Progress In Electromagnetics Research, PIER 56, 195–232, 2006

GLOBAL OPTIMIZATION AND ANTENNAS SYNTHESIS AND DIAGNOSIS, PART ONE: CONCEPTS, TOOLS, STRATEGIES AND PERFORMANCES

A. Capozzoli and G. D'Elia

Dipartimento di Ingegneria Elettronica e delle Telecomunicazioni
Università di Napoli "Federico II"
Via Claudio, 21, 80125 Napoli, Italia

Abstract—This is the first of two companion papers on global optimization and antenna analysis and synthesis. In Part I, an analysis of the problems involved in Global Optimization is presented by critically discussing the basic concepts and tools, the performances to be expected, the required computational complexity and the guidelines to select algorithms solving efficiently the problem at hand. The relevance of stochastic techniques is enhanced and the role of double phase algorithms is stressed. The proof of the convergence property of an idealized version of a simplified evolutionary algorithm is provided. In Part II, the selected algorithm, a hybrid evolutionary algorithm, is tested against two real world problems relevant in electromagnetics, the power synthesis of contoured beam hybrid reflector antennas and the reflector antenna diagnosis from only amplitude data. The results of an extensive numerical analysis are presented.

1. INTRODUCTION

The solution of many practical problems in several application areas, from engineering to economics, requires the global optimization of a non linear multimodal objective functional. F.i., if we restrict our attention only to electromagnetics, solving inverse problems such as antennas diagnosis and synthesis, the practical applications discussed in the second part of our paper, sources and scatterers localization, microwave tomography and wave front reconstruction in optical astronomy, frequently demands for the optimization of a functional that, in principle, has local optima.

In many instances, such optimization problems are afforded by exploiting local optimization techniques [1–8].

However, in the case of multiextremal functionals, this choice suffers from a relevant drawback: depending on the starting point of the searching procedure, the algorithm can be trapped into suboptimal solutions or useless false solutions [9–19].

Accordingly, new strategies able to overcome or significantly reduce the occurrence of false solutions are of serious interest [20–26].

Two different approaches can be followed to overcome such a drawback.

By means of a proper reformulation of the problem at hand, the first strategy should lead to the "convexification" of the objective functional and obtain an unimodal optimization problem [27]. This effect can be obtained by using additional information and constraints.

Obviously, as long as it is possible, this approach seems to be the privileged way toward effectiveness and reliability. However, it is not always a simple task to be accomplished. In some sense, asking always for unimodal objective functions is analogous to requiring that, since linear operators are simpler to be managed, all scientific problems should be modeled by means of only linear relationships. On the other side, in principle, "convexification" is not always strictly necessary and/or convenient. F.i., in the antenna synthesis area, each reformulation worsening the quality of the optimal solution should be avoided if all the feasibility constraints have been already enforced. Analogously, in the diagnosis area, no additional information should be required beside the ones strictly needed to achieve the well position of the problem.

The second approach asks for advanced optimization techniques of global nature, able to traps out from local optima and transcend the incumbent solution [20–26]. Concerning this point, it must be stressed that, as long as the absence of local optima has not been proved, global optimization algorithms should always be used.

However, even if the problem of global optimization is of particular relevance, the available theoretical and algorithmic results have enjoyed a wide spreading in the scientific community of practitioners only in the recent years so that, in the practical applications, local optimization techniques are very often adopted to optimize multimodal functionals. This behavior appears evident if we observe that commercial numerical packages implementing some of the most popular global optimization algorithms are becoming available only recently.

The reasons of this lacking of attention in the past years should be ascribed both to the absence of sufficient computing resources and to the youngness of the discipline. In fact, only in the last three decades the need for more accurate solutions to more sophisticated practical problems and the renewal of attention by researchers all over the world have produced a significant theoretical evolution in this area. In particular, new algorithms able to give new solutions to old problems as well as solutions to problems never afforded before have been designed [20–26].

Anyway, the distance between the theoretical world of Mathematical Programming applied to Global Optimization and the scientific community directly involved in the solution of practical problems is still wide [28]. Obviously, the theoretical results and the advanced optimization tools can significantly contribute to managing the practical optimization problems of interest. On the other side, neither a dramatic simplification of the searching procedure nor a drastic reduction of the computational complexity should be expected. In fact, the exploration of a complex landscape, when it does not enjoy strong known peculiar properties, is always a very hard task and necessarily requires a high computing effort. To clarify such concept, let us argue from analogy. The topographical mapping of a region on the Earth surface, still based on a very old idea, has been recently made very efficient by exploiting the advanced technology of the Remote Sensing, the counterpart of the advanced computing resources in the optimization area [29].

On the other side, theorists of Global Optimization claim the absence of real world applications of global algorithms able to effectively test their performances (test functions are generally unable to provide a convincing measure of the attainable performance in practical problems) [21].

The aim of the first part of the present work is to discuss some basic concepts and tools in global optimization, the performances to be expected from global tools, the required computational complexity and the guidelines to select the algorithms and efficiently solve the problem at hand. The first part enlighten the reasoning that led us to select a particular approach, a hybridized Evolutionary Algorithm (EA), to solve the two relevant problems in applied electromagnetics discussed in the second part of the paper.

Furthermore, a new convergence property of a simplified and idealized version of an EA is provided. In particular, the probabilistic analysis shows that such an algorithm is able to find the solution of a global optimization problem with a finite number of steps, a property already proved in the literature for Adaptive Search Algorithms (Simulated Annealing) [30].

The aim of the second part of the paper is to present the application of our global optimization algorithm to reflector antennas power pattern synthesis and to reflector antennas surface diagnosis from amplitude only far-field data. These two problems are of interest for the electromagnetic community as long as high performances are required in the modern telecommunication and radar applications.

2. PROBLEM FORMULATION

In the introductory Section we disclosed that Global Optimization deals with finding the absolute extreme points of a function. In this Section, we formally introduce the problem and we discuss the conditions involved in its practical solution.

2.1. The Problem

Let us now state the problem in a general mathematical framework. We assume that the functional to be optimized, in the following the objective functional, is a real valued function of n real variables. The set of real numbers will be denoted with $\mathbf{R}$ and, as usual, the set of vectors of n real numbers, $\underline{x} = (x_1, \ldots, x_n)$, by $\mathbf{R}^n = \overbrace{\mathbf{R} \times \ldots \times \mathbf{R}}^{n \text{ times}}$. Both sets are considered equipped with the standard inner product and the corresponding induced norm, we denote with $< \cdot, \cdot >$ and $\| \cdot \|$, respectively. According to our notations, the functionals to be optimized herein will be in the form:

$$F|(x_1, \ldots, x_n) = \underline{x} \in \Theta \subseteq \mathbf{R}^n \rightarrow \mathbf{R} \tag{1}$$

where the domain of definition Θ is assumed to be an open set of $\mathbf{R}^n$ to avoid troubles at boundary points.

Often a priori information on the solution to be searched are available, suggested, for instance, by the physics of the problem. They should be taken into account in the optimizing procedure both to obtain an acceptable solution and to make smaller the search domain. To this end a set, Λ say, of the feasible solutions is introduced, generally specified by known constraint functions $C_i|\Theta \subseteq \mathbf{R}^n \rightarrow \mathbf{R}$:

$$\Lambda = \{\underline{x} \in \Theta | C_i(\underline{x}) \leq 0, \; i \in \{1, \ldots, m\}\} \tag{2}$$

However, as will be further discussed later on, the complexity of the optimization problem, beside the objective functional, depends also on the topological properties of Λ. As a consequence, it could not be always convenient to increase the complexity of Λ by incorporating any available information.

The global minimization problem[†] for the function F over the set

[†] All the definitions and results can be straightforwardly extended to global maximization. In this case the maximum of F will be denoted by F^* and the set of points of Λ wherein F attains f^* by $\underset{\Lambda}{\operatorname{argmax}} F = \{\underline{x}^* \in \Lambda | F(\underline{x}^*) = f^* \forall \underline{x} \in \Lambda\}$

Λ amounts to find:

a) the lower bound of F over Λ, f_* say;

b) the set of global optimizers, i.e., the set:

$$\arg\min_{\Lambda} F = \{\underline{x}_* \in \Lambda | F(\underline{x}_*) \leq F(\underline{x}) \forall \underline{x} \in \Lambda\} \tag{3}$$

The definition evidently enlighten the substantial difference between global and local optimization. Local minimization only requires to find a point of Λ such that the inequality (3) is satisfied in a neighborhood of the point $\underline{x}_*$.

Obviously, assumptions on F and Λ are needed to ensure the feasibility of the global optimization problem [31]. Beside the trivial hypothesis that the domain of F is not void, the more relevant requirement should ensure the existence of a minimum of F over Λ. It is worth noting that, when dealing with global minimization, maxima of F are not of concern; in particular the function F can even be not upper bounded at all over Λ [32]. As a consequence, the weaker hypothesis generally considered is that the objective functional is lower semi-continuous (l.s.c.), a property that ensures the existence of a minimum of F on compacta [32, 33], or that F enjoys other properties when the search domain is unbounded [31, 32].

2.2. Feasibility Conditions

Once the existence of the absolute minimum has been ensured, we have to define those requirement making the optimization problem solvable from an algorithmic point of view. In other words we need to discuss if and under which assumptions a real-world algorithm is able to get the optimum.

As shown by Dixon about twenty years ago [34], the global optimization problem is unfeasible in a finite number of function evaluations. The sampled minimum value can in fact be arbitrarily different from the real minimum of the objective function.

To clarify this point let us consider a l.s.c. function on an open set Θ, say $F|\Theta \subseteq \mathbf{R}^n \to \mathbf{R}$, and a feasible compact set $\Lambda \subseteq \Theta$. We denote with $\{\underline{x}_1, \ldots, \underline{x}_k\}$ the sampling points till the k-th iteration and consider the function:

$$F' = F + aB\left(\frac{\|\underline{x} - \underline{x}_0\|^2}{b}\right) \tag{4}$$

where $\underline{x}_0$ is a point of Λ, a and b are two scalars and B is the well known bell function of the Theory of Distributions [35]

$$B(x) = \begin{cases} -\dfrac{1}{e}\exp\left(-\dfrac{1}{1-x^2}\right) & x \in [-1,1] \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where e is the neper number.

