Signal Processing 92 (2012) 1338–1352

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Application of natural computing algorithms to maximum likelihood estimation of direction of arrival

Levy Boccato [a,*], Rafael Krummenauer [b], Romis Attux [a], Amauri Lopes [b]

[a] Department of Computer Engineering and Industrial Automation, FEEC/6101, University of Campinas - UNICAMP, 13083-852 Campinas, SP, Brazil
[b] Department of Communications, FEEC/6101, University of Campinas - UNICAMP, 13083-852 Campinas, SP, Brazil

ARTICLE INFO

Article history:
Received 17 November 2010
Received in revised form
9 September 2011
Accepted 2 December 2011
Available online 9 December 2011

Keywords:
Array signal processing
Direction of arrival estimation
Maximum likelihood estimation
Optimization
Natural computing

ABSTRACT

This work presents a study of the performance of populational meta-heuristics belonging to the field of natural computing when applied to the problem of direction of arrival (DOA) estimation, as well as an overview of the literature about the use of such techniques in this problem. These heuristics offer a promising alternative to the conventional approaches in DOA estimation, as they search for the global optima of the maximum likelihood (ML) function in a framework characterized by an elegant balance between global exploration and local improvement, which are interesting features in the context of multimodal optimization, to which the ML-DOA estimation problem belongs. Thus, we shall analyze whether these algorithms are capable of implementing the ML estimator, i.e., finding the global optima of the ML function. In this work, we selected three representative natural computing algorithms to perform DOA estimation: differential evolution, clonal selection algorithm, and the particle swarm. Simulation results involving different scenarios confirm that these methods can reach the performance of the ML estimator, regardless of the number of sources and/or their nature. Moreover, the number of points evaluated by such methods is quite inferior to that associated with a grid search, which gives support to their application.

1. Introduction

A major problem of interest in array signal processing corresponds to estimating the direction of arrival (DOA) of signals impinging on a sensor array. In this context, the maximum likelihood (ML) method emerges as the standard approach due to its excellent performance in both asymptotic and threshold regimes. However, the ML estimator may be difficult to implement, since it requires the minimization of a cost function which is nonlinear and multimodal, especially under critical conditions of signal-to-noise ratio (SNR) [1–3]. Furthermore, the surface of the ML cost function is considerably modified when the SNR varies, including in terms of the location of its global optima.

These peculiar characteristics motivated the development of several alternative ML-based methods, such as Alternating Projection [4], Expectation-Maximization (EM) [5,6] and Space Alternating Generalized EM (SAGE) [7,8]. All these techniques involve an iterative procedure that depends on the initialization of the unknown parameters to adequately locate the solution. Thus, they are constantly haunted by the possibility of converging to local minima. Other examples of ML-based methods are IQML [9], MODE [2], MODEX [10] and Modified MODEX [11]. Unfortunately, all these approaches present a performance degradation when the SNR or the number of data samples are reduced below certain critical values. In other words, they are not capable of reproducing the performance of the ML estimator in some situations.

* Corresponding author. Tel.: +55 19 34215051.
E-mail address: lboccato@dca.fee.unicamp.br (L. Boccato).

0165-1684/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2011.12.004

More recently, Stoica and Gershman analyzed the data-supported optimization (DSO) approach in DOA estimation [12,13]. Basically, the idea is to generate a grid of candidate solutions by means of an ESPRIT or MUSIC-like method and then evaluate each point according to the ML criterion, keeping the best one as the DOA estimate. In both works, the performance of the proposed approach was compared to that reached by a genetic algorithm (GA), which was employed to provide an approximation for the ML estimator, i.e., the reference of performance in terms of estimation error. However, the GA was applied in a black box way, i.e., without a detailed analysis regarding its parameters or the operators it employs, which means that many important details concerning its real potential to perform DOA estimation have not been addressed.

The idea of applying genetic algorithms and another well-known meta-heuristic called Simulated Annealing (SA) to perform the direction of arrival estimation has been initially explored by Sharman in [14,15], respectively. However, the performance of such algorithms were assessed only for a small number of scenarios and SNR values. It is also worth mentioning the works in [16–18], in which GA-based algorithms were proposed and used to perform the DOA estimation. Clearly, there are many important issues related to the use of meta-heuristics in ML-DOA estimation problem that still require deeper studies. Moreover, there are many different meta-heuristics other than GA, for instance, immune-inspired algorithms, that should be tested.

In this sense, this work is intended to step forward in this direction by expanding the repertoire of studied algorithms as well as analyzing the performance of these methods considering different DOA scenarios, including the case with more than two sources. Additionally, this work also represents an extension of the study performed in [19], where six different bio-inspired meta-heuristics have been applied to DOA estimation, achieving the expected performance of the ML estimator in a DOA scenario we may refer to as a classical case study within the literature. Here, we explore the parameter setting issue with more details and consider DOA scenarios beyond that used in [19], in order to analyze the influence of some characteristics related to the source nature, like the presence of correlation between the signals, in the performance of the natural computing algorithms. Moreover, we analyze how these algorithms cope with the presence of different number of sources.

In other words, the application of representative natural computing algorithms in ML-DOA estimation has been addressed in this paper with the purpose of providing evidences that these methods are capable of identifying the direction of arrival according to the ML criterion. The computational burden associated with these algorithms will be expressed in terms of the number of points they sample and evaluate during the search for the global optima of the ML cost function, and it shall be compared to that of a grid search, which in fact represents the only approach that locates the global optima in all experiments.

This paper is organized as follows: Section 2 presents the signal model and the ML estimator, illustrating the particular features of the cost function to be optimized; Section 3 emphasizes the motivation of this work and provides the description of the employed natural computing algorithms: from evolutionary computation, the differential evolution approach [20]; from immune computing, the clonal selection algorithm (CLONALG) [21]; and, from swarm intelligence, the particle swarm [22]; Section 4 describes the methodology adopted to the performance analysis, along with the DOA scenarios considered in this work, while Section 5 presents the obtained results and some important remarks; finally, Section 6 concludes the paper.

