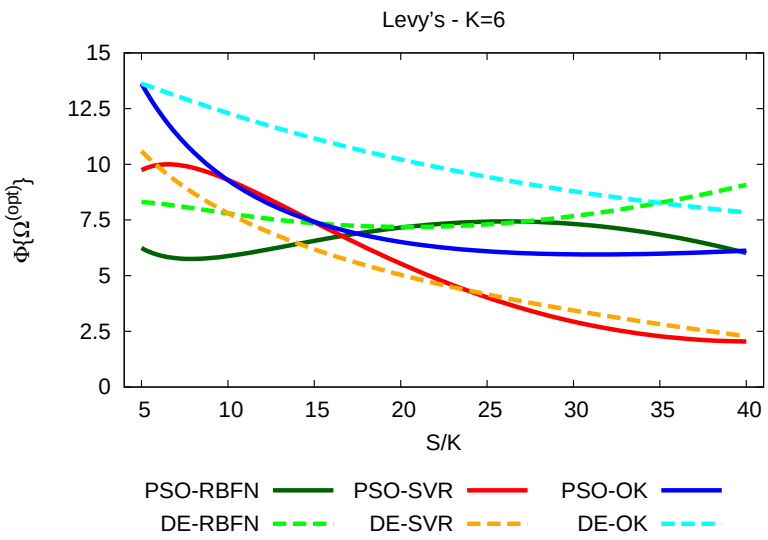
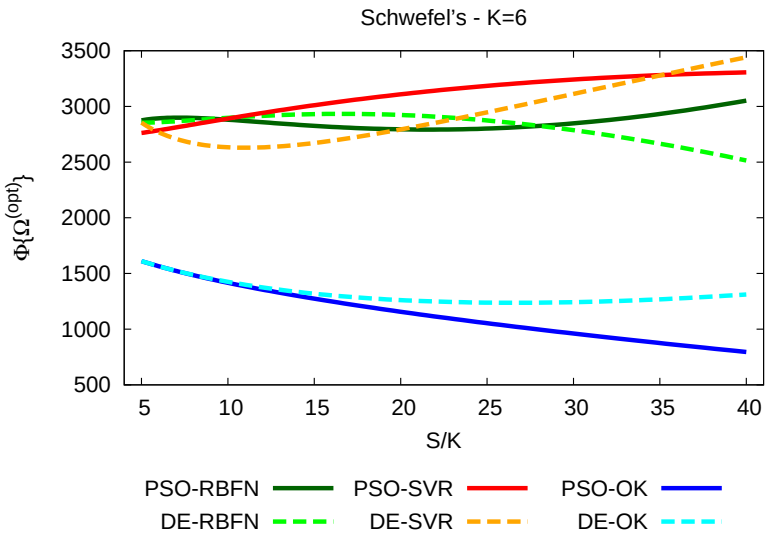


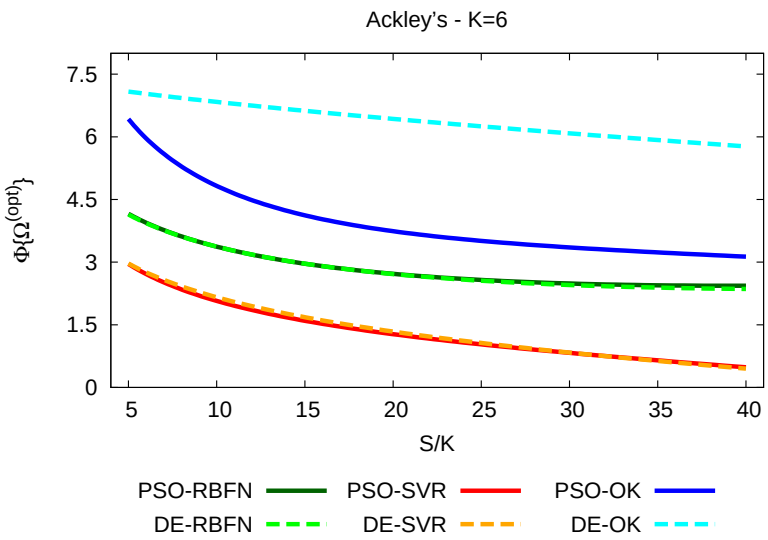
Fig. 5 - A. Massa et *al.*, "On the Design of Complex ..."



(*a*)

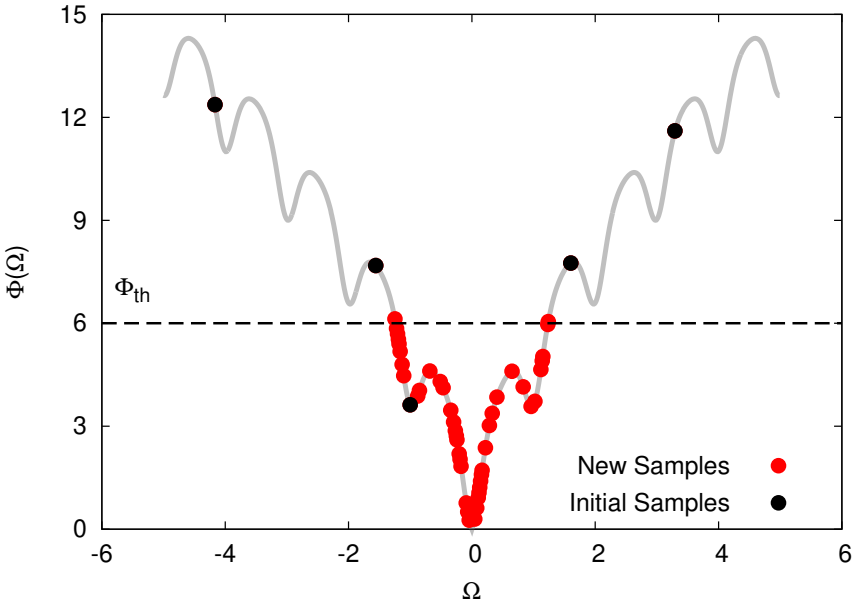


(*b*)