If we choose

$$\underline{x}_0 \in \left\{\underline{x} \in \Lambda \backslash \{\underline{x}_i\}_{i\in\{1,\ldots,k\}}\right\} \tag{6a}$$

$$b < \frac{1}{2}\min_{i\in\{1,\ldots,k\}} \|\underline{x}_0 - \underline{x}_i\|^2 \tag{6b}$$

$$a < \min_{\Lambda} F - F(\underline{x}_0) \tag{6c}$$

the function F' assumes the same values of F at the sampling points, but its absolute minimum can be located "everywhere" in Λ and its value can be arbitrarily small.

Accordingly, by only inspecting the levels of the objective function at the sampling points, an algorithm cannot distinguish between F and F', neither smoothness hypothesis's on F are useful since the bell function B is indefinitely continuously differentiable. Moreover, sampling the derivatives does not help.

However, this dramatic result can be applied also to local minimization algorithms that process information on the objective function and its derivatives at the sampling points. As a consequence, also local minimization cannot be accomplished in a finite number of steps.

The astonishing conclusion conflicts evidently with the enormous number of successful numerical results obtained by these optimization techniques. And so, this behavior should not be considered as a proof against Global Optimization. It only shows that the exact attainment of the global minimum is an excessive claim, at least when a priori information are not available.

The previous example could suggest that a constraint on the variations of the functional should solve the paradox. In particular, it could be natural to require that F or its derivatives [36–38] be a Lipschitz function with a known Lipschitz constant[‡]. However, it is worth noting that the simple a priori information expressed by a known upper bound on the maximum variations of F is ineffective [37].

[‡] We define the class of Lipschitz functions, $\mathcal{K}$ say, with Lipschitz constant κ as:

$$\{F : |F(\underline{x}_1) - F(\underline{x}_2)| \le \kappa\|\underline{x}_1 - \underline{x}_2\| \quad \underline{x}_1, \underline{x}_2 \in \Theta\}$$

In fact, given a Lipschitz monovariate objective function $F \mathbf{I} \mathbf{R} \rightarrow \mathbf{R}$ belonging to $\mathcal{K}$, no black box algorithm exists able to find x_* and prove its optimality in a finite number of function evaluations unless F has a saw tooth shape in a neighborhood of x_* [37]:

$$\exists \delta > 0 \, \mathbf{I} \quad F(x) = F(x_*) + \kappa |x - x_*| \quad \forall x \in [x_* - \delta, x_* + \delta] \tag{7}$$

Furthermore, even if (7) is fulfilled, the attainment of the optimum in a finite number of function evaluations is not guaranteed unless κ is known exactly, an unusual circumstance [37].

All these difficulties are circumvented if our claims are weakened. In fact, even if the data are known and managed with infinite precision, only e-optimal solution can be found in a finite number of function evaluations. Accordingly, it is widely assumed that a global problem is solved if, for some prescribed value of ε, a point in one of the following sets is found [21,39]:

$$\Lambda_x(\varepsilon) = \{\underline{x} \in \Lambda \, \mathbf{I} \|\underline{x} - \underline{x}_*\| < \varepsilon\} \tag{8a}$$

$$\Lambda_f(\varepsilon) = \|\underline{x} \in \Lambda \, \mathbf{I} \, |F(\underline{x}) - F(\underline{x}_*)| < \varepsilon\} \tag{8b}$$

which are certainly non void if F is a continuous function and Λ is a robust set [23].

2.3. When Global Optimization Is Required

From the definitions given in Section 2.1 the hardness of global optimization shows immediately. Obviously, the task becomes harder and harder depending on both the nature of F and the topological properties of the set Λ. And so, it is really useful to enlighten those aspects that, at least in principle, should be exploited in practice to obtain an efficient optimization algorithm.

The mathematical tool oriented to this analysis is Convex Analysis [26,85]. Even if developed in the convex and local optimization area, the underlying convex nature of non convex problems makes many of its concepts and results essential when investigating a wide class of global optimization problems [23,24–26].

Above all, it is of primary interest to point out the aspects making unimodal the problem.

Convex functions and convex feasibility sets play a relevant role [40]. In fact, the Convex Programming Problem, i.e. finding the minimum point of a convex function on a convex set, is the easiest to solve, since each local minimum of a convex function over a convex set can be easily proved also global[§] [20].

[§] It is easy to prove with a counterexample that this property does not keep if the convex hypothesis on Λ is relaxed.

On the other side, any non convex optimization problem can be reduced to a convex one by considering the convex hulls [20, 31].

Unfortunately, this result has essentially a theoretical life. The convex hull of a given function F, conf(F) later on, is generally hard to calculate, often as difficult as solving the original problem. On the other side, the use of the Legendre-Fenchel transform [32] is not very helpful, even if it can be evaluated via algorithms similar to the FFT ones [31]. Only in the particular case of quadratic objective functions subjected to linear constraints, the task becomes easier [41].

However, beside the required degree of smoothness, convex functions do not exhaust the whole class of functions whose local minima are also global. To clarify this point, let us consider the bell function B introduced in Subsection 2.2, restricted to the interval $[-1, 1]$. It is trivial to show that B (an indefinitely continuously differentiable function over $[-1, 1]$) has only one local minimum ($x = 0$) but it is not convex.

And so convexity is not mandatory when unimodality is required. Furthermore, it is worth stressing that, even if the objective function is convex, the optimization problem can be hard when Λ is not convex. To clarify this point let us observe that, given a continuous function F on $\mathbf{R}^n$ and a compact set Λ, the problem of finding $\arg\min\limits_{\Lambda} F$ is equivalent to find $\arg\min\limits_{(\underline{x},t)\in\Gamma} t$, where:

$$\Gamma = \{(\underline{x}, t) \in \Lambda \times \mathbf{R} \,|\, F(\underline{x}) - t \leq 0\| \tag{9}$$

Let us observe that the function to be optimized in the problem $\arg\min\limits_{(\underline{x},t)\in\Gamma} t$ is convex (linear). In other words, we have transformed a continuous optimization problem and converted the non convexity of the objective function into the non convexity of the constraint set. Despite the transformation, the resulting problem is, obviously, not less hard to solve. Accordingly, as remarked in the Subsection 2.1, it could not be always convenient to increase the topological complexity of Λ by incorporating any available information.

However, fixed the convexity of Λ, by introducing the concept of quasi-convex (strict quasi-convex [26]) function, we can enlarge the functional space wherein those aspects of convex functions that make them useful in the optimization area keep true. In fact, each local minimizer (strict local minimizer) of a strict quasi-convex (quasi-convex) function on a convex set can be easily proved also a global minimizer[‖].

[‖] The hypothesis of strict quasi-convexity cannot be relaxed into quasi-convexity without assuming the local minimizer a strict local minimizer. For instance, the bell function B is obviously quasiconvex and all the points in $R - [-1, 1]$ are only local minimizers.

Unfortunately, neither quasi-convexity allows to describe the whole set grouping all functions with only one local minimum (even if in a strict sense).

In the one dimensional case something more can be said on the relationship between the occurrence of local optima and the concepts of quasi-convexity and quasi-concavity [42]. In fact, given a continuous function F over the interval $[x_a, x_b]$, we have:

F has a finite number of locally optimal points $\Rightarrow$

$\Rightarrow \exists \delta > 0 | F$ is strictly quasiconvex or quasiconcave on any subinterval $[x_c, x_c + \delta]$ with

$$x_c \in [x_a, x_b - \delta].$$

In the multidimensional case, a similar result cannot be proved. In fact, in a neighborhood of a saddle point a multivariate function is not quasiconvex nor quasiconcave. And so, generally speaking, as stressed in [20], it is difficult to completely characterize the class of constrained problems which have a unique local minimum [43].

The theoretical investigation allows also to characterize and classify the optimization problems according to theoretical properties enjoyed by the objective function and the constraint functions. In particular, it will be enlightened the underlying convex nature of any optimization problem and the relationship between the "simplest" to be accomplished, and the one that would seem the hardest, since it do not meet any peculiar property.

Concerning the classification, we will refer here to the one proposed in [26], wherein Concave Minimization [20, 23, 26, 44], Reverse Convex Programming [23, 26], D.C. Programming [23, 26, 45] and Continuous Programming [26] are considered. Only the general aspects will be herein considered. For a detailed description and a rigorous definition specifying all the hypothesis to be considered, we refer the reader to [26].

We do not spend any word about convex problems, since they are not multiextremal, so that they can be solved by exploiting "simple" local optimization tools.

The next problem to be considered is Concave Minimization. The only aspect that distinguish this programming from the previous one resides in the hypothesis that F is assumed a concave function [40], instead of a convex function. Obviously, the considered problem can be relaxed into a more general one, if we refer to quasiconcave functions [26].

The third problem is the Reverse Convex Programming. It requires:

$$\text{Find} \arg\min_{\Delta \backslash \Lambda} F$$

where, Λ is a closed convex set in $\mathbf{R}^n$, Δ is the interior of the set $\{\underline{x} | C_i(\underline{x}) \leq 0\}$ and, as in Convex Programming, F and C_i are convex functions.

It appears evidently that this problem differs from Convex Programming because the inequalities considered in the definition of the effective constraint set are reversed.

As in the previous case, the hypothesis F convex can be relaxed into the hypothesis F quasiconvex [26].

Concerning the effective difference between the last two problems now considered, we note that it can be proved that quasiconcave minimization over a convex set is essentially equivalent to quasiconvex maximization over the complement of a convex set [26].

The last optimization problem enjoying some peculiar mathematical properties is the D.C. Programming [23, 45].

Let us remember that D.C. functions are functions expressible as the difference of convex functions.

As a consequence, a D.C. programming problem can be framed as follows:

$$\text{Find} \arg\min_{\Delta \cap \Lambda} F$$

where, Λ is a closed convex set in $\mathbf{R}^n$, $\Delta = \{\underline{x} | C_i(\underline{x}) \leq 0\}$ and F and C_i are D.C. functions.

It is worth noting that D.C. Programming includes a large variety of practical problems, since D.C. functions are very common. In fact, it can be proved that any continuous function can be expressed as the difference of two convex functions, even if the explicit decomposition is not always available.

Finally, the more general optimization problem that, at least in principle do not enjoy any peculiar theoretical property, is the Continuous Programming Problem.

However, it can be proved that any Continuous Programming Problem falls in the large class of D.C. Programming. Furthermore, it can be shown that any D.C. Problem can be reduced to a canonical D.C. programming problem, belonging to the class of Reverse Convex Programming Problems.