2. Direction of arrival estimation

2.1. Data model

Consider M narrowband plane waves impinging on a uniform linear array containing N ($N > M$) isotropic sensors equally spaced with $d = \lambda/2$, where λ represents the wavelength. The waves arrive at angles $\boldsymbol{\theta} = [\theta_1 \dots \theta_M]^T$ measured from the normal to the array axis ($[\cdot]^T$ denotes the transpose).

It can be shown [23] that the problem of estimating the angles $\{\theta_m\}_{m=1}^{M}$ can be reduced to the problem of estimating the frequencies

$$\omega_m = \pi \sin \theta_m, \tag{1}$$

in the following classical mathematical model [2,24]:

$$\mathbf{y}_k = \mathbf{A}\mathbf{x}_k + \boldsymbol{\eta}_k, \quad k = 1, 2, \dots, K, \tag{2}$$

where $\mathbf{y}_k \in \mathbb{C}^{N\times 1}$ represents the noisy signal received in the k-th snapshot, $\mathbf{x}_k \in \mathbb{C}^{M\times 1}$ represents the unknown transmitted signals, $\boldsymbol{\eta}_k \in \mathbb{C}^{N\times 1}$ is the noise vector, K is the number of snapshots and $\mathbf{A} = [\mathbf{a_1} \dots \mathbf{a_M}] \in \mathbb{C}^{N\times M}$ is the array transfer matrix, with $\mathbf{a_m} = [1\ e^{j\omega_m} \dots, e^{j(N-1)\omega_m}]^T$.

The number M of sources is assumed to be known (given or estimated) [25]. The signal and the noise are independent zero-mean complex stationary Gaussian random processes with second-order moments given as follows:

$$E\{\mathbf{x}_k \mathbf{x}_i^H\} = \mathbf{C}\delta_{k,i}, \quad E\{\mathbf{x}_k \mathbf{x}_i^T\} = \mathbf{0},$$

$$E\{\boldsymbol{\eta}_k \boldsymbol{\eta}_i^H\} = \sigma^2 \mathbf{I}\delta_{k,i}, \quad E\{\boldsymbol{\eta}_k \boldsymbol{\eta}_i^T\} = \mathbf{0}, \tag{3}$$

where $E\{\cdot\}$ corresponds to the statistical expectation operator, $\mathbf{C}$ is the unknown signal covariance matrix, $\delta_{k,i}$ is the Kronecker delta operator, σ^2 is the unknown noise power, $\mathbf{I}$ denotes the identity matrix and $[\cdot]^H$ refers to the complex conjugate transpose.

We shall adopt the conditional model (CM) [26]. Hence, the same realization of the signal vector $\mathbf{x}_k$ is used in all experiments with K snapshots, whereas $\boldsymbol{\eta}_k$, $k = 1, \dots, K$, varies from experiment to experiment.

The spatial covariance matrix of $\mathbf{y}_k$ is

$$\mathbf{R}_\mathbf{y} = E\{\mathbf{y}_k \mathbf{y}_k^H\} = \mathbf{A}\mathbf{C}\mathbf{A}^H + \sigma^2 \mathbf{I}, \tag{4}$$

and, in the case of a finite number of snapshots, it can estimated using the sample covariance matrix

$$\hat{\mathbf{R}}_{\mathbf{y}} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{y}_k \mathbf{y}_k^H. \tag{5}$$

2.2. Maximum likelihood estimator

Considering the hypotheses presented in the previous section and the conditional model assumption, the deterministic maximum likelihood estimate of $\boldsymbol{\theta}$ is obtained by solving

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\min_{\boldsymbol{\theta}} J_{ML}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \mathrm{Tr}\{\mathbf{P}_{\mathbf{A}}^{\perp}\hat{\mathbf{R}}_{\mathbf{y}}\}, \tag{6}$$

where $\mathrm{Tr}\{\mathbf{M}\}$ denotes the trace of a matrix $\mathbf{M}$, $\mathbf{P}_{\mathbf{A}}^{\perp} = \mathbf{I} - \mathbf{A}\mathbf{A}^{\perp}$ is the projection matrix related to the noise subspace, $\mathbf{A}^{\perp} = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H$ is the pseudo-inverse of $\mathbf{A}$, $\mathbf{A} = \mathbf{A}(\boldsymbol{\theta})$ and $\boldsymbol{\theta} \in \mathbb{R}^{M\times 1}$ is a candidate direction of arrival vector [27].

The cost function J_{ML} is nonlinear, non-quadratic and multimodal. Fig. 1 displays its surface and its contour considering the following DOA scenario: $N=10$, $M=2$, $K=100$, $\boldsymbol{\theta} = [10^\circ\ 15^\circ]^T$, $\mathbf{C} = \mathbf{I}$. The signal-to-noise ratio, given by $10\log 1/\sigma^2$, assumes the values 0 dB and -15 dB.

It is possible to notice, in Fig. 1, that, as the SNR decreases, not only does the number of local minima increase, but also the global optima suffer a displacement, moving apart from the true DOA angles. This means that the surface of J_{ML} can be deeply modified when the SNR changes. Therefore, from the optimization standpoint, any ML-based method is faced with several different optimization surfaces because of the SNR effect, even considering a single DOA scenario.

These evidences highlight the obstacles that ML-based methods must tackle in order to correctly estimate the angles. Due to these aspects, only an exhaustive search can surely locate the global optimum of J_{ML}. However, this approach is not practical because of the intensive computation it requires, especially when the number of sources increases.