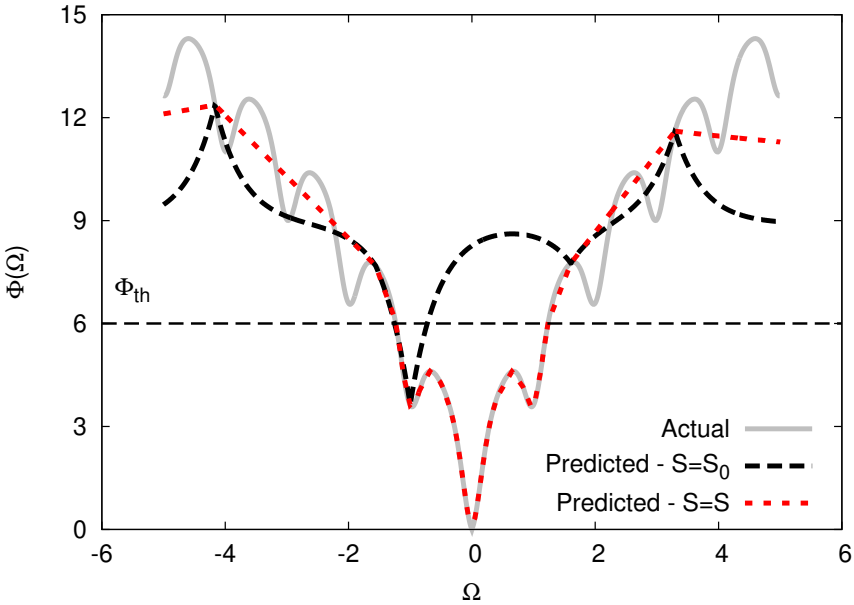


(*c*)

Fig. 6 - A. Massa et *al.*, "On the Design of Complex ..."



(*a*)



(*b*)

Fig. 7 - A. Massa et *al.*, "On the Design of Complex ..."

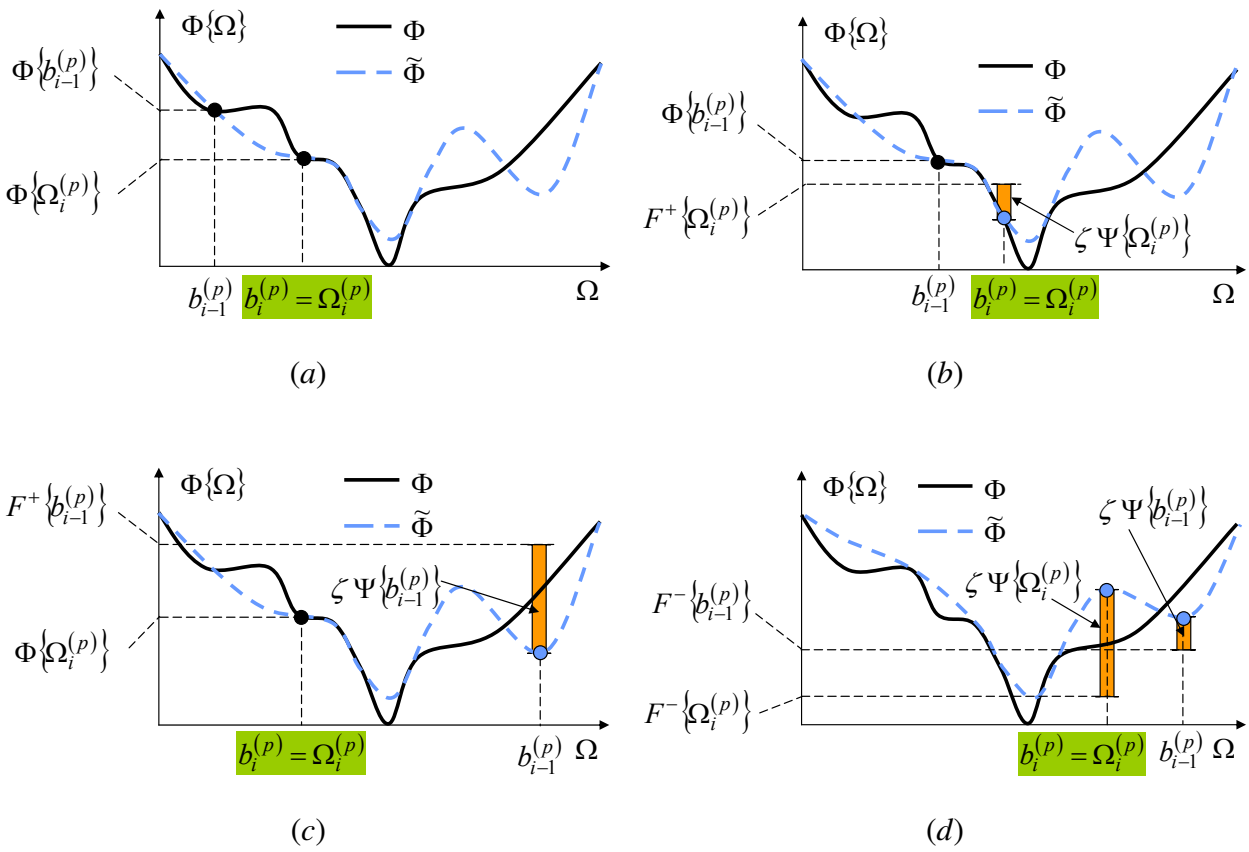
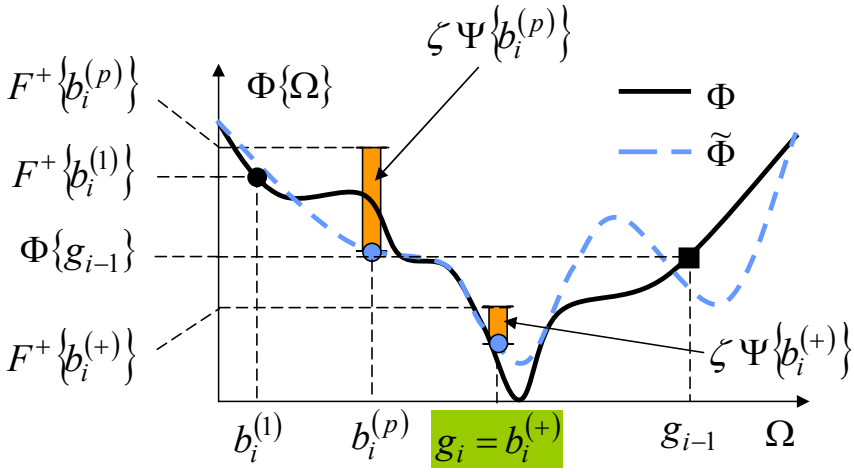
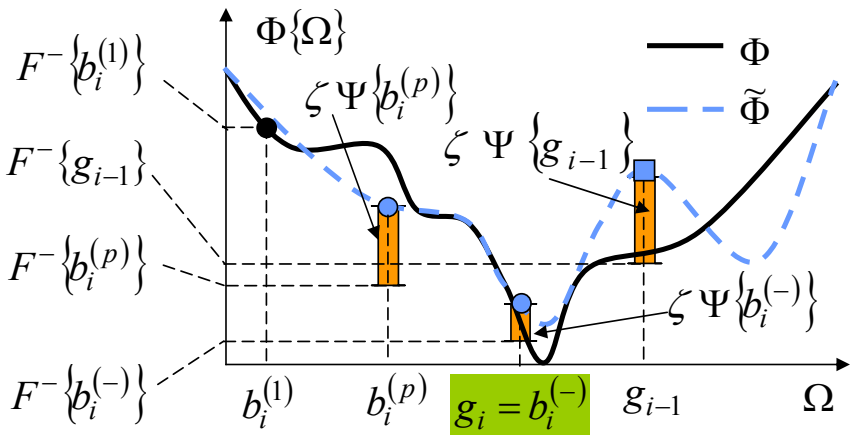