Accordingly, the above classification shows the underlying convex nature of any optimization problem. Unfortunately, this feature is not always obvious nor can be easily exploited to devise an efficient optimization algorithm.

3. GENERAL REMARKS ON GLOBAL OPTIMIZATION

In this Section we will discuss the general results available in the literature concerning the feasibility of global optimization, its complexity issues and the optimum search strategy.

3.1. Conditions for Global Optimality

To recognize the global optimal solution is a key point when affording a global optimization problem. Each algorithm is required to check if the feasible solution is effectively the searched solution and, when this is not the case, to transcend it, avoiding the stagnation in a false result. Accordingly, it is of real interest to know if there exist necessary and/or sufficient conditions in the global world corresponding to the well known local optimality conditions (see the Karush-Khun-Tuker conditions [46]) and to enlighten their role as theoretical and practical tools.

In the global world this “recognition” problem becomes quite hard. In fact, to be really useful, an optimality condition should satisfy some specific requirements. In particular, according to Strekalovsky [47]:

a) it should be an analytical condition involving only the data of the problem under investigation;

b) it should have a relation with the classical extremum theory;

c) it should reduce to solving a problem which is simpler (in some sense) than the original one;

d) it should posses the algorithmic property, i.e., the ability of constructing a new feasible point better then the incumbent one, when it does not satisfy the optimality conditions.

Several conditions for global optimality have been proposed.

Firstly, it is worth noting that global optimality can be proved by exploiting the well known necessary condition for local optimality $\nabla F = 0$ together with a check on the convex hull or by showing that all the points at the same level of the incumbent solution are local minimizer.

In other words the following relationships hold [31, 48]:

$$\underline{x}_0 \text{ is a global minimum of } F \Leftrightarrow \nabla F(\underline{x}_0) = 0 \text{ and } \mathrm{conv}\,(F)(\underline{x}_0) = F(\underline{x}_0)$$

$$\underline{x}_0 = \arg\min_{\Lambda} F \;\; \Leftrightarrow \;\; \underline{x} \in \Lambda | F(\underline{x}) = F(\underline{x}_0) \text{ is a local minimizer on } \Lambda$$

Unfortunately, these propositions are useful in the negative sense. The convex hull is a global concept that cannot be computed by locally sampling the objective functional (see Section 2.2). Accordingly, as pointed out by Hirriart-Urruty [31], the first condition has a practical application when a tight upper bound on the convex envelope is available. On its side, the second equivalence outlines the reason why procedures attempting to transcend the incumbent solution by looking for points wherein F remain constant should be expected to work [48, 49].

Integral conditions for global optimality refer to the general global optimization problem of continuous functions over compact sets, the results being obtained via a "convexification" process of the objective functional.

The first result in this area, well known in Probability and Statistics, probably dates back to Laplace. More recently new criteria have been addressed such as the Zheng's, Falk's [50] and Pinkus' [51] criteria. These results solve, in principle, the global optimization for an enormously wide class of problems. Again, their practical application is cumbersome, since the involved integrals must be evaluated by stable and efficient procedures. Anyway some applications have been proposed by Duong for polynomial optimization (referred by Tuy [45]) as well as by Glinkin and Sukharev [21].

Other conditions for Global Optimality are based on the concept of subdifferential, ε-subdifferential or on concepts in Convex Analysis such as polar sets, normal cones etc. [26, 31, 40, 47, 48].

In particular, by exploiting the concept of ε-subdifferential, conditions for D.C. problems can be stated. Anyway, even if the class of D.C. problems is quite general to contain the most part of practical problems, their easy practical application fails (the decomposition should be available).

This is a typical characteristic of the cited global optimality conditions. As pointed out in [31], general accepted conditions for global optimality does not exist. Hirriart-Urruty stresses in [31] that no practical conditions can be derived for the general continuous optimization problem. He observes that, even in the restrict area of quadratic optimization, unsolved problems and non well understood situations still remain [48].

However, these troubles should not raise objections against the practical feasibility of Global Optimization. All these theoretical difficulties comes out when generality is pursued. As we will see in the next section also the general purpose best algorithm does not exist. And this behavior does not affect only global optimization. Also the familiar local checks, often based on the widespread analytic tools of differential calculus and involving also derivatives of higher order [48], cannot be easily accomplished when general objective functions are involved [48, 52].

3.2. The Best Algorithm

In Subsection 2.1 we stated that the feasible goal for global optimization is to find, in a finite number of function evaluation, a point in one of the two sets (8a) and (8b). However, it is easy to prove that the task of finding a point in one of the two sets in Eqs. (8) can be solved in a finite number of function evaluations.

To this end, it is sufficient to consider the rough grid search with a properly chosen dimension of meshes.

Obviously, this method is not efficient because it determines the sampling points without taking into account the previous outcomes.

Algorithms like this, sampling at present without considering the previously observed values, are called passive. In contrast, all algorithms establishing the next evaluation point by processing the information gathered in previous observations are called sequential.

At first sight it seems natural to believe that sequential algorithms should outperform passive algorithms. Unfortunately this is not completely true and the following result clarifies this point. Let us consider the class of Lipschitz functions $\mathcal{K}$ and let us denote with $x_*(f, \mathbb{A}, k)$ the solution provided after k function evaluations, by the algorithm $\mathbb{A}$ when optimizing F.

When the algorithm $\mathbb{A}$ is used to optimize the functions in $\mathcal{K}$, its performances can be evaluated by considering the result relative to the worst case, i.e.,

$$\max_{F\in\mathcal{K}} \quad \|x_*(F, \mathsf{A}, k) - x_*\| \tag{10}$$

Obviously, the best algorithm on $\mathcal{K}$ is the one that minimizes the merit parameter in Eq. (10). The following proposition, due to Sukharev, allows to compare the best sequential algorithm with the best passive algorithm [21]:

$$\begin{aligned}\min_{\mathsf{A}\text{ passive algorithm}} \max_{F\in\mathcal{K}} \|x_*(F, \mathsf{A}, k) - x_*\| &= \\ \min_{\mathsf{A}\text{ sequential algorithm}} \max_{F\in\mathcal{K}} \|x_*(F, \mathsf{A}, k) - x_*\| &\end{aligned} \tag{11}$$

This proposition leads to an astonishing consequences: the best passive algorithm has the same performances of the best sequential algorithm on $\mathcal{K}$.

As suggested by Torn and Zilinskas, the game theory can be used to better be convinced. In fact, as long as the algorithm explores the searching space to locate the optimum, an ideal opponent can change the function $F \in \mathcal{K}$, by substituting it with another element of $\mathcal{K}$ that has the same values at the sampled points.

Again, this dramatic result should not deject attempting to finding an efficient way to solve a global optimization problem.

The Sukarev's Theorem discusses only what happens in the worst case, assuming that an ideal opponent exists who knows the searching strategy of the algorithm [21].

Now, let us turn our attention to some other theoretical results, known as the No Free Lunch Theorems (NFL) [53], which provide more insight in choosing the strategy to get a successful algorithm for the problem at hand.

NFL are a set of results found by Wolpert and Macready during their stay at the Santa Fe Institute [54] and relative to stationary as well as to time varying optimization problems. They explore the relationship between an efficient optimization algorithm and the problem that it is asked to solve.

As usual, the optimized function is assumed to be available to the algorithm in a black box form and a probabilistic approach is considered.

These results state that a general purpose algorithm that can be efficiently used in all optimization problems does not exist: on average, the performances of any two optimization algorithms are the same across all the possible optimization problems [53]. In other words, strategies which work well over a set of optimization problems, will necessary work bad on some of the remaining ones. Accordingly, the optimizer cannot be chosen by blindly referring to results obtained in other application areas neither results on canonical problems are of great interest when a complex problem must be solved in practice.

Inserting some a priori information on the structure of the problem at hand should improve the performances.

It is worth noting that the NFL do not exclude that some algorithms which do not take explicitly into account the structure of the problem can, nevertheless, work well in practice. The geometric interpretation of the NFL suggests that an algorithm can be "aligned" to the structure of the problem because of an implicit tuning procedure due to training and/or to the years of research [53].

It must be noted that, although general theoretical results, as the ones cited above, allows finding the main guidelines to compare two algorithms, we also need criteria allowing a practical comparison of optimization algorithms. Unfortunately, as enlighten by the NFL, this is a quite difficult task requiring the introduction of suitable "quality" parameters.

According to Torn and Zilinskas [21], two parameters are generally considered as quality factors: the number of function evaluations and the total computing time.

However, "in all methods there exist a number of heuristic parameters for which explicit values have to be chosen. When this is done by solving the set of standard problems, the methods are calibrated to efficiently solve these particular problems. Because the tests problem are rather arbitrarily chosen and easy to solve, it is not clear that equal success will occur when solving other problems without again modifying the methods" [21] (see the No Free Lunch Theorems). "Important information about the efficiency and the applicability of the algorithms come from solving real world problems".

The first sentence of this note allow us to briefly introduce the role of heuristic tricks in optimization algorithm. Torn and Zilinskas refer that "it is natural that often, in applications, heuristic methods are used", since "solving practical problems may be considered somewhat of an art" [21].

Concerning this point, Ratschek and Rokne [55] believe that, although choosing something at hand "is familiar to everybody who is working with computational mathematics, it is still a taboo in wide mathematical circles which are dreaming of perfect algorithms that may overcome every numerical problem".

In [55] they also refer that "a computer can be much more powerful than otherwise if the brain of a mathematician is allowed as one peripheral" and that "on line working is the most convenient medium for this type of work".

For our part, we believe that the complexity of the problem naturally leads to heuristic methods unless new theoretical tools will be devised to (more or less) definitely clarify the open questions either in a positive or in the negative sense [54].

3.3. Complexity Issues

As stressed in the introductory Section, the high dimensionality of the problem is the main obstacle towards effective Global Optimization. The amount of computing time required to get a reliable solution becomes inordinate as long as the number of variables increase [56, 57]. Many optimization algorithms have good performances when few variables are involved, but become impractical when dealing with tens of variables, unless heuristic tricks are introduced to speed up convergence.

In particular, the dependence of the performances of a global optimization algorithm versus the dimension of the searching space can be rigorously established.