We can approximate the exhaustive search by using a uniform M-dimensional grid. Assuming the same resolution in each coordinate, given by the distance p_x between adjacent points, the total number of points within the grid, which corresponds to the number of evaluations of J_{ML}, is expressed as follows:

$$N_P = \left\lfloor \frac{x_{max} - x_{min}}{p_x} + 1 \right\rfloor^M, \tag{7}$$

where (x_{min}, x_{max}) represents the excursion interval in each coordinate. Observe that N_P grows exponentially as M increases, which restricts the use of this approach to scenarios with a small number of sources.

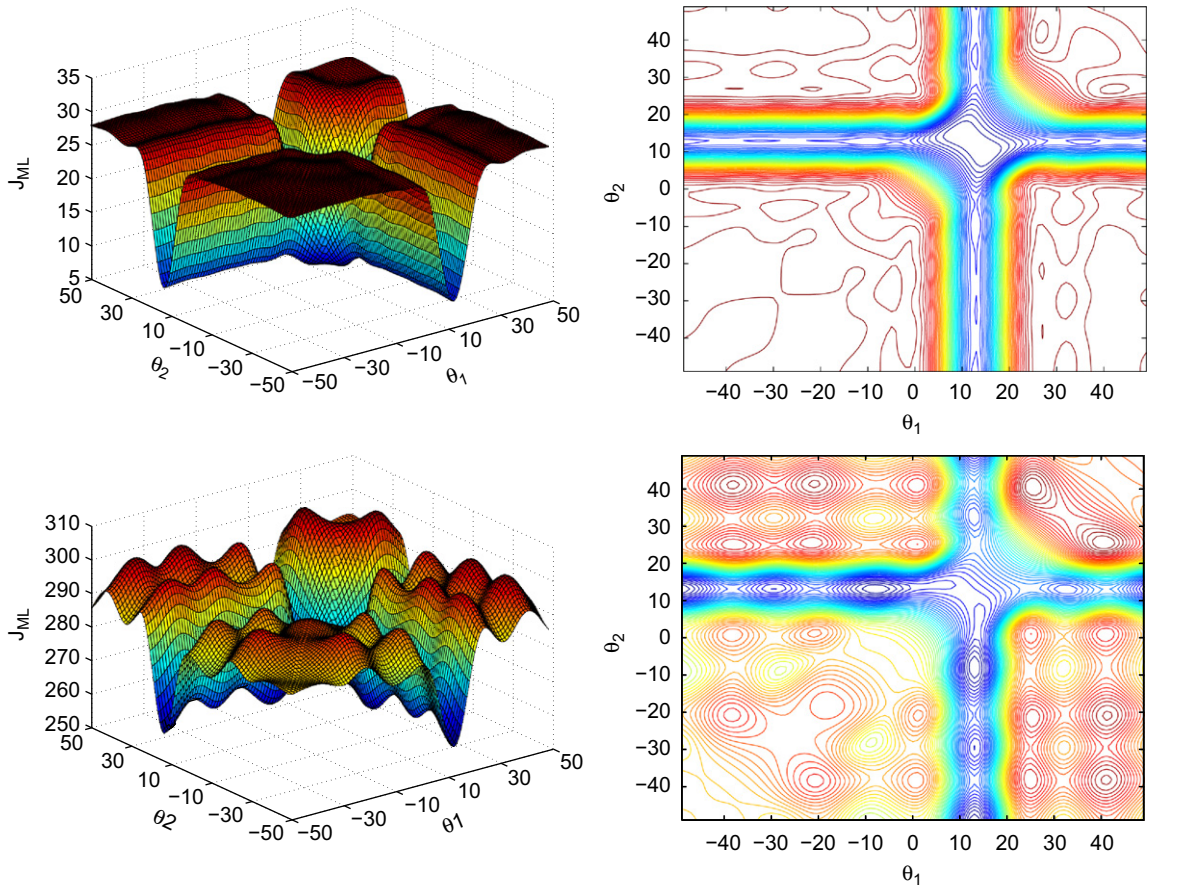

Fig. 1. Surface and contour of the ML cost function. (a) SNR=0 dB. (b) SNR=0 dB. (c) SNR=−15 dB. (d) SNR=−15 dB.

In order to assess the performance of an estimator, we usually employ the root mean squared error (*RMSE*):

$$RMSE = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N_e}\frac{(\theta_i-\hat{\theta}_{ij})^2}{MN_e}}, \quad (8)$$

where θ_i is the i-th direction of arrival, $\hat{\theta}_{ij}$ denotes the i-th angle estimate in the j-th experiment, M is the number of sources and N_e is the number of experiments.

Fig. 2 shows the performance of the grid search and the Cramér–Rao bound (CRB), which gives the theoretical minimum variance for any unbiased estimator [26], considering the same DOA scenario described above. Aiming for sufficiently reliable RMSE values, we adopted $N_e=1000$ and $p_x=0.01^\circ$.

Some interesting remarks can be drawn from Fig. 2: (1) the grid search performance asymptotically converges to the CRB. (2) as the SNR decreases, both the performances of the CRB and the grid search deteriorate; this is because the increasing noise displaces the global optima of J_{ML} from the true DOA angles, so that the estimation error becomes more pronounced. (3) below a specific SNR value, commonly referred to as threshold SNR, the grid search cannot offer the same performance of CRB. This phenomenon is called threshold effect [23,28].

It is important to remark that, for the low values of SNR, i.e, below the threshold, the CRB does not represent a tight bound for realizable estimators. In this region, other lower bounds, like the Ziv–Zakai bound [29], the Weiss–Weinstein bound [30], and the Barankin bound [31], provide much tighter performance limits.

Nonetheless, the RMSE curve displayed in Fig. 2 actually corresponds to the desired performance, since it represents a close approximation of the ML estimator. However, as mentioned before, the computational burden of grid search practically hinders its application.

In order to circumvent the intensive computation required for the search of the global optima of J_{ML}, various ML-based methods have been developed. Among them, we highlight MODE [2], MODEX [10], and Modified MODEX [11].