Fig. 8 - A. Massa et *al.*, "On the Design of Complex ..."

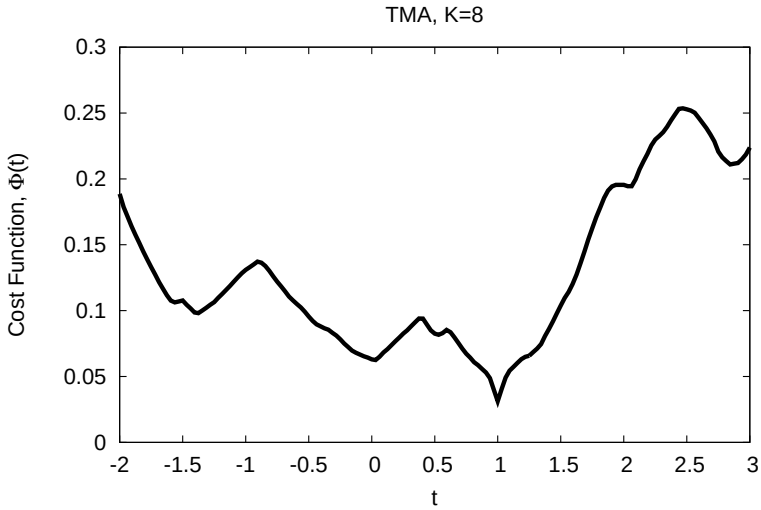


(*a*)

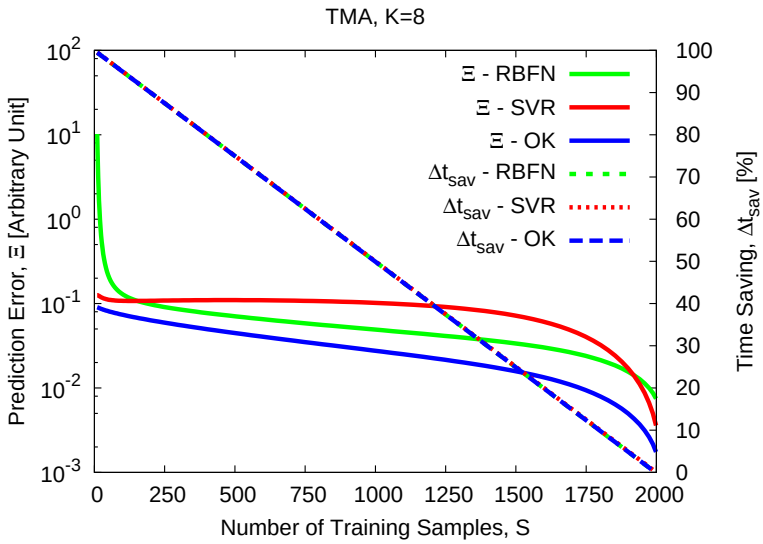


(*b*)