As a matter of fact, let us quote the theorem due to Nemirovsky and Yudin [57] concerning the optimization problem in the functional space, say $\mathcal{F}$, of k-differentiable functions over $[0, 1]^n$, with bounded derivatives in the sense that:

$$\left| \frac{d^k}{dt^k} F(\underline{x} + t\underline{e}) \right| \leq \kappa \qquad \forall \underline{x} \in [-1, 1]^n \tag{12}$$

$\underline{e}$ being any unit vector.

They prove that each optimization algorithm that is guaranteed to find a point in one of the sets (8), for a given ε, by sampling the objective function and its derivatives, requires a number of steps proportional to $(\kappa/\varepsilon)^{\frac{n}{k}}$ for at least one function in $\mathcal{F}$.

The theoretical result clearly states the exponential dependence of the number of minimization steps on the dimensionality of the problem. It also shows that the smoothness hypothesis has not significant benefits: it affects the exponential dependence, but it does not turn the exponential complexity into a polynomial one.

However, this result should not be interpreted as the further proof that Global Optimization is a too hard task to have practical applications. Also Local Optimization presents, as shown by Murty and Kabadi [52], similar complexity issues.

As a concluding remark, it is worth stressing that multivariate optimization can be effectively afforded only by designing new algorithms, without attempting to leading back the problem to the univariate case.

In fact, given a multivariate objective function F on an interval $\xi = [x_{1a}, x_{1b}] \times \ldots \times [x_{na}, x_{nb}]$, it can be optimized by solving the sequence of one dimensional problems by fixing $n - 1$ variables:

$$\min_{\xi} F = \min_{[x_{1a}, x_{1b}]} \left[\ldots \min_{[x_{na}, x_{nb}]} F \right] \tag{13}$$

Obviously the minima of each single step can be evaluated by exploiting any one-dimensional minimization algorithm. Unfortunately, the complexity of this approach, firstly proposed by Piyavskii, make it quite difficult, since the number of one dimensional optimization problems grows rapidly with the dimension of the problem. As a matter of fact, few computational results based on this approach have published till now [58].

Neither the approach proposed by Strogin and Butz, relying on the mathematical properties of the Peano's curve [42], is successful. In this case the basic idea is to solve a one dimensional problem on a curve that essentially fills the searching domain, according to the required accuracy and the maximum variations of the objective function. However the one dimensional optimization becomes a hard task, since points of the searching domain close to each other are mapped into points which can be also very far on the filling curve. Accordingly, the univariate function to be optimized can present many local minima very similar to the global one.

4. THE ALGORITHM

In this section we discuss the way that led us to choose a particular algorithm for the practical problem of concern in the Part II of the paper, i.e., the diagnosis and the synthesis of reflector antennas.

Obviously, such considerations are completely general and define the general strategy that should be followed, in any application of global optimization, to practical problems.

4.1. Deterministic and Stochastic Approaches

We stressed in the previous Sections that global optimization is generally required when we deal with the optimization of multiextremal functions, and no manipulations appear evident to simplify the problem.

The problem we have to face in Part II requires a continuous optimization algorithm of high dimensionality.

Unfortunately, a large number of unknown is involved and no useful assumptions can be drawn to be explicitly used at the designing stage of the optimization algorithm. As we stressed, it must be expected that high dimensional problems are the most difficult to manage. This general trend is effectively confirmed in practice, being dimensionality the first obstacle toward efficiency and reliability of a great deal of optimization problems.

The choice of the optimization algorithm becomes quite difficult since the best general purpose algorithm does not exist. We stressed that efficiency is obtained only if some information on the problem at hand is inserted explicitly or implicitly in the optimization algorithm.

On the other side, many algorithm have been devised and some among them enjoy interesting theoretical properties.

They are generally classified into deterministic and stochastic approaches, depending on whether they incorporate some stochastic elements [21, 23, 24, 26, 39].

First of all, it must be noted that a general feature leads to a typical working strategy shared by many approaches. Let us remember that the key point in any global optimization procedure is the problem of transcending the incumbent solution [26]: to check the ε-optimality of the obtained feasible solution and, when the goal is not fulfilled, to find a new point whose level is not higher than the current one.

Accordingly, global algorithms subdivide the searching process into two phases:

the local phase: given a point in the searching space, the goal of the local phase is to search for a better new point. Obviously, this task can be accomplished by using local optimization tools and/or non deterministic techniques.

the global phase: the goal of the global phase is to transcend the incumbent solution, providing a better estimate of the solution from which restarting the local phase. At this stage the global optimality conditions we briefly discussed above, assume a very relevant role.

Deterministic methods are promising optimization tool when well defined analytical properties of the objective functional are available. Many specialized algorithms have been devised, dealing with quadratic, bilinear, concave, fractional and D.C. problems [20, 23, 24, 26]. On the other side, algorithms for continuous optimization, such has the branch and select techniques, the relief indicator technique, the Lipschitz optimization procedures, the generalized descent algorithms, have been developed only in the last three decades and the attempts to solve problems with dimensionality higher than two have begun only in recent years. Unfortunately, when the number of unknowns grows, this techniques require very large computing resources and some heuristic tricks (with the consequent lack in theoretical rigor) to skip the most time consuming searching steps.

On the other side, besides the non unanimous consent, Stochastic Algorithms are retained in [39] the most efficient methods for finding the global minimum of an objective function and by Rastrigin [21] are even considered the only efficient approach to global optimization. Anyway, stochastic elements in optimization algorithms can be important building blocks of efficient global minimization techniques and their appeal often depends on an easy numerical implementation.

Obviously, in contrast with deterministic approaches, the absolute guarantee of success is missed. However, the convergence properties of this class of algorithms can be analyzed in a probabilistic sense. In particular, asymptotic convergence can be considered, i.e., it can be required that, according to a probabilistic criterion, the minimizing sequence generated by the algorithm converges to the global optimum as long as the number of iterations grows to infinity. On the other side, to state the performances of the searching procedure, the asymptotic convergence of an optimization algorithm is simply a necessary condition to be considered.

In fact, even the simplest and unsatisfactory Pure Random Search approach, that samples the feasible domain with a sequence of independent and uniformly distributed points, meet the asymptotic convergence to an ε-optimal point. This result can be immediately proved by observing that, since the probability of getting a point in $\Lambda_x(\varepsilon)$ after k samples is given by:

$$Pr\{\exists m \in \{1,\ldots,k\} | \underline{x}_m \in \Lambda_x(\varepsilon)\} = 1 - \left(1 - \frac{\mathbb{M}(\Lambda_x(\varepsilon))}{\mathbb{M}(\Lambda)}\right)^k \tag{14}$$

the probability of success goes to one as k goes to infinity, as long as the Lebesgue measure of $\Lambda_x(\varepsilon)$, say $\mathbb{M}(\Lambda_x(\varepsilon))$, is greater than zero.

Moreover, the simplest deterministic search technique, the Grid Search, converges, even in a finite number of function evaluations, to a point of ξ_ε.

On the other side, at the first sight, no reasons should make a random sampling covering preferable to a simple Grid Search.

However, firstly, it has been proved by Anderssen and Bloominfield that, according to a probabilistic criterion, Pure Random Search provides a covering of the feasible region more efficient then Grid Search, at least when dealing with high dimensional problems [59]. Secondly, finite convergence is quite useless if the computational effort exceeds the available resources. As a matter of fact, only an upper bound on the needed computational complexity is of real interest to estimate the algorithm effectiveness.

On its side, Pure Random Search has a natural predisposition to become an Adaptive Search Algorithm, the basic building block of really efficient algorithms, enjoying surprisingly interesting convergence properties.

The first straightforward modification of Pure Random Search naturally leads to Multistart [39].

Its basic idea is very simple: the global phase randomly explores the feasible domain, looking for promising regions, but without neglecting any part of it, while the neighborhood of each sampling point is inspected efficiently by using Local Search (LS). The local phase processes the information gathered in the global step to refine the data quickly and to reject unfruitful sampling points.

The Multistart Algorithm in its canonical formulation at the k-th iteration works as follows:

1) take a sample, $\underline{x}$ say, randomly and uniformly in the feasible region Λ;
2) apply a local search to $\underline{x}$, getting a point $\hat{\underline{x}}$;
3) if $F(\hat{\underline{x}}) < f_k$ then $\underline{x}_{k+1} = \hat{\underline{x}}$ and $f_{k+1} = F(\hat{\underline{x}})$, otherwise $\underline{x}_{k+1} = \underline{x}_k$ and $f_{k+1} = F(\underline{x}_k)$.

However, two reasons make Multistart inefficient.

Firstly, the random sampling of Λ has no memory. The uniform distribution, even if very useful from a theoretical point of view to easily prove convergence properties, samples without accounting for the searching history, without adapting itself to the previous outcomes and to the iteration number.

Secondly, the LS, the most time consuming step, is started blindly. Accordingly, computing time is wasted when it is applied to points lying in the attraction region of the same local minimum.

To overcome these drawbacks, two different classes of algorithms have been developed: the Random Search Technique [21, 60, 61] and the Clustering Methods [21, 62, 63].

The Random Search Techniques are the evolution of Pure Random Search. The random sampling is improved by making it adaptive, by considering variable probability sampling distributions that depend on the iteration number and previous outcomes. These algorithms enjoy very attractive theoretical properties and their practical implementations leads naturally to Random Walk based approaches [64].

A first approach to the problem leads to Pure Adaptive Search Algorithm [64, 65]. In this case the searching procedure samples uniformly only the region of Λ where the objective function F assumes values better that the current one, say $\Lambda^{(k)}$, being k the current iteration number.

It can be proved that Pure Adaptive Search requires a bounded number of steps to get a solution with a prefixed accuracy. In other words, the expected performance is bounded. Furthermore, the bound grows only linearly with the dimension of the problem (this seems to contradict the result by Nemirovski and Yudin).

Unfortunately, this algorithm is only of conceptual nature as clearly stressed in [60].

Pure Adaptive Search is quite difficult to be practically implemented, since it is not available an efficient method to estimate $\Lambda^{(k)}$. In particular, Boender and Romeijn note that the inefficient implementation of Pure Adaptive Search based on an acceptance and rejection mechanism, leads to an algorithm whose complexity grows exponentially with the dimensionality of the problem [60].

However, this is not the point of main interest here. The interesting point is that Pure Adaptive Search can be thought as a random algorithm with a variable sampling probability distribution on $\mathbf{R}^n$.

Accordingly, rather than evaluating $\Lambda^{(k)}$ and uniformly sampling it, a sampling distribution varying with the iteration number can be considered.

This approach leads to Adaptive Search Algorithms [60].