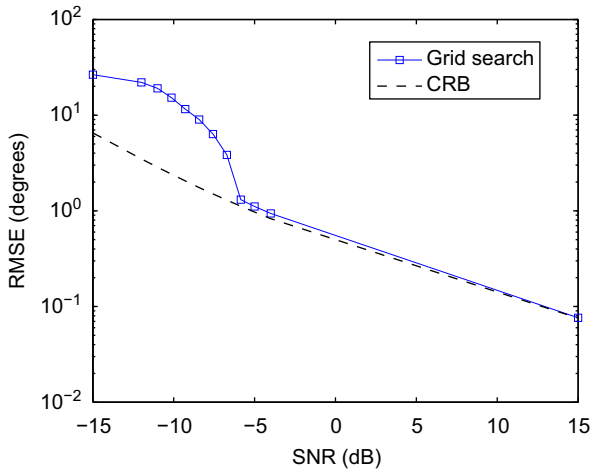

Fig. 2. Grid search RMSE curve and Cramér–Rao bound.

Basically, MODE employs a reparametrization of the original ML estimator and a restriction to the signal subspace of the snapshot covariance matrix, leading to a constrained quadratic optimization problem. The solution of this problem yields the coefficients of a polynomial whose roots correspond to the DOA estimates. In order to avoid poor DOA estimates that remain very distant from the true DOA angles, known as outliers, MODEX uses an extended polynomial, thus generating a larger repertoire of roots. Each combination of these roots produces a DOA candidate solution and a maximum likelihood procedure selects the best of these combinations. Finally, Modified MODEX alters the MODEX extra root generator aiming to produce fewer and better candidates. As verified in [11], this modification reduces the computational effort of MODEX and achieves a better estimation performance.

Still, none of these methods has been proved to achieve the performance of the ML estimator. Indeed, as already evinced in the literature, these methods fail to determine the global optimum of J_{ML} in a range of SNR values. In this work, we propose the application of natural computing algorithms, in the ML-estimation of direction of arrival. Our main goal is to analyze whether these methods can solve the J_{ML} optimization problem, i.e., find the best DOA estimates according to ML criterion.

3. Natural computing algorithms

We start this section by defining some concepts related to natural computing in order to familiarize the reader. Then, the motivation of this work is described, and, finally, we proceed to the presentation of the three natural computing algorithms considered in this paper.

Briefly, natural computing can be defined as the research field that, based on or inspired by nature, endeavors to develop new computational tools (software or hardware) for problem solving, simulate and mimic natural phenomena, behaviors and organisms, and may result in the design of novel computational systems using different natural materials [32].

The application scope of natural computing tools is quite vast and diversified: planning, machine learning, pattern recognition, clustering, autonomous navigation, function approximation and optimization are some examples [32,33].

But, when should we consider using natural computing algorithms? In [32], the author provides some pertinent hints on the cases where these methods could be employed. For example, when the problem to be solved is too complex, i.e, involves a large number of variables or potential solutions, is highly dynamic, nonlinear or multimodal, these algorithms represent an interesting alternative.

That is precisely the case of maximum likelihood DOA estimation. As explained before, this problem requires the minimization of a nonlinear and multimodal cost function that also significantly varies according to the SNR. These challenging features hamper the correct identification of the DOA estimates and contribute to the performance deterioration verified for many different ML-based methods.

Besides, the fact that some natural computing algorithms have obtained excellent performances in similar problems, including in the signal processing context [34], also gives

support to the proposal of using such methods to perform DOA estimation. However, it is always necessary to keep in mind that, since we are dealing with populational metaheuristics and, in view of No-Free-Lunch Theorem [35,36], there are no a priori guarantees that these algorithms will actually reach the desired performance. Nevertheless, we expect that, by employing a population of candidate solutions, these algorithms succeed in simultaneously exploring different regions of the search space and in improving the candidate solutions by promoting interactions among them, in order to finally locate the global optima of J_{ML}.

Because of these reasons, we believe it is necessary, and certainly beneficial, to study the application of natural computing algorithms in DOA estimation. In the subsequent sections, we describe the details of the three natural computing algorithms used in this work.

Before proceeding to the actual description of these methods, it is important to define three basic concepts: the representation, the objective and the evaluation function (also called fitness function) [37]. The representation establishes how each candidate solution to a particular problem is encoded, and, therefore, how it can be manipulated. The objective defines the goal, i.e., the problem to be solved, whereas the fitness function provides a quality measure for each candidate solution, thus allowing a comparison among different solutions.

Specifically in the DOA estimation problem, since we are interested in obtaining $\boldsymbol{\theta} = [\theta_1 \dots \theta_M]^T$, it is appropriate to represent each candidate solution as a real-valued vector. The objective, in turn, is given in Eq. (6): to minimize $J_{ML}(\boldsymbol{\theta})$ to find the ML-DOA estimates.

Finally, we must choose the evaluation function $F_{fit}(\boldsymbol{\theta})$: $\mathbb{R}^M xmap; \mathbb{R}$ for the DOA estimation problem. Given a vector $\boldsymbol{\theta} \in \mathbb{R}^M$ with DOA estimates, the function $F_{fit}(\boldsymbol{\theta})$ shall yield a value indicating the quality of these estimates. In this sense, the function $J_{ML}(\boldsymbol{\theta})$ itself could represent the fitness function. However, since most natural computing algorithms usually work towards privileging candidate solutions with higher fitness, it is convenient to convert the minimization of J_{ML} into an equivalent maximization problem. One way to perform this adaptation is given as follows:

$$F_{fit}(\boldsymbol{\theta}) = \frac{1}{1+J_{ML}(\boldsymbol{\theta})}. \tag{9}$$

Thus, $F_{fit}(\boldsymbol{\theta})$ represents the function to be optimized, whose global maximum corresponds to the best DOA estimates according to the ML criterion.