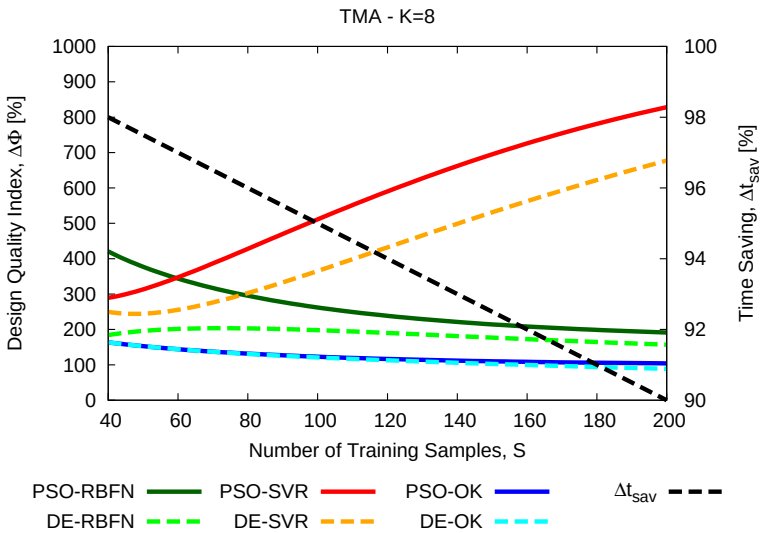
Fig. 9 - A. Massa et *al.*, "On the Design of Complex ..."



(*a*)



(*b*)



(*c*)

Fig. 10 - A. Massa et *al.*, "On the Design of Complex ..."

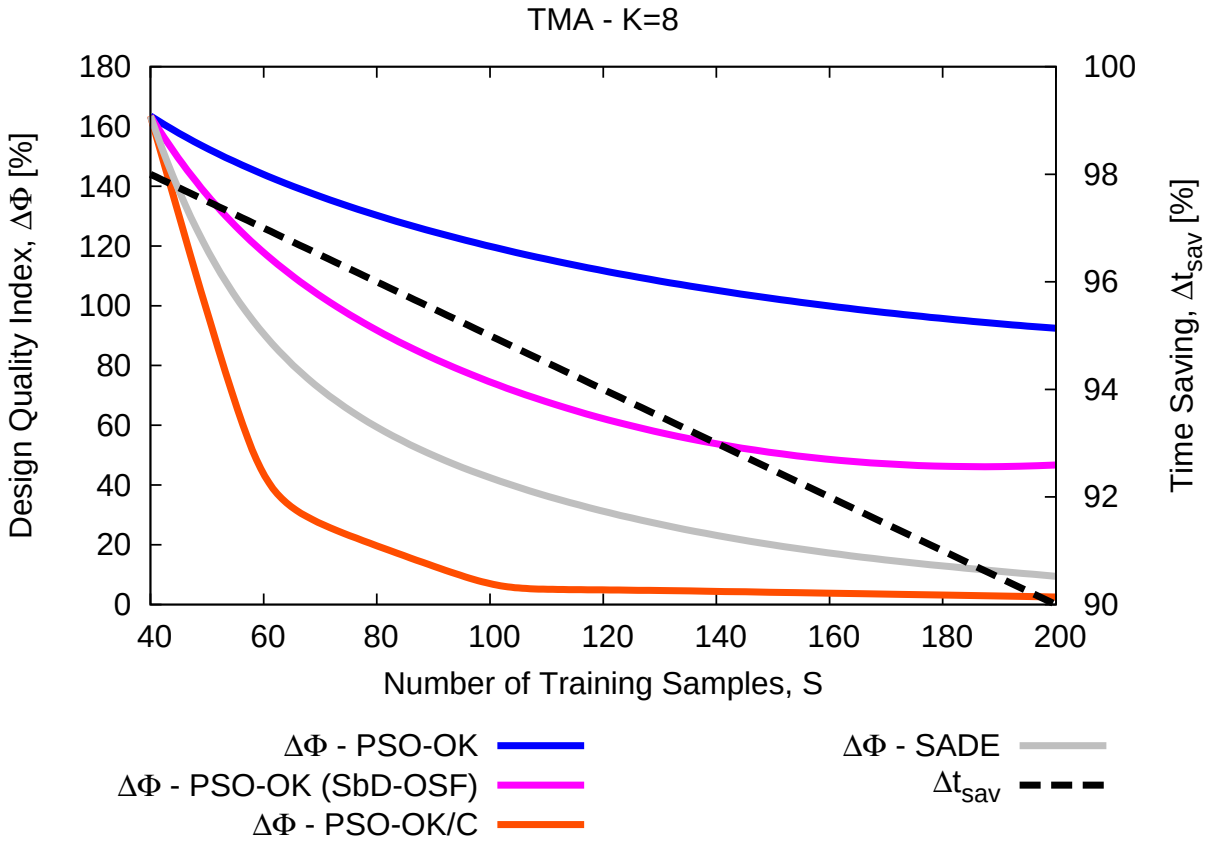
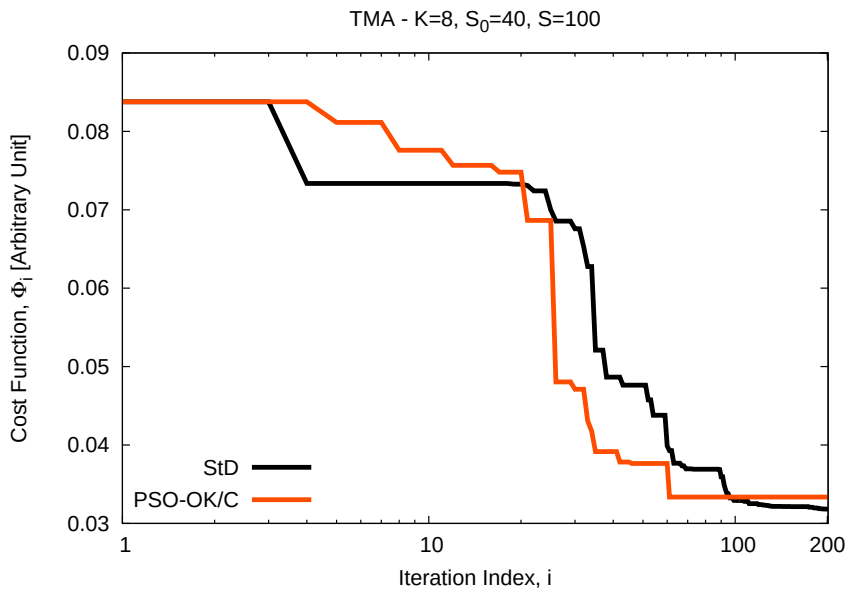
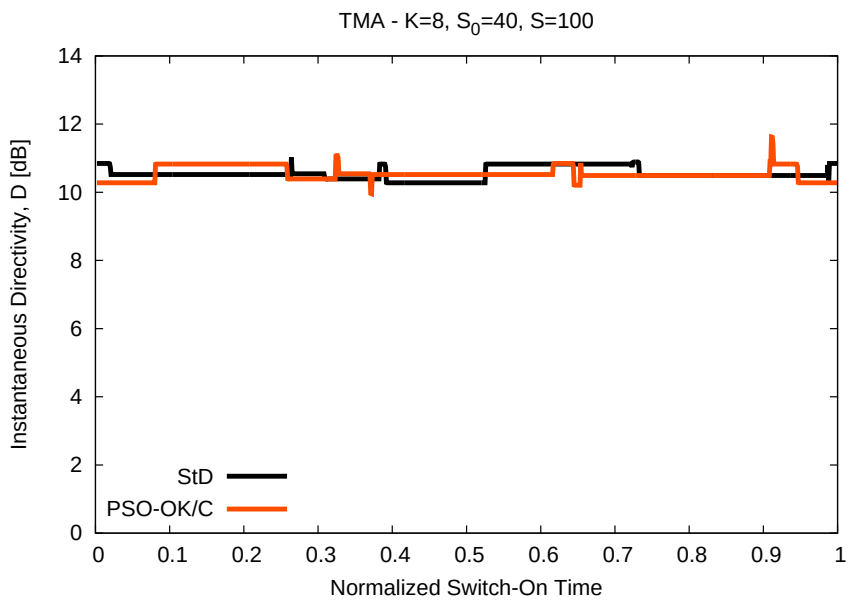