These algorithms enjoy the same theoretical properties of the Pure Adaptive Search Algorithm.

In particular, it has been proved by Romeijn and Smith [30] that the expected number of function evaluations required to get the solution with a given accuracy is not greater than that of Pure Adaptive Search.

Obviously, the problem is to define and implement the sampling according with the probability density distribution as a function of k.

Both problems can be solved and Markov Chain based algorithms can be effectively used to approximate these distributions. In particular, many algorithms have been devised to this end. They approximate the desired distribution as the limiting distribution of the chain.

Immediately, this approach leads to algorithms such as Simulated Annealing and Evolutionary Algorithms [22, 25, 60, 66, 67].

The clustering techniques attempt to avoid starting LS uselessly. The basic idea is to cluster (according to a clustering criterion) the points provided by the global phase and assume that points belonging to the same group lies in the attraction region of the same minimum. Accordingly, only one LS can be started in each group. Furthermore, previously determined local minima can be memorized and used as seed points avoiding to start LS in the corresponding cluster.

When properly structured, clustering techniques enjoy very attractive theoretical properties. In particular, it can be proved that, for the Single Linkage Algorithm [62], the probability of starting LS goes to zero as the number of iteration goes to infinity and that, if the parameters of the algorithm are properly set, the total number of invoked LS remains finite even if the procedure continues forever.

It is evident the relevance of such properties. They state that the algorithm starts LS efficiently.

Unfortunately, the guarantee of success of the rough Multistart is lost. In particular, both methods are affected by a drawback: the possibility that each cluster contains more than one attraction region. Accordingly, by starting only one LS for each cluster, some local optimum (which could be also global) may be missed.

The main reason of this lack of guaranteed convergence resides on the fact that the approaches do not use at all the values of the objective function when clustering.

Multilevel methods have been proposed to circumvent this difficulty and restore the convergence to the solution [63]. They are proved to converge to the global optimum as Multistart, but are also proved to invoke the local search only one time, with probability equal to one, in each region of attraction.

As above mentioned, this algorithm shares with single Linkage its strong theoretical properties we presented. Furthermore, it can be proved that it is able to find all local minima of F in a finite number of iterations with probability equal to one [63].

As a concluding remark, it must be noted that these clustering algorithms, even if considered very efficient, also for practical applications, may fail to find the global minimum when stopped too early (when computing resources have been used out) [63]. Furthermore their performances, as usual in global optimization, depend on the dimensionality of the problem [63]. In particular, it must be stressed that the performances of the algorithms strongly depend on the amount of available computing resources. As a consequence, when the adopted algorithm can be easily parallelized, the use of parallel computing machine becomes quite convenient [68].

A suggested way to improve the performances is to change the sampling probability distribution, adapting it to the objective function (i.e., to the history of the search) [63]. In particular, it could be useful to consider advanced random search procedures (see above) as a global phase.

4.2. Why Evolutionary Algorithms as the Global Phase?

In the previous subsections it has been noted that the performances of stochastic algorithms can be greatly improved by adopting two different strategies.

The first one smartly interweaves the global random step with an effective local search procedure to speed up convergence without wasting computing time in unpromising regions of the searching space.

The second one adopts advanced stochastic search techniques exploiting the information gathered from previous outcomes to smartly sampling the searching space and adapting itself to the problem at hand. It has been observed that practical implementation of algorithms that samples, with memory, the searching space are random walk techniques such as Evolutionary Algorithms and Simulated Annealing Algorithms.

Obviously, as suggested in [63], better results should be expected by combing together both these approaches. For our part we choose an Evolutionary Algorithm as the global phase of the optimization algorithm [25].

Evolutionary Algorithms are all the approaches that arise from the analogy with the Theory of Evolution, a very wide class (to which also Simulated Annealing can be reduced [69]) containing techniques that, in practice, can be quite different from each other, sharing only the general inspiring concepts. Anyway, their strict relationship allows to define a prototype algorithm to which all algorithms can be led back when the search modules are properly specified [70–72].

It is worth noting that usually the objective function to be optimized in the Evolutionary Computing area is called fitness and is related to a maximization process instead of a minimization one. To this end we introduce the fitness function G, properly related to F and the sets:

$$\Psi_x(\varepsilon) = \{\underline{x} \in \Lambda \,|\, \|\underline{x} - \underline{x}^*\| < \varepsilon\| \tag{15a}$$

$$\Psi_g(\varepsilon) = \{\underline{x} \in \Lambda \,|\, |(G(\underline{x}) - G(\underline{x}^*)| < \varepsilon\} \tag{15b}$$

Generally speaking, Evolutionary Algorithms are subdivided into three large subclasses: Genetic Algorithms, Evolution Strategies and Evolutionary Programming [70–76]. The distinction has essentially an historical origin. Genetic Algorithms effectively were ideated as a formal tool to study the complex process of adaptation [25, 71, 75–78]. The other two approaches, Evolution Strategies and Evolutionary Programming, since their birth, were used to solve practical problems, as a new way to Optimization and Artificial Intelligence, respectively [71, 79].

Nowadays, many aspects which characterized each subclass are adopted in the same algorithm so that a well defined classification is not till possible.

In the last two decades, Evolutionary Computing has proved a very attractive and effective optimization tool whose number of successful practical applications has grown impressively. On the other side, the attempts to explain, in a theoretical framework, the dynamical behavior of the searching process as well as the reasons of its success in solving real world problems [25, 80–89] are multiplying.

In the following we will focus our attention only on two theoretical aspects of Evolutionary Algorithms: the asymptotic convergence and the finite convergence.

In contrast with a widespread idea in the practitioner community, as other stochastic approaches, Evolutionary Algorithms, when properly structured, enjoy the asymptotic convergence.

This property, even if crucial to state the effectiveness of the algorithm (before the proof appeared, Evolutionary Algorithms were frequently criticized when compared to those approaches whose asymptotic convergence was already proved [90]), is of limited interest both from a theoretical and practical point of view.

From one side also the simplest Search Algorithm, the pure random search, converges asymptotically to the Global Optimum as stated above. Secondly, since in real world applications infinite computational resources are not available, the asymptotic convergence does not tell anything about the efficiency of the searching procedure, in particular on the expected number of iterations required by the algorithm to attain an ε-optimal solution.

Accordingly, the most appealing property for an optimization algorithm is its ability of finding an ε-optimal solution in a finite number of (expected) iterations whose upper bound can be expressed in a closed form as a function of the key feature of the problem at hand.

Concerning this point, Romeijn and Smith have provided [30, 67] an upper bound to the number of iterations required by the Adaptive Search (AS) algorithm to find an ε-optimal solution of Lipschitz functions over a convex set. Furthermore, [30] shows that a practical implementation of AS can be approximately obtained by Simulated Annealing algorithms with a properly structured cooling schedule.

In this paper we provide an analogous result for an idealized version of a simplified Evolutionary Algorithm.

In particular, we prove, in Appendix A, that the number of expected iterations required to such an algorithm is, as for the AS algorithm, stochastically less than that of the Pure Adaptive Search (PAS) algorithm.

This result reassesses a sort of equivalence, at least from a theoretical point of view, between SA and EA and, moreover, provide a further explanation of the effectiveness of EA.

4.3. The Hybrid Evolutionary Algorithm

In the previous Section we have shown the reasons which led us to adopt an Hybrid Evolutionary Algorithm. Let us now describe explicitly the proposed algorithm.

The first problem which must be faced when constructing an Evolutionary Algorithm relies on how to code the unknowns.

Furthermore, in contrast with a widespread idea, a particular cardinality of the alphabet does not offer any intrinsic advantage. In other words, the well known Holland's rule [75], which suggests to reduce the cardinality to enhance the implicit parallelism of the algorithm, and, consequently, its performances, is not well founded [91–93].

In fact, a recent result by Fogel and Ghoezil shows clearly that, regardless of the cardinality of the representation, different algorithms can be constructed with the same performance if the reproduction operators are suitably chosen [91].

This conclusion not only perfectly agrees with the NFL but also explains the empirical results obtained in the latest years, which shows that real coding of continuous problems generally works better then binary coding [93].

Let us now discuss the hybrid scheme.

Different approaches can be followed when combining a global optimizer and a LS. In this paper we devote our attention to two schemes which are suggested by the analogy with the Theory of Evolution: the Lamarckian and the Darwinian [94].

The underlying concept is "life". During life the individual demonstrates the potential of its genetic inheritance with respect to the environment. In a hybrid algorithm "life" is LS which allows to efficiently explore the region of the fitness landscape in an individual's neighborhood. The Lamarckian and Darwinian approaches distinguish from each other depending on if the chromosomal inheritance changes during "life" or not, respectively.

To describe this mechanism with more details, we will introduce the following formal framework.

Let us first denote with k the population cardinality and with $\Pi_p = \{\underline{c}_{1,p}, \ldots, \underline{c}_{k,p}\}$ the population at the p-th generation.

Let $\mathcal{H}$ be the operator:

$$\mathcal{H} \,|\, \Pi_p \in \mathbf{R}^{n*k} \to \underline{g} = (g_1, \ldots, g_k) \in \mathbf{R}^k \tag{16}$$

which associate to each population the fitness vector of its elements.

Let $\mathcal{S}$, $\mathcal{R}$, $\mathcal{M}$ be the reproduction operators, i.e., the selection, the recombination and the mutation operators respectively:

$$\mathsf{S}\Big| \; (\Pi, g) \in \mathbf{R}^n \times \mathbf{R}^k \times \mathbf{R}^n \quad \to \quad \Pi' \in \mathbf{R}^n \times \mathbf{R}^k \tag{17a}$$

$$\mathsf{R}\Big| \; \Pi' \in \mathbf{R}^n \times \mathbf{R}^k \quad \to \quad \Pi'' \in \mathbf{R}^n \times \mathbf{R}^k \tag{17b}$$

$$\mathsf{M}\Big| \; \Pi'' \in \mathbf{R}^n \times \mathbf{R}^k \quad \to \quad \Pi''' \in \mathbf{R}^n \times \mathbf{R}^k \tag{17c}$$

Let us denote with $\mathcal{L}\mathfrak{s}$ the LS operator, i.e., the operator which produce a new population by applying the LS q times with starting points equal to the individuals in the current population:

$$\mathsf{Ls}\Big| \; \Pi \in \mathbf{R}^n \times \mathbf{R}^k \to \Pi' \in \mathbf{R}^n \times \mathbf{R}^k \tag{18}$$

And so, by denoting with the symbol "$\circ$" the product of two operators, a generation of a Darwinian Algorithm is structured as:

$$\Pi_{p+1} = \mathsf{M} \circ \mathsf{R} \circ \mathsf{S}\Big(\Pi_p, \mathsf{H}(\mathsf{Ls}(\Pi_p))\Big) \tag{19}$$

while a generation of a Lamarckian Algorithm is structured as:

$$\Pi_{p+1} = \mathsf{M} \circ \mathsf{R} \circ \mathsf{S}\Big(\mathsf{Ls}(\Pi_p), \mathsf{H}(\mathsf{Ls}\ (\Pi_p))\Big) \tag{20}$$

Even if the Darwinian approach seems to be more robust, we have considered the other one because of its faster convergence rate [94].