Having these concepts in mind, we now proceed to the presentation of the natural computing algorithms considered in this paper.

3.1. Differential evolution

In 1995, Storn and Price proposed a new algorithm devoted to the optimization of real functions called differential evolution (DE) [20,38]. This method has the distinctive feature of performing perturbations in each candidate solution using solely the information that lies within the population, in contrast with the usual procedure in evolutionary computation, which is based on predetermined probability density functions.

DE uses a population of N_P parameter vectors M-dimensional $\mathbf{x}_{i,G}$, $i = 1, \dots, N_P$, in each generation G. For each individual, a new mutated vector is obtained according the following expression:

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F \cdot (\mathbf{x}_{r_3,G} - \mathbf{x}_{r_2,G}), \tag{10}$$

where $r_1, r_2, r_3 \in \{1, 2, \dots, N_P\}$ are mutually distinct random indexes and different from i. The real constant $F \in [0, 2]$ defines the step size in the direction given by the difference vector $\mathbf{x}_{r_3,G} - \mathbf{x}_{r_2,G}$. This operation is usually referred to as mutation [38,39].

Next, we have the recombination phase, also known as crossover. Let $\mathbf{x}_{i,G}$ be the i-th individual and $\mathbf{v}_{i,G+1}$ the respective mutated vector. The vector $\mathbf{u}_{i,G+1} = [u_{1i,G+1}\ u_{2i,G+1}\ \dots\ u_{Mi,G+1}]$, called trial vector, is obtained as follows:

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } r_j \le CR \text{ or } j = I_i, \\ x_{ji,G} & \text{if } r_j > CR \text{ and } j \ne I_i, \end{cases} \tag{11}$$

where $j = 1, \dots, M$, $r_j \backsim \mathcal{U}(0, 1)$, $CR \in [0, 1]$ is the crossover rate and I_i is a randomly chosen index from the interval $\{1, \dots, M\}$, which ensures that $\mathbf{u}_{i,G+1}$ gets at least one component from $\mathbf{v}_{i,G+1}$.

Finally, the selection of the individuals that will survive to next generation is performed according to a greedy (elitist) criterion: for each $\mathbf{x}_{i,G}$, if the respective trial vector $\mathbf{u}_{i,G+1}$ has a better fitness value, then $\mathbf{x}_{i,G+1} = \mathbf{u}_{i,G+1}$. Otherwise, $\mathbf{x}_{i,G+1} = \mathbf{x}_{i,G}$. These steps are repeated until a fixed number of iterations is reached.

The notation DE/rand/1/bin refers to the algorithm previously described and represents the standard version of differential evolution [38]. Algorithm 1 depicts the pseudocode of DE.

Algorithm 1. Differential evolution.

```
Function φ = DE(N_P,M,CR,F,range,F_fit)
[φ,f_φ] ⇐ initialize(N_P,M,range,F_fit)
while stopping criterion is not reached do
  for i=1 until N_P do
      v_{i,G+1} ⇐ mutation(φ_{i,G},F)
      u_{i,G+1} ⇐ crossover(φ_{i,G},v_{i,G+1},CR)
  end for
  f_u ⇐ evaluate(u,F_fit)
  for i=1 until N_P do
      if f_u(i) > f_φ(i) then
      φ_{i,G+1} ⇐ u_{i,G+1}
    else
      φ_{i,G+1} ⇐ φ_{i,G}
    end if
  end for
end while
```

3.2. CLONALG

Developed by de Castro and Von Zuben, in 2002, the algorithm named CLONALG employs the clonal selection principle and the affinity maturation process as the fundamental mechanisms to manipulate a population of candidate solutions, called antibodies [21].

In a few words, the clonal selection principle establishes the idea that only the cells which are capable

of recognizing a specific antigen, i.e., those with higher affinity, proliferate, thus being selected instead of the cells that could not recognize the antigen [40]. The selected cells are then stimulated to reproduce, generating copies (clones) in a number proportional to its affinity: the higher the affinity, the more clones are generated.

As all reproductive events, the cloning process is susceptible to errors (mutations). However, the mutations within the immune system have the particular feature of occurring in a rate inversely proportional to the affinity between the cells and the antigen. This adaptation process of immune cells is called affinity maturation [41].

Originally designed to cope with pattern recognition tasks, the CLONALG algorithm was adapted to the multimodal optimization domain. Next, we present a basic description of CLONALG [32]:

1. Initialization: create an initial population **Ab** with N_P antibodies.
2. Evaluation: determine the fitness of each antibody $\mathbf{Ab_i}$, with $i=1,\ldots,N_P$.
3. Clonal selection and expansion: select the best n_1 antibodies from **Ab** and clone these antibodies proportionally to their fitness—the higher the fitness value, the more clones are generated, and vice versa.
4. Affinity maturation: mutate all clones with a mutation rate inversely proportional to the fitness of the original antibodies—the higher the fitness value, the smaller the mutation rate becomes, and vice versa. Then, evaluate the clone repertoire in order to form the new population **Ab** with the n_1 best clones.
5. Metadynamics: replace a percentage p_{Ab} of the antibodies in **Ab** with the smallest fitness values for randomly generated antibodies. This procedure may be performed in periods of T_{Ab} iterations.
6. Cycle: repeat steps 2–5 until a pre-determined stopping criterion is reached.

Let $\mathbf{C_{Ab}^{(i)}}$ represent a clone of the i-th antibody in **Ab**. The mutation operator consists of adding a zero-mean Gaussian random variable with a standard deviation inversely proportional to the fitness of $\mathbf{Ab_i}$, as shown in the following expressions:

$$\overline{\mathbf{Ab_i}} = \mathbf{C_{Ab}^{(i)}} + \alpha_m N(0,1), \tag{12}$$

$$\alpha_m = \frac{1}{\rho} e^{-fAb_i^*}, \tag{13}$$

where $\overline{\mathbf{Ab_i}}$ indicates the mutated antibody, ρ is a parameter that controls the decay of the inverse exponential function, and fAb_i^* denotes the fitness of the i-th antibody normalized in the interval (0,1).