Fig. 11 - A. Massa et *al.*, "On the Design of Complex ..."



(*a*)



(*b*)

Fig. 12 - A. Massa et *al.*, "On the Design of Complex ..."

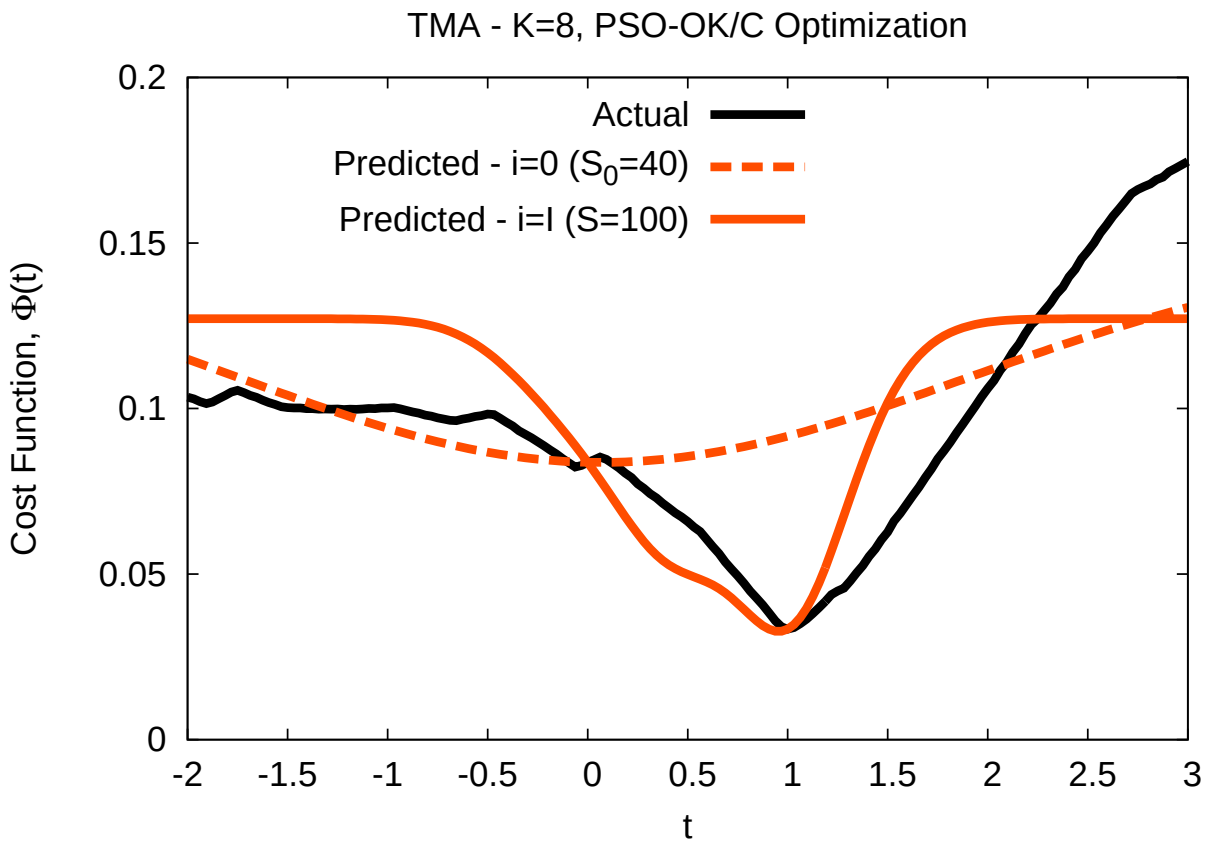


Fig. 13 - A. Massa et *al.*, "On the Design of Complex ..."