Furthermore, as widely discussed in the previous Sections, since LS is a time consuming step in the procedure, it must be invoked as rarely as possible. Accordingly, it has not been applied to each individual of the population, but only to a fraction of it, according to a simple clustering criterion choosing the best fit individuals.

Now, let us describe the reproduction operators.

In a Hybrid Evolutionary Algorithm the role of the Evolutionary Algorithm is essentially to explore the searching space and locate the more promising regions. Accordingly, high selection pressures [95] even if is able to speed up the convergence rate, can induce high probable premature convergence to false solutions. And so, a roulette wheel proportional selection with elitist [96] has been considered.

In order to get a non negative fitness function and to control the selection pressure, a "quasi-linear" scaling of the objective functional has been adopted.

The quasi-linear scaling realizes the following non increasing correspondence between the objective functional and the fitness function

$$G = \begin{cases} aF + b & \text{for } aF + b \geq 0 \\ 0 & \text{for } aF + b < 0 \end{cases} \tag{21}$$

where the constants a and b are chosen in order to fix the mean value and get $\max(G) = h^* \min(F)$, where $h > 1$ is a scaling parameter. In Authors' experience this scaling strategy works better then more standard ones such as:

$$G = c/F; \tag{22a}$$

$$G = f_0 - F \tag{22b}$$

wherein c and f_0 are suitably chosen constants.

It must be stressed that elitist must be introduced when dealing with the proportional selection mechanisms to retain the asymptotic convergence to the global minimum [88]. Furthermore, the elitist strategy ensures, essentially, that the performance of a hybrid algorithm cannot be worse than the one of the LS when the starting point of LS is inserted in the initial population.

Concerning the mutation scheme, a uniform mutation on the searching space has been considered. It is worth noting that normal mutation distribution is more usual [97]. The central limit theorem and, again, the analogy with the Theory of Evolution are generally invoked to justify this predominant choice in real coded EA's, even if the Gaussian distribution is not the only limit distribution of a sum of random variables [98]. In nature, in fact, mutation is determined by different physical and chemical causes which interact in a very complex way. Accordingly, mutation can be interpreted as the overall effect, the sum, of a sequence of independent random variables.

However, in our experience, the uniform mutation scheme demonstrated better performance then the gaussian one. This behavior can be explained by observing that the explorative features of the global phase must be enhanced in a hybrid scheme.

Concerning the recombination operator, an extended intermediate recombination has been considered [99].

The last critical point to be faced when implementing an EA is the parameter setting. Obviously, only heuristics rules are available to determine the population size and the parameters of reproduction operators [100]. Furthermore also self adaptation of the parameters has been considered [101–103].

It must be stressed that, by enlarging the population size, the performances of the algorithm does not necessarily improve. In fact, as pointed out in [104], an efficient algorithm can be obtained only by trading off between an accurate search space sampling and the amount of available computing resources.

We will not discuss this point further in this paper, and we refer the reader to the review papers already referenced.

5. CONCLUSIONS

The problems involved in global optimization have been critically discussed, by referring to the main results recently presented in the literature.

The basic concepts and tools, that let us to select a particular approach, a hybridized evolutionary algorithm, in solving two relevant problems in applied electromagnetics have been enlightened. The proof of the convergence property of an idealized version of a simplified evolutionary algorithm has been provided.

APPENDIX A.

Following [81, 82], an EA algorithm with an infinite population and characterized by only a proportional selection reduces to a stochastic process $\{\mathcal{X}_k\}_{k \in \mathsf{N}}$ defined by the sequence of probability distributions

$\{\mathbb{P}_k\}_{k\in\mathsf{N}}$ over the searching space Λ and the corresponding sequence of probability cumulative distribution functions $\{P_k\}_{k\in\mathsf{N}}$.

The stochastic process $\{\mathcal{X}_k\}_{k\in\mathsf{N}}$ defines a sequence of random variables. $\{G(\mathcal{X}_k) = \mathcal{G}_k\}_{k\in\mathsf{N}_0}$ ($\mathcal{G}_0 = g_*$ is assumed).

According to [105], an index k is called a record if

$$\mathcal{G}_k = \max\{\mathcal{G}_0, \ldots, \mathcal{G}_k\} \tag{A1}$$

and the sequence of record values $\{r_k\}_{k\in\mathsf{N}_0}$ defines the time indexes corresponding to an increasing values of $\{\mathcal{G}_k\}_{k\in N_0}$.

Let us now introduce explicitly the PAS algorithm.

The PAS is the simplest evolution of Pure Random Search. While PRS samples uniformly the whole search space, PAS attempts to improve at each step the current feasible solution by sampling, uniformly, as PRS, the set of the points of Λ corresponding to functional levels higher that the current one.

Formally, PAS is defined by the following iterative searching procedure:

1) $k = 0;\ w_0 = g_*$
2) Generate x_{k+1} uniformly over $\{x \in \Lambda \,|\, G(x) > w_k\}$
3) Set $w_{k+1} = G(x_{k+1}),\ k = k + 1$ and go to 2)

The above procedure defines a stochastic process $\mathcal{W}_k$ that, according to the definition in [105], stochastically dominates $\mathcal{G}_{r_k}$, a key property for our purposes. In fact, [65] provides an upper bound to the expected number of iteration required by PAS to get an ε-optimal solution of an optimization problem of a Lipshitzian function on a convex bounded set.

However, to demonstrate the above mentioned result, we need to prove the preliminary proposition:

A) $Pr\{\mathcal{G}_{r_{k+1}} > u_2 | \mathcal{G}_{r_k} = u_1\} \geq Pr\{\mathcal{W}_{k+1} > u_2 | \mathcal{W}_k = u_1\}$

Let us evaluate the conditional probability distributions:

$$Pr\{\mathcal{G}_{r_{k+1}} > u_2 | \mathcal{G}_{r_k} = u_1\} \tag{A2a}$$

$$Pr\{\mathcal{W}_{k+1} > u_2 | \mathcal{W}_{r_k} = u_1\} \tag{A2b}$$

We have:

$$Pr\{\mathcal{W}_{k+1} > u_2 | \mathcal{W}_k = u_1\} = \begin{cases} \dfrac{\mathbb{M}\left(\Psi_g(g^* - u_2)\right)}{\mathbb{M}\left(\Psi_g(g^* - u_1)\right)} & u_1 < u_2 \\ 1 & \text{otherwise} \end{cases} \tag{A3}$$

Similarly, it results that:

$$Pr\{\mathcal{G}_{r_{k+1}} > u_2 | \mathcal{G}_{\tau_{k+1}} = u_1\} = \begin{cases} \dfrac{\mathbb{P}_{k+1}\left(\Psi_g(g^* - u_2)\right)}{\mathbb{P}_{k+1}\left(\Psi_g(g^* - u_1)\right)} & u_1 < u_2 \\ 1 & \text{otherwise} \end{cases} \tag{A4}$$

From the very definition of stochastic dominance, A) follows immediately once it is proved that:

$$Pr\{\mathcal{G}_{r_{k+1}} > u_2 | \mathcal{G}_{r_k} = u_1\} \geq Pr\{\mathcal{W}_{k+1} > u_2 | \mathcal{W}_k = u_1\} \quad \forall u_1, u_2 \tag{A5}$$

The result is obvious for $u_1 \geq u_2$. To prove the assertion in the case $u_1 < u_2$, let us introduce the function:

$$H(g) = \frac{\mathbb{M}\left(\Psi_g(g^* - u_1)\right)}{\mathbb{P}_{r_{k+1}}\left(\Psi_g(g^* - u_1)\right)} a_{r_{k+1}} g^{r_{k+1}} \tag{A6}$$

where $a_{r_{k+1}}$ is a sequence defined by the probability distribution of the EA searching process [81, 82].

Let us firstly assume that $H(u_2) \geq 1$. We have:

$$Pr\{\mathcal{G}_{r_{k+1}} > u_2 | \mathcal{G}_{r_k} = u_1\} = \frac{1}{\mathbb{M}(\Psi_g(g^* - u_1))} \int_{\Psi_g(g^* - u_2)} H \circ G(\underline{x}) d\underline{x} \tag{A7}$$

Since H is an increasing function it results that:

$$\begin{aligned} Pr\{\mathcal{G}_{r_{k+1}} > u_2 | \mathcal{G}_{r_k} \geq u_1\} &\geq \frac{1}{\mathbb{M}(\Lambda(g^* - u_1))} \int_{\Psi_g(g^* - u_2)} d\underline{x} \\ &= Pr\{\mathcal{W}_{k+1} > u_2 | \mathcal{W}_k = u_1\} \end{aligned} \tag{A8}$$

On the other side, if $H(u_2) < 1$,

$$H(y) < 1 \quad \forall y = G(\underline{x}) | \underline{x} \in \Lambda \backslash \Psi_g(g^* - u_2) \tag{A9}$$

Accordingly:

$$\begin{aligned} Pr\{\mathcal{G}_{r_{k+1}} > u_2 | \mathcal{G}_{r_k} = u_1\} &= 1 - \frac{\mathbb{P}_{r_{k+1}}\left(\Psi_g(g^* - u_1) \backslash \Psi_g(g^* - u_2)\right)}{\mathbb{P}_{r_{k+1}}\left(\Psi_g(g^* - u_1)\right)} \\ &= 1 - \frac{1}{\mathbb{M}\left(\Psi_g(g^* - u_1)\right)} \int_{\Psi_g(g^* - u_1) \backslash \Psi_g(g^* - u_2)} \\ &\qquad H \circ G(\underline{x}) d\underline{x} \geq Pr\{\mathcal{W}_{k+1} > u_2 | \mathcal{W}_k = u_1\} \end{aligned} \tag{A10}$$

that completely proves A).