It is important to mention that, when we are interested in finding multiple optima using a single population, some recommendations concerning the parameters of CLONALG are useful: (1) $n_1 = N_P$, i.e., all antibodies in **Ab** are selected in the cloning phase; (2) the number N_c of clones generated for each antibody is the same, that is, it does not depend on the fitness. Algorithm 2 displays the pseudocode of CLONALG.

Algorithm 2. CLONALG.

```
Function [Ab]=CLONALG(N_P,M,N_c,range, ρ,T_Ab,p_Ab, max_it,F_fit)
[Ab,fAb] ⇐ initialize(N_P,M,range,F_fit)
for it=1 until max_it do
  for i=1 until N_P do
    C_Ab^(i) ⇐ clone(Ab_i,N_c)
    Ab_i (mutated) ⇐ mutate(C_Ab^(i),fAb_i,N_c, ρ)
    fC ⇐ evaluate(Ab_i (mutated),F_fit)
    [f_max,imax] ⇐ max(fC)
    if f_max > fAb_i then
      Ab_i ⇐ Ab_i (mutated)(imax)
      fAb_i ⇐ f_max
    end if
  end for
  if it is a multiple of T_Ab then
    [Ab,fAb] ⇐ insert(Ab, fAb,p_Ab,N_P,M,range,F_fit)
  end if
end for
```

3.3. Particle swarm

In 1995, Kennedy and Eberhart proposed the algorithm named particle swarm (PS) inspired by the way human societies process knowledge [42,43]. Like other approaches related to swarm intelligence, PS is based on a population of individuals capable of interacting with the environment and with one another, in particular with some of its neighbors.

A relatively simple sociocognitive theory underlies the PS approach. In [44,45], the authors state that social adaptation has a low level component, which corresponds to the actual behavior of the individuals, and a high level component, which is related to the formation of patterns across individuals. Hence, each particle within the swarm has its own experience and is capable of assessing its quality. Additionally, as social beings, they are informed about how other neighboring particles are performing. These two types of information refer to individual learning and social (cultural) transmission, respectively [32].

The individuals in PS are viewed as points in the search space and their changes are represented as movements. In mathematical terms, the position of the i-th particle is represented by $\mathbf{x_i} \in \mathbb{R}^{M\times 1}$, and is updated according to

$$\mathbf{x_i}(t+1) = \mathbf{x_i}(t) + \boldsymbol{v_i}(t+1). \tag{14}$$

The vector $\boldsymbol{v_i}$, usually referred to as the velocity of the particle, indicates the direction of the next movement, and is a function of the current position of the particle $\mathbf{x_i}(t)$, its change in position $\boldsymbol{v_i}(t)$, the location of the particle's best success so far $\mathbf{p_i}(t)$, and the best position found by any member of its neighborhood $\mathbf{p_g}(t)$:

$$\boldsymbol{v_i}(t+1) = \boldsymbol{v_i}(t) + \boldsymbol{\psi_1} \odot (\mathbf{p_i}(t) - \mathbf{x_i}(t)) + \boldsymbol{\psi_2} \odot (\mathbf{p_g}(t) - \mathbf{x_i}(t)), \tag{15}$$

where $\boldsymbol{\psi_1}$ and $\boldsymbol{\psi_2}$ are positive random vectors drawn from uniform distributions with predefined upper limits: $\boldsymbol{\psi_1} \backsim \mathcal{U}(0,L_1)$ and $\boldsymbol{\psi_2} \backsim \mathcal{U}(0,L_2)$.[1] The operator $\odot$ denotes an elementwise vector multiplication.

[1] According to Kennedy [45], the sum between the acceleration constants should be equal to $L_1+L_2=4.1$, and the most common procedure is to adopt $L_1=L_2=2.05$.

It is important to mention that neighborhood does not necessarily mean similarity in the parameter space. In fact, this concept refers to topological similarity in a given population structure. Thus, each particle has access to the performance and position of its neighbors in the adopted topology. There are many different proposals for neighborhood topology, among which we highlight the ring structure (each particle has a left and right neighbor in a ring-shaped topology) and the globally connected neighborhood (all particles are mutually connected) [32].

In 2002, Clerc and Kennedy [22] introduced a modification in the original PS algorithm that is currently known as the standard particle swarm algorithm. It involves the use of a global constriction factor χ, as follows:

$$\boldsymbol{v_i}(t+1) = \chi\{\boldsymbol{v_i}(t) + \boldsymbol{\psi_1} \otimes (\mathbf{p_i}(t) - \mathbf{x_i}(t)) + \boldsymbol{\psi_2} \otimes (\mathbf{p_g}(t) - \mathbf{x_i}(t))\}. \tag{16}$$

The introduction of the constriction factor χ eliminates the need for defining boundaries for the elements of $\boldsymbol{v_i}$. Besides, depending on the values of L_1 and L_2, the choice of χ can mitigate the occurrence of undesirable behaviors, such as explosion, and even favor swarm convergence [22]. This is the PS version we employed in this work, and Algorithm 3 exhibits its pseudocode.

Algorithm 3. Particle swarm.