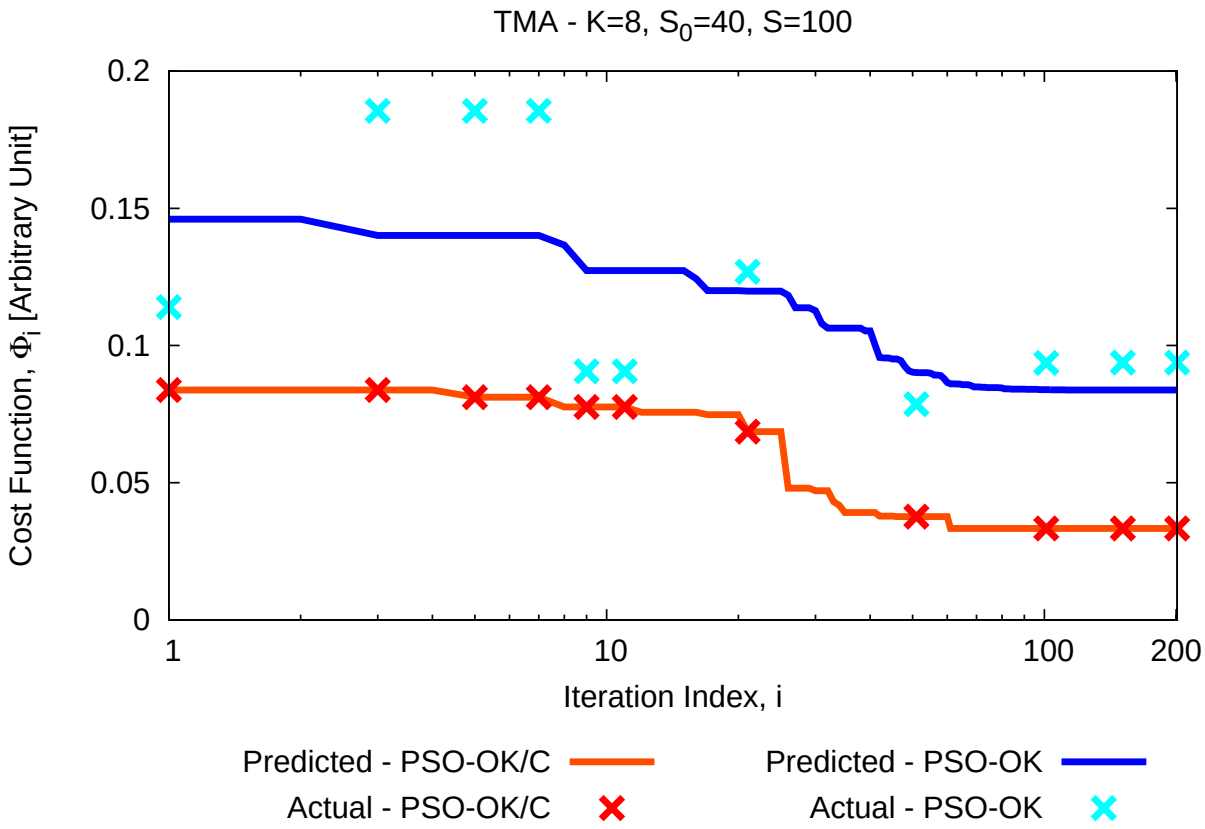
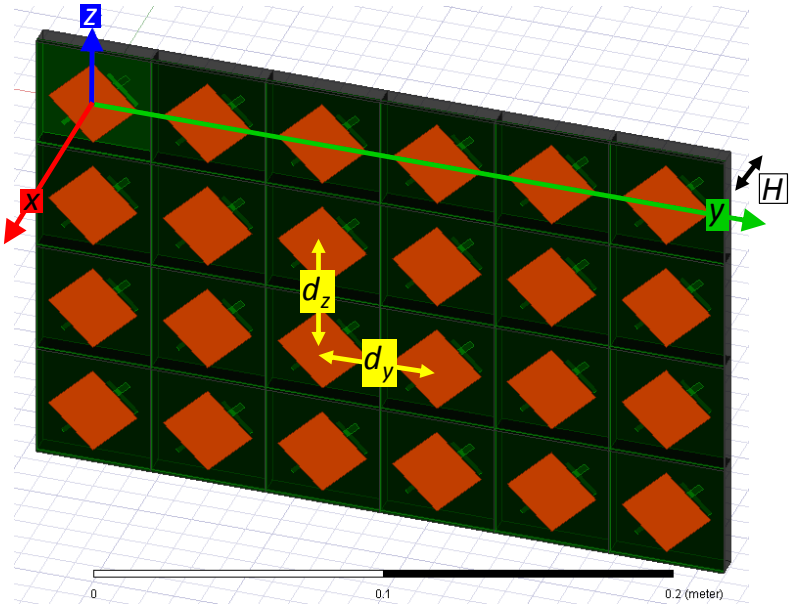
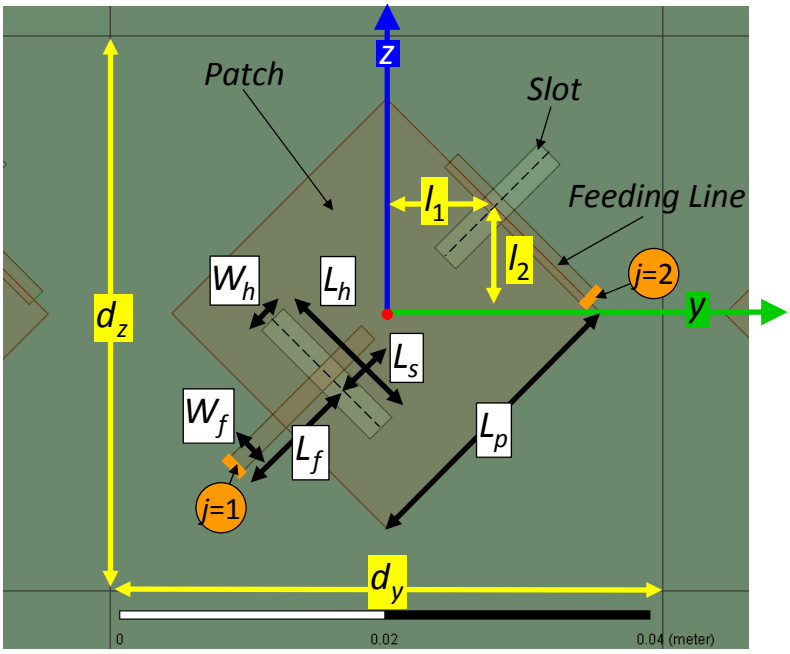