Let us now prove the proposition:

B) $\mathcal{G}_{r_k} \geq_{st} \mathcal{W}_k \qquad \forall k \in \mathsf{N}_0$

The proof is based on an induction process.

Obviously $\mathcal{G}_0 \geq_{st} \mathcal{W}_0$. And so we need to prove that if $\mathcal{G}_{r_k} \geq_{st} \mathcal{W}_k$ then $\mathcal{G}_{r_{k+1}} \geq_{st} \mathcal{W}_{k+1}$.

By exploiting the total probability theorem and A) it follows that:

$$\begin{aligned} Pr\{\mathcal{G}_{r_{k+1}} > u\} &= \int_{\mathsf{R}} Pr\left\{\mathcal{G}_{r_{k+1}} > u \,|\mathcal{G}_{r_k} = v\right\} dP_{\mathcal{G}_{r_k}}(v) \\ &\geq \int_{\mathsf{R}} Pr\left\{\mathcal{W}_{k+1} > u \,|\mathcal{W}_k = v\right\} dP_{\mathcal{G}_{r_k}}(v) \end{aligned} \tag{A11}$$

Since $Pr\{\mathcal{W}_{k+1} > u|\mathcal{W}_k = v\}$ is an increasing function of v, from the induction hypothesis and the very definition of Stieltjes integral, B) follows immediately.

Now, given the two stochastic processes:

$$\mathcal{M}(u) = \min_{k\in\mathsf{N}_0}\{\mathcal{G}_{r_k} > u\} \tag{A12a}$$

$$\mathcal{N}(u) = \min_{k\in\mathsf{N}_0}\{\mathcal{W}_k > u\} \tag{A12b}$$

from proposition B), once it is observed that:

$$Pr\{\mathcal{M}(u) > i\} = Pr\{\mathcal{G}_{r_i} > u\} \tag{A13a}$$

$$Pr\{\mathcal{N}(u) > i\} = Pr\{\mathcal{W}_i > u\} \tag{A13b}$$

it follows immediately that

$$\mathcal{M}(u) \geq_{st} \mathcal{N}(u) \tag{A14}$$

Consequently, from the basic properties of stochastic dominance [105], we have immediately:

$$E[\mathcal{M}(u)] \leq E[\mathcal{N}(u)] \tag{A15}$$

By exploiting the upper bound to the computational complexity of a PAS algorithm [65], we get:

$$E[\mathcal{M}(u)] \leq 1 + n\ln(\kappa d/(g^* - u)) \tag{A16}$$

where n, as usual, is the dimensionality of the optimization problem, d is the diameter of the searching region and κ is the Lipshitz constant of the optimized function.

As a concluding remark it is noted that the above results concerns only the process $\mathcal{G}_{r_k}$.

However, on average, according to the theory of the repeated trials process, the number of iterations required to improve the current feasible solution, is finite.

REFERENCES

1. Bucci, O. M., G. D'Elia, G. Mazzarella, and G. Panariello, "Antenna pattern synthesis: a new general approach," *Proc. IEEE*, Vol. 82, No. 3, 358–371, 1994.
2. Bucci, O. M., G. D'Elia, and G. Romito, "Reflector distortions diagnosis from far-field amplitude pattern," *IEEE Trans. Antennas and Propagat.*, Vol. 43, No. 11, 1217–1225, 1995.
3. Bucci, O. M., G. D'Elia, and G. Romito, "Synthesis technique for scanning and/or reconfigurable beams reflector antennas with phase-only control," *IEE Proc. Microw. Antennas Propagat.*, Vol. 143, 402–412, 1996.
4. Bucci, O. M., A. Capozzoli, and G. D'Elia, "New technique for wavefront reconstruction in optical telescopes," *J. Opt. Soc. Am. A*, Vol. 14, No. 12, 3394–3401, 1997.
5. Bucci, O. M. and G. D'Elia, "Power synthesis of reconfigurable conformal arrays with phase-only control," *IEE Proc. Microw. Antennas Propagat.*, Vol. 1, No. 45, 131–136, 1998.
6. Bucci, O. M., A. Capozzoli, and G. D'Elia, "Regularizing strategy for image restoration and wave-front sensing by phase diversity," *JOSA A*, Vol. 16, No. 7, 1759–1768, 1999.
7. Bucci, O. M., A. Capozzoli, and G. D'Elia, "A method for image restoration and wavefront sensing by using phase diversity," *Signal Recovery and Synthesis*, Vol. 11, OSA, Technical Digest Series, 28–30, 1998.
8. Pierri, R., G. D'Elia, and F. Soldovieri, "A two probes scanning phaseless near-field far-field transformation technique," *IEEE Trans. Antennas and Propagat.*, Vol. 47, No. 5, 792–802, 1999.
9. McCallum, B. C., "Blind deconvolution by simulated annealing," *Opt. Commun.*, Vol. 75, No. 2, 101–105, 1990.
10. Morris, D., "Simulated annealing applied to the Misell algorithm for phase retrieval," *IEE Proceedings, Pt. H*, Vol. 143, No. 4, 298–303, 1991.
11. Scales, J. A., M. L. Smith, and T. L. Fischer, "Global

optimization methods for multimodal inverse problems," *J. Comput. Phys.*, Vol. 103, 258–268, 1992.

12. Isernia, T., F. Soldovieri, G. Leone, and R. Pierri, "On the local minima in phase reconstruction algorithms," *Radio Science*, Vol. 31, No. 6, 1887–1999, 1996.
13. Pierri, R. and A. Tamburino, "On the local minima problem in conductivity imaging via a quadratic approach," *Inverse Problems*, Vol. 13, 1547–1568, 1997.
14. Zhai, J., Y. Yan, G. Jin, and M. Wu, "Global/local united search algorithm for global optimization," *Optik*, Vol. 108, 4, 1998.
15. Bucci, O. M., A. Capozzoli, and G. D'Elia, "A hybrid evolutionary algorithm in the diagnosis of reflector distortions," *JINA 98, International Symposium on Antennas*, Nizza, Francia, November 17–19, 1998.
16. Bucci, O. M., A. Capozzoli, and G. D'Elia, "A hybrid evolutionary algorithm in the diagnosis of reflector distortions from far-field amplitude pattern," *Atti della Fondazione Ronchi*, Vol. 54, No. 3, 503–519, 1999.
17. Bucci, O. M., A. Capozzoli, and G. D'Elia, "A Lamarckian-type evolutionary algorithm for large reflector diagnosis," *XXVI General Assembly of the International Union of Radio Science*, Toronto, Canada, August 13–21, 1999.
18. Bucci, O. M., A. Capozzoli, and G. D'Elia, "Diagnosis of array faults from far-field amplitude-only data," *IEEE Trans. Antennas and Propagat.*, Vol. 48, No. 5, 647–652, 2000.
19. Bucci, O. M., A. Capozzoli, and G. D'Elia, "A global optimization technique in the synthesis of reflector antennas," *17th Annual Review of Progress in Applied Computational Electromagnetics*, March 19–23, 2001.
20. Pardalos, P. M. and J. B. Rosen,*Constrained Global Optimization: Algorithms and Applications*, Springer-Verlag, 1987.
21. Torn, A. and A. Zilinskas, *Global Optimization*, Springer-Verlag, 1987.
22. Van Laarhoven, P. J. M. and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Kluwer, Dordrecht, 1987.
23. Horst, R. and H. Tuy, *Global Optimization: Deterministic Approaches*, Springer, 1992.
24. Horst, R. and P. M. Pardalos (Eds.), *Handbook of Global Optimization*, Kluwer, Dordrecht, 1995.
25. Back, T., D. B. Fogel, and Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, Institute of Physics Publishing,

Oxford University Press, 1997.

26. Tuy, H., *Convex Analysis and Global Optimization*, Kluwer, Dordrecht, 1998.

27. Gu, J. and X. Huang, "Efficient local search with search space smooting: A case of study of the traveling salesman problem (TSP)," *IEEE Trans. On Systems, Man, and Cybernetics*, Vol. 24, No. 5, 728–735, 1994.

28. Zabinsky, Z. B., "Stochastic methods for practical global optimization," *J. of Global Optimization*, Vol. 13, 433–444, 1998.

29. Graham, L. C., "Synthetic interference radar for topographic mapping," *Proc. IEEE*, Vol. 62, 763–768, 1974.

30. Romeijn, H. E. and R. L. Smith, "Simulated annealing and adaptive search in global optimization," *Probability in Engineering and Informational Sciences*, Vol. 8, 571–590, 1994.

31. Hiriart-Urruty, J. B., "Conditions for global optimality," *Handbook of Global Optimization*, R. Horst and P. M. Pardalos (Eds.), 1–26, Kluwer, Dordrecht, 1995.

32. Giaquinta, M. and S. Hildebrandt, *Calculus of Variation*, Vols. I–II, Springer, Berlin, 1996.

33. Kantorovic, L. V. and G. P. Akilov, *Functional Analysis*, MIR, Moscow, 1980.

34. Rinnooy Kan, A. H. G. and G. T. Timmer, "Stochastic methods for global optimization," *American Journal of Mathematical and Management Sciences*, Vol. 2, 1–2, 1984.

35. Zemanian, A. H., *Distribution Theory and Transform Analysis*, Dover, New York, 1987.

36. Gergel, V. P., "A global optimization algorithm for multivariate functions with Lipschitzian first derivatives," *Journal of Global Optimization*, Vol. 10, 257–281, 1997.

37. Hansen, P., B. Jaumard, and S. Lu, "Global optimization of univariate Lipschitz functions; I. Survey and properties," *Mathematical Programming*, Vol. 55, 251–272, 1992.

38. Hansen, P., B. Jaumard, and S. Lu, "Global optimization of univariate Lipschitz functions; II. New algorithms and computational comparison," *Mathematical Programming*, Vol. 55, 273–292, 1992.

39. Rinnooy Kan, A. H. G. and G. T. Timmer, "Stochastic methods for global optimization," *American Journal of Mathematical and Management Sciences*, Vol. 2, 1–2, 1984.

40. Hiriart-Urruty, J. B. and C. Lemarechal, *Convex Analysis and Minimization Algorithms*, Vols. I–II, Springer-Verlag, 1993.