Function [**Ab**]=PS(N_P,M,range, v_{min}, v_{max},L_1,L_2,F_{fit})
[**Pop**,**fPop**,$\boldsymbol{v}$] ⇐ initialize(N_P,M,range, v_{min}, v_{max},F_{fit})
while stopping criterion is not reached **do**
  for $i=1$ until N_P **do**
    update particle's best position ($\mathbf{p_i}$) and the best position of neighbors ($\mathbf{p_g}$)
    $\boldsymbol{v_i}(t+1) \Leftarrow \chi\{\boldsymbol{v_i}(t) + \boldsymbol{\psi_1} \otimes (\mathbf{p_i}(t) - \mathbf{x_i}(t)) + \boldsymbol{\psi_2} \otimes (\mathbf{p_g}(t) - \mathbf{x_i}(t))\}$
    $\mathbf{Pop_i} \Leftarrow \mathbf{Pop_i} + \boldsymbol{v_i}(t+1)$
  end for
end while

4. Methodology

The first step in the application of natural computing algorithms to a particular problem consists in specifying how each candidate solution is represented and a function to assess its quality according to the desired objective. These important elements have been defined in Section 3 for the DOA estimation problem. The following step is the definition of the basic operators each algorithm will employ in order to manipulate the population of candidate solutions. Again, Section 3 contains the full description of the operators used in this work.

Therefore, the remaining issue to be dealt with is the parameter setting, which plays a crucial role in determining the behavior of each algorithm and, ultimately, whether the algorithm will succeed in the desired application. However, assigning values for each parameter can often be an arduous task.

As remarked in Section 2.2, the surface of the ML cost function is significantly modified when the SNR varies. Moreover, similar modifications may also occur when we consider different experiments for the same SNR. These peculiarities aggravate the parameter setting difficulty, since each algorithm must cope with diverse surfaces to be optimized, hopefully, with the same excellence, using a fixed set of values for its parameters.

Due to these facts, we decided to perform sensitivity tests for the most relevant parameters of each natural computing algorithm considered in this work. Through these tests, we expect to visualize the influence of each parameter alone on the performance of the algorithm in DOA estimation.

Consider the following DOA scenario: $N=10$, $M=2$, $K=100$, $\boldsymbol{\theta} = [10^\circ\ 15^\circ]^T$ and $\mathbf{C} = \mathbf{I}$. The SNR assumes the values -10 dB, -5 dB and 15 dB, so that we may obtain indicatives of the performance in both threshold and asymptotic regimes. In the former cases, we performed $N_e=1000$ experiments, while in the latter, $N_e=100$.

In order to better assess the performance of an algorithm as a function of a particular parameter, not only did we determine the RMSE but also the percentage of successful experiments by checking if the algorithm really found the ML-DOA estimate, since we approximately obtained the location of the global optima of J_{ML} with the aid of a grid search.

Although we shall omit the presentation of the curves of RMSE and the percentage of successful experiments in relation to each parameter of the natural computing algorithms, in the interest of brevity, we would like to briefly expose the most relevant observations.

With regard to the number of individuals N_P, we observed that as N_P increases, the higher are the chances of finding the global optima and the smaller the estimation error becomes, approaching the reference RMSE value given by the grid search. On the other hand, the number of sampled and evaluated points increases, making the computational burden of these algorithms grow. The same behavior was verified relatively to the number of clones N_c, used in CLONALG.

Another interesting remark refers to the mutation step size ρ used in Eq. (12): when ρ is small, the clones tend to suffer more pronounced mutations, as the standard deviation of the Gaussian random variable becomes large; this may not be interesting, since the idea of locally exploring the search space by means of affinity maturation disappears; on the other hand, when ρ is too large, the clones barely change, since the standard deviation of the Gaussian random variable is very small; this may increase the dependence on the position of each individual in the initial population and may force the algorithm to progress really slowly.

Finally, we noticed that the constriction factor χ, as long as it does not become very large, leads to a good performance of PS. The remaining parameters of the algorithms did not show significant influence over the performance.

However, it is important to remind that the parameter set also exert a direct influence over the average number of fitness evaluations (N_{fit}), which, to a certain extent, reflects the computational cost associated with each algorithm. So, this is another concern we must take into account in order to adequately define the set of values for the parameters. Fortunately, it is possible to express N_{fit} as a function of the parameters of each algorithm, as shown in Eq. (17), which offers a valuable aid to the task

of adjusting each parameter.

$$N_{fit} = \begin{cases} N_P + \max_{it} \cdot N_P & \text{for DE,} \\ N_P + \max_{it} N_P N_c + \left\lfloor \dfrac{\max_{it}}{T_{Ab}} \right\rfloor \cdot \lfloor p_{Ab} N_P \rfloor & \text{for CLONALG,} \\ N_P + \max_{it} \cdot N_P & \text{for PS.} \end{cases} \tag{17}$$

Since we are interested in performing a fair comparison between the performance of the algorithms, each parameter set of values must approximately lead to the same value of N_{fit}. Moreover, the number of fitness evaluations should be significantly smaller than that associated with the grid search. With all these observations in mind, the following values were defined for the parameters:

- DE: N_P=150, CR=0.9, F=0.5 and $\max_{it} = 700$.
- CLONALG: N_P=80, N_c=5, $\rho = 2$, $T_{Ab} = 20$, $p_{Ab} = 0.15$ and $\max_{it} = 250$.
- PS: N_P=150, $\chi = 0.729$ and $\max_{it} = 700$.

We can now investigate the performance of each natural computing algorithm in the ML-DOA estimation, and this is the subject of the next section.

5. Simulation results

Firstly, we shall analyze the performance of the natural computing algorithms considering the same scenario used in Section 2.2 and in the sensitivity tests reported in Section 4. Then, we examine whether different conditions of the sources impact the performance of these algorithms. In all simulations, the parameters were adjusted as shown in Section 4.

5.1. First scenario

Fig. 3 shows the performances of DE, CLONALG and PS, along with the RMSE values obtained by MODE and MODEX, for the classical scenario described in Section 2.2. We also display the RMSE values associated with the grid search, which represents an approximation to the performance of the ML estimator, and the Cramér–Rao bound.