Fig. 14 - A. Massa et *al.*, "On the Design of Complex ..."



(*a*)



(*b*)

Fig. 15 - A. Massa et *al.*, "On the Design of Complex ..."

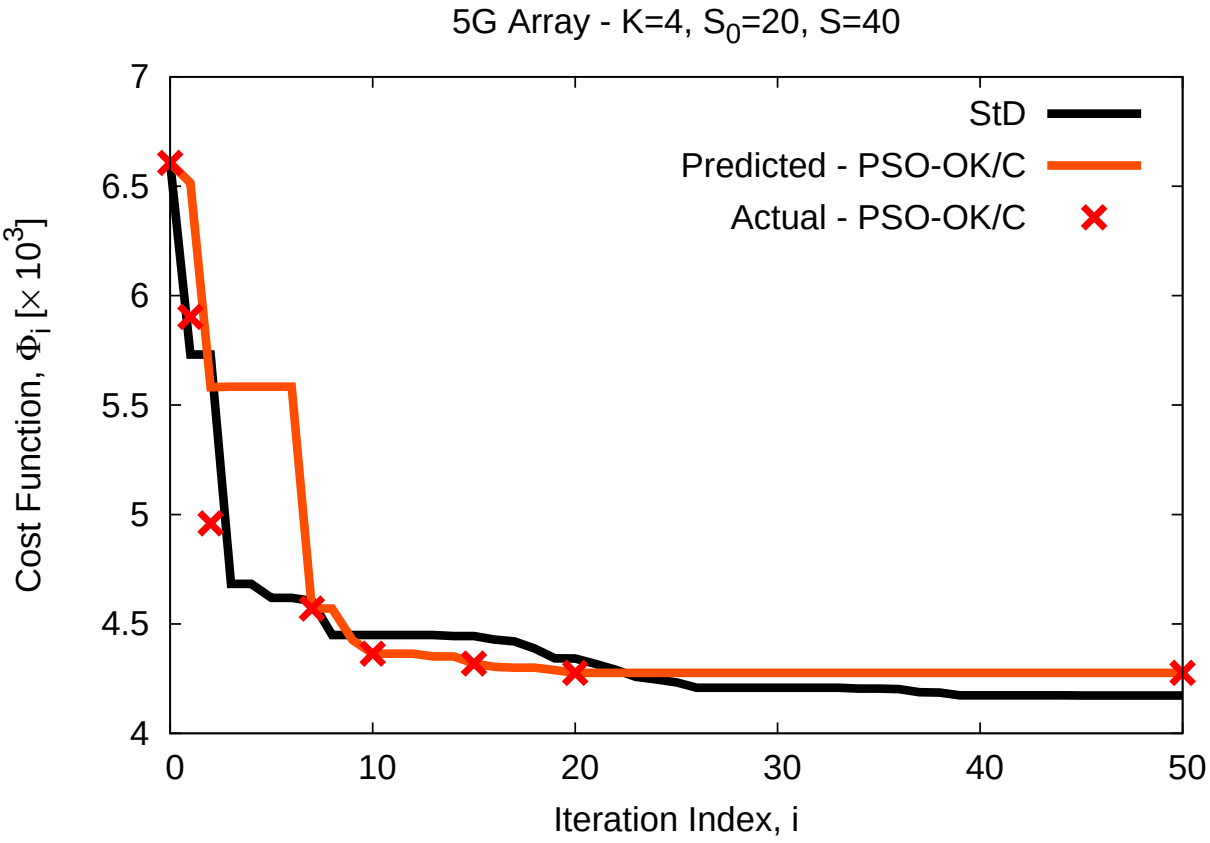
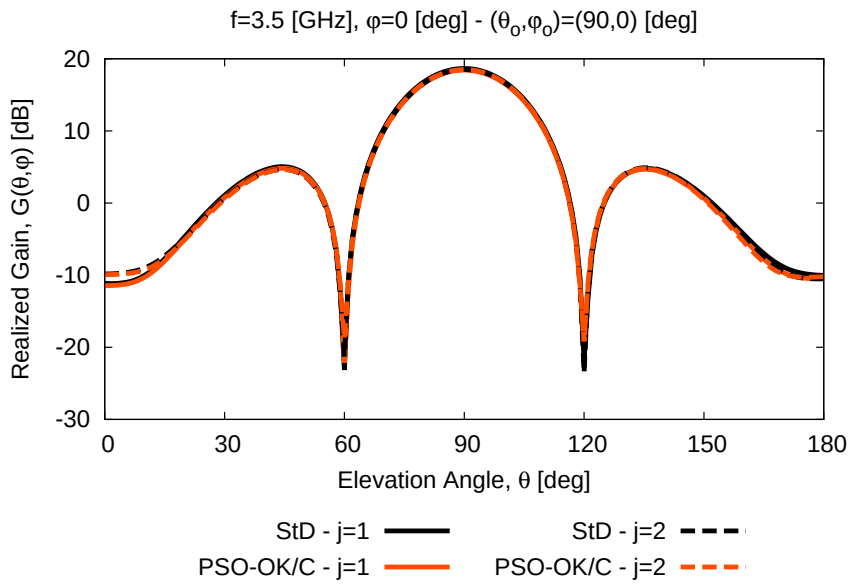
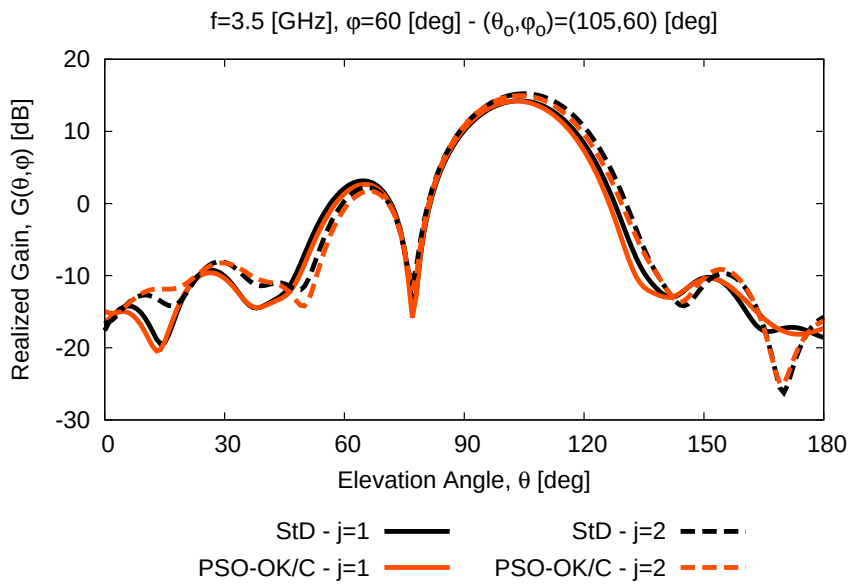