41. Floudas, C. A. and V. Visweswarn, "Quadratic optimization," *Handbook of Global Optimization*, R. Horst and P. M. Pardalos (Eds.), 217–269, Kluwer, Dordrecht, 1995.
42. Hansen, P. and B. Jaumard, "Lipschitz optimization," *Handbook of Global Optimization*, R. Horst and P. M. Pardalos (Eds.), 407–493, Kluwer, Dordrecht, 1995.
43. Bonanno, G., "On a class of functions whose local minima are global," *Journal of Global Optimization*, Vol. 12, 101–104, 1998.
44. Benson, H. P., "Concave minimization: Theory, applications and algorithms," *Handbook of Global Optimization*, R. Horst and P. M. Pardalos (Eds.), 43–148, Kluwer, Dordrecht, 1995.
45. H. Tuy, "D.C. optimization: Theory, methods and algorithms," *Handbook of Global Optimization*, R. Horst and P. M. Pardalos (Eds.), 149–216, Kluwer, Dordrecht, 1995.
46. Luenberger, D. G., *Linear and Nonlinear Programming*, Addison-Wesley, 1984.
47. Strekalovsky, A. S., "Global optimality conditions for non convex optimization," *Journal of Global Optimization*, Vol. 12, 415–534, 1998.
48. Hiriart-Urruty, J. B., "Conditions for global optimality 2," *Journal of Global Optimization*, 349–367, 1998.
49. Levi, A. V. and A. Montalvo, "The tunneling algorithm for the global minimization of functions," *Siam J. Sci. Stat. Comput.*, Vol. 6, 1, 1985.
50. Falk, J. E., "Conditions for global optimality in nonlinear programming," *Operation Research*, Vol. 21, 337–340, 1973.
51. Pincus, M, "A closed form solution of certain programming problems," *Operation Research*, Vol. 16, 690–694, 1968.
52. Murty, K. G. and S. N. Kabadi, "Some NP-complete problems in quadratic and nonlinear programming," *Mathematical Programming*, Vol. 38, 117–129, 1987.
53. Wolpert, D. H. and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evolutionary Comput.*, Vol. 1, 1, 1997.
54. Waldrop, M. M., *Complexity: The Emerging Science at the Edge of Order and Chaos*, Simon & Schuster, 1992.
55. Ratschek, H. and J. Rokne, "Interval methods," *Handbook of Global Optimization*, R. Horst and P. M. Pardalos (Eds.), 751–828, Kluwer, Dordrecht, 1995.
56. Vavasis, S. A., *Nonlinear Optimization: Complexity Issues*, Oxford Science Publication, New York, 1991.

57. Nemirovsky, A. S. and D. B. Yudin, *Problem Convexity and Method Efficiency in Optimization*, John Wiley and Sons, 1983.

58. Vescovo, R., "Constrained and unconstrained synthesis of array factor for circular arrays," *IEEE Trans. on Antennas and Péropagat.*, Vol. 43, No. 12, 1405–1410, 1995.

59. Anderssen, R. S. and P. Bloominfield, "Properties of the random search in global optimization," *Journal of Optimization Theory and Applications*, Vol. 16, 383–398, 1975.

60. Boender, C. G. E. and H. E. Romeijn, "Stochastic methods," *Handbook of Global Optimization*, R. Horst and P. M. Pardalos (Eds.), 8299–869, Kluwer, Dordrecht, 1995

61. Solis, F. J. and R. J. B. Wets, "Minimization by random search techniques," *Math. Operation Res.*, Vol. 6, 19–30.

62. Rinnooy Kan, A. H. G. and G. T. Timmer, "Stochastic global optimization methods Part I: Clustering methods," *Mathematical Programming*, Vol. 39, 27–56, 1987.

63. Rinnooy Kan, A. H. G. and G. T. Timmer, "Stochastic global optimization methods Part II: Multi level methods," *Mathematical Programming*, Vol. 39, 57–78, 1987.

64. Patel, N. R., R. L. Smith, and Z. B. Zabinsky, "Pure adaptive search in Monte Carlo optimization," *Mathematical Programming*, Vol. 43, 317–328, 1988.

65. Zabinsky, Z. B. and R L. Smith, "Pure adaptive search in global optimization," *Mathematical Programming*, Vol. 53, 323–338, 1992.

66. Kirkpatrik, S., C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, Vol. 220, No. 4598, 671–680, 1983.

67. Romeijn, H. E. and R. L. Smith, "Simulated annealing for constrained global optimization," *Journal of Global Optimization*, Vol. 5, 101–126, 1994.

68. Byrd, R., C. L. Dert, A. H. G. Rinnooy Kan, and R. B. Schnabel, "Concurrent Stochastic methods for global optimization," *Mathematical Programming*, Vol. 46, 1–29, 1990.

69. Mahfoud, S., "Boltzmann selection," *Handbook of Evolutionary Computation*, T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), Institute of Physics Publishing, Oxford University Press, 1997.

70. Back, T., U. Hammel, and H. P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Trans. on Evolutionary Comput.*, Vol. 1, 1, 1997.

71. Back, T. and H. P. Schwefel, "An overview of evolutionary algo-

rithms for parameter optimization," *Evolutionary Computation*, Vol. 1, No. 2, 1–20, 1993.

72. Fogel, D. B., "An introduction to simulated evolutionary optimization," *IEEE Trans. Neural Networks*, Vol. 5, 1, 1994.
73. Kim, J. H. and H. Myung, "Evolutionary programming techniques for constrained optimization problems," *EEE Trans. Evolutionary Comput.*, Vol. 1, 2, 1997.
74. Tang, K. S., K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Processing Magazine*, 22–37, November 1996.
75. Holland, J. H., *Adaptation in Natural and Artificial Systems*, MIT Press, 1995.
76. De Jong, K., D. B. Fogel, and H. P. Schwefel, "A history of evolutionary computation," *Handbook of Evolutionary Computation*, T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), Institute of Physics Publishing, Oxford University Press, 1997.
77. Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
78. Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.
79. Back, T., G. Rudolph, and H. P. Schwefel, "Evolutionary programming and evolution strategies: Similarities and differences," *The Evolutionary Computation Repositary Network (ENCORE).*
80. Muhlenbein, H. and D. Schlierkamp-Voosen, "Predictive models for breeder genetic algorithms," *Evolutionary Computation*, Vol. 1, No. 1, 25–49, 1993.
81. Qi, X. Q. and F. Palmieri, "Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space Part I: Basic properties of selection and mutation," *IEEE Trans. Neural Networks*, Vol. 5, 1, 1994.
82. Qi, X. Q. and F. Palmieri, "Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space Part II: Analysis of the diversification role of crossover," *IEEE Trans. Neural Networks*, Vol. 5, 1, 1994.
83. Rudolph, G., "Convergence analysis of canonical genetic algorithms," *IEEE Trans. Neural Network*, Vol. 5, 1, 1994.
84. Francois, O., "An evolutionary strategy for global minimization and its Markov chain analysis," *IEEE Trans. Evolutionary Computat.*, Vol. 2, No. 3, 77–90, 1998.
85. Naudts, B. and L. Kallel, "A comparison of predictive measures of problem difficulty in evolutionary algorithms," *IEEE Trans.*

Evolutionary Computat., Vol. 4, 1, 2000.

86. Kellel, L., B. Naudts, A. Rogers, and G. Rozenberg, *Theoretical Aspects of Evolutionary Computing*, Springer Verlag, 2001.
87. Suzuki, J., "A Markov chain analysis on simple genetic algorithms," *IEEE Trans. Systems, Man and Cybernetics*, Vol. 25, 4, 1995.
88. Rudolph, G., "Stochastic processes," *Handbook of Evolutionary Computation*, T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), Institute of Physics Publishing, Oxford University Press, 1997.
89. Rudolph, G., "Modes of stochastic convergence," *Handbook of Evolutionary Computation*, T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), Institute of Physics Publishing, Oxford University Press, 1997.
90. Goldberg, D. E., "A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing," *Complex Systems*, Vol. 4, 445–460, 1990.
91. Fogel, D. B. and A. Ghozeil, "A note on representation and variation operators," *IEEE Trans. Evolutionary Computat.*, Vol. 1, 2, 1997.
92. Radcliffe, N. J, "Schema processing," *Handbook of Evolutionary Computation*, T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), Institute of Physics Publishing, Oxford Universitypress, 1997.
93. Eshelman, L. J. and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," *Foundation of Genetic Algorithm 2*, L. D. Whitley (Ed.), Morgan Kaufmann, 1993.
94. Renders, J. M. and S. P. Flasse, "Hybrid methods using genetic algorithms for global optimization," *IEEE Trans. Systems, Man and Cybernetics - Part B: Cybernetics*, Vol. 26, 2, 1996.
95. Hancock, P. J. B., "A comparison of selection mechanisms," *Handbook of Evolutionary Computation*, T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), IOP Publishing Ltd and Oxford University Press, 1997.
96. Grefenstette, J., "Proportional selection and sampling algorithms," *Handbook of Evolutionary Computation*, T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), IOP Publishing Ltd and Oxford University Press, 1997.
97. Back, T., D. B. Fogel, D. Whitley, and P. J. Angeline, "Mutation," *Handbook of Evolutionary Computation*, T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), IOP Publishing Ltd and Oxford University Press, 1997.
98. Rudolph, G., "Local convergence rates of simple evolutionary

algorithms with Cauchy mutation," *IEEE Trans. Evolutionary Computat.*, Vol. 1, 4, 1997.

99. Booker, L. B., D. B. Fogel, D. Whitley, and P. J. Angeline, "Recombination," *Handbook of Evolutionary Computation*, T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), IOP Publishing Ltd and Oxford University Press, 1997.

100. Eiben, A. E., R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. Evolutionary Computat.*, Vol. 3, 2, 124–141, 1999.

101. Back, T., "Self adaptation in genetic algorithms," *Evolutionary Computation*, Vol. 1, No. 2, 1–20, 1993.

102. Srinivas, M. and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. Systems, Man and Cybernetics*, Vol. 24, 4, 1994.

103. Grefenstette, J. J., "Optimization of control parameter for genetic algorithms," *IEEE Trans. Systems, Man and Cybernetics*, Vol. 16, 1, 1986.

104. Smith, R. E., "Population size," *Handbook of Evolutionary Computation*, T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), Institute of Physics Publishing-Oxford University Press, 1997.

105. Ross, S. M., *Stochastic Processes*, J. Wiley and Sons, New York, 1983.