Fig. 16 - A. Massa et *al.*, "On the Design of Complex ..."



(*a*)



(*b*)

Fig. 17 - A. Massa et *al.*, "On the Design of Complex ..."

	$\underline{\Omega}_i^{(p)}$ Simulated	$\underline{\Omega}_i^{(p)}$ Predicted
$\underline{b}_{i-1}^{(p)}$ Simulated	$\underline{b}_i^{(p)} = \begin{cases} \underline{\Omega}_i^{(p)} & \text{if } \Phi\left\{\underline{\Omega}_i^{(p)}\right\} < \Phi\left\{\underline{b}_{i-1}^{(p)}\right\} \\ \underline{b}_{i-1}^{(p)} & \text{otherwise} \end{cases}$	$\underline{b}_i^{(p)} = \begin{cases} \underline{\Omega}_i^{(p)} & \text{if } \mathcal{F}^+\left\{\underline{\Omega}_i^{(p)}\right\} < \Phi\left\{\underline{b}_{i-1}^{(p)}\right\} \\ \underline{b}_{i-1}^{(p)} & \text{otherwise} \end{cases}$
$\underline{b}_{i-1}^{(p)}$ Predicted	$\underline{b}_i^{(p)} = \begin{cases} \underline{\Omega}_i^{(p)} & \text{if } \Phi\left\{\underline{\Omega}_i^{(p)}\right\} < \mathcal{F}^+\left\{\underline{b}_{i-1}^{(p)}\right\} \\ \underline{b}_{i-1}^{(p)} & \text{otherwise} \end{cases}$	$\underline{b}_i^{(p)} = \begin{cases} \underline{\Omega}_i^{(p)} & \text{if } \mathcal{F}^-\left\{\underline{\Omega}_i^{(p)}\right\} < \mathcal{F}^-\left\{\underline{b}_{i-1}^{(p)}\right\} \\ \underline{b}_{i-1}^{(p)} & \text{otherwise} \end{cases}$

Tab. I - A. Massa et *al.*, "On the Design of Complex ..."

$\underline{g}_{i-1}$ Simulated	$\underline{g}_i = \begin{cases} \underline{b}_i^{(+)} & \text{if } \mathcal{F}^+\left\{\underline{b}_i^{(+)}\right\} < \Phi\left\{\underline{g}_{i-1}\right\} \\ \underline{g}_{i-1} & \text{otherwise} \end{cases}$ $\text{s.t. } \underline{b}_i^{(+)} = \arg\left[\min_{p=1,\ldots,P} \mathcal{F}^+\left\{\underline{b}_i^{(p)}\right\}\right]$
$\underline{g}_{i-1}$ Predicted	$\underline{g}_i = \begin{cases} \underline{b}_i^{(-)} & \text{if } \mathcal{F}^-\left\{\underline{b}_i^{(-)}\right\} < \mathcal{F}^-\left\{\underline{g}_{i-1}\right\} \\ \underline{g}_{i-1} & \text{otherwise} \end{cases}$ $\text{s.t. } \underline{b}_i^{(-)} = \arg\left[\min_{p=1,\ldots,P} \mathcal{F}^-\left\{\underline{b}_i^{(p)}\right\}\right]$

Tab. II - A. Massa et *al.*, "On the Design of Complex ..."