As we can observe in Fig. 3, the performance of the three natural computing algorithms are practically identical to that of the grid search. This means that DE, CLONALG and PS were capable of implementing the ML estimator, finding the global optima of J_{ML} in almost

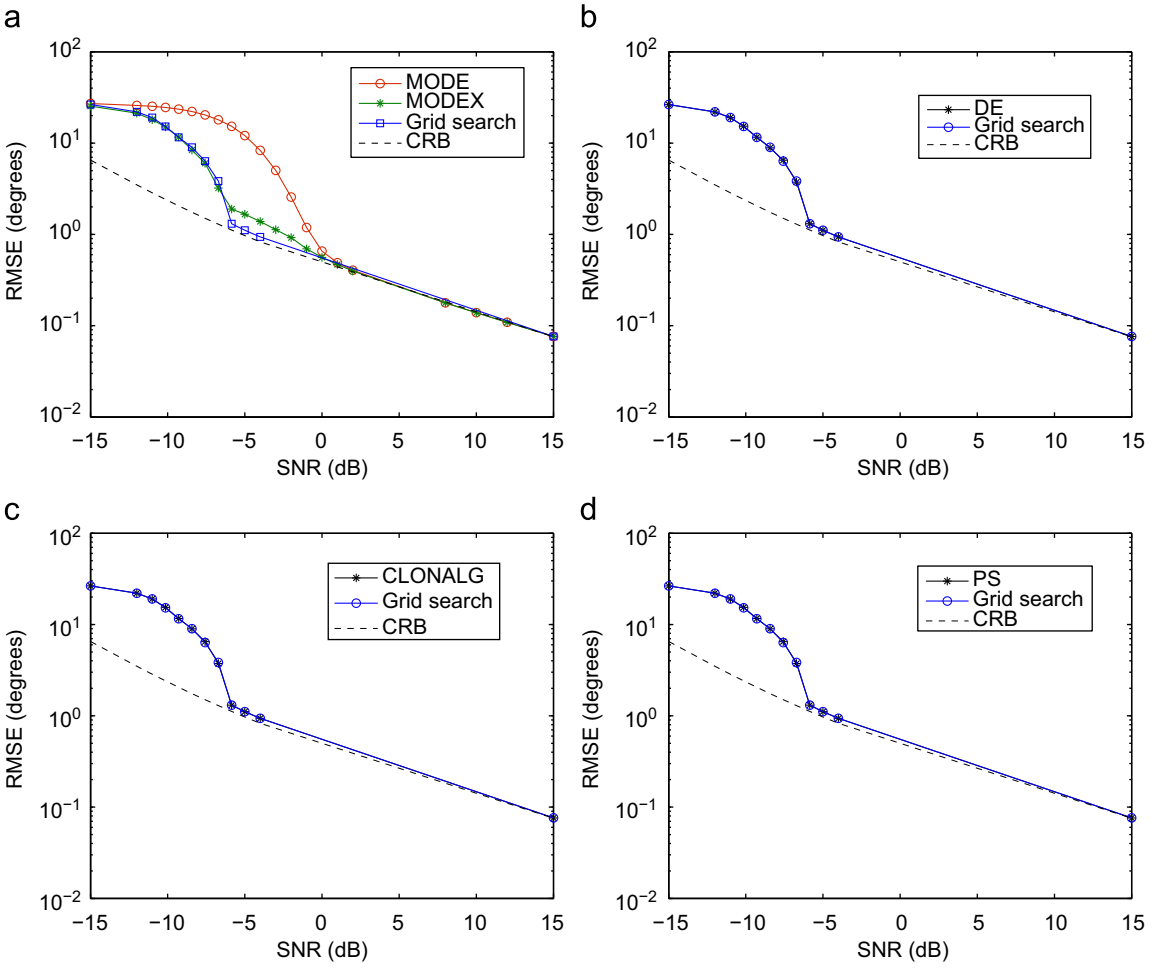

Fig. 3. Performance of the natural computing algorithms. (a) MODE and MODEX. (b) DE. (c) CLONALG. (d) PS.

all experiments. Moreover, these algorithms overcome MODE and MODEX, especially in the threshold domain.

Another important aspect to be highlighted is that DE, CLONALG and PS evaluated a number of points of the search space that is quite inferior to that associated with the grid search. In fact, these algorithms succeeded in estimating the direction of arrival using, approximately, 10^5 candidate solutions, while the grid search required 3.2×10^8 points to achieve the same level of performance.

However, we would like to emphasize that the natural computing algorithms can actually find the global optima of the ML cost function using a number of points that is inferior to the maximum value of 10^5 candidate solutions. In other words, this latter value, obtained via Eq. (17), usually corresponds to a conservative upper bound.

This fact becomes clear when we look at Fig. 4, which displays the average number of points used by DE, CLONALG and PS, until the global optima was found, as a function of the SNR. Basically, we considered that the global optima was found when the angular distance between this point and the best individual within the population became smaller than 0.1°.

It is possible to verify in Fig. 4 that the three natural computing algorithms really located the global optima requiring a number of fitness evaluations that is much smaller than the allowed amount of 10^5 points. For instance, PS was able to reach the global optima using less than 6000 points in all experiments for the whole range of SNR values. So, these methods may be considered to be effective and, in a certain sense, efficient, when applied to DOA estimation.

Therefore, all the evidences presented in this section demonstrate the adequacy of the proposal of using natural computing algorithms to perform the ML-DOA estimation.

5.2. Second scenario

Here, we assess the performance of the natural computing algorithms in cases where the angles assume different values. How good will the natural computing algorithms handle these conditions? In order to answer this question, firstly consider the same DOA scenario used in the previous section, except the direction of arrival vector, which now is given by $\boldsymbol{\theta} = [-10^\circ \; 45^\circ]^T$.

As already evinced in the literature, a higher angular resolution eases the DOA estimation. Hence, we expect that the natural computing algorithms continue to reach the desired performance of the ML estimator in this scenario. Fig. 5 displays the performances of the natural computing algorithms along with MODE, MODEX, grid search and the CRB. Again, $N_e = 1000$ experiments have been carried out for each SNR value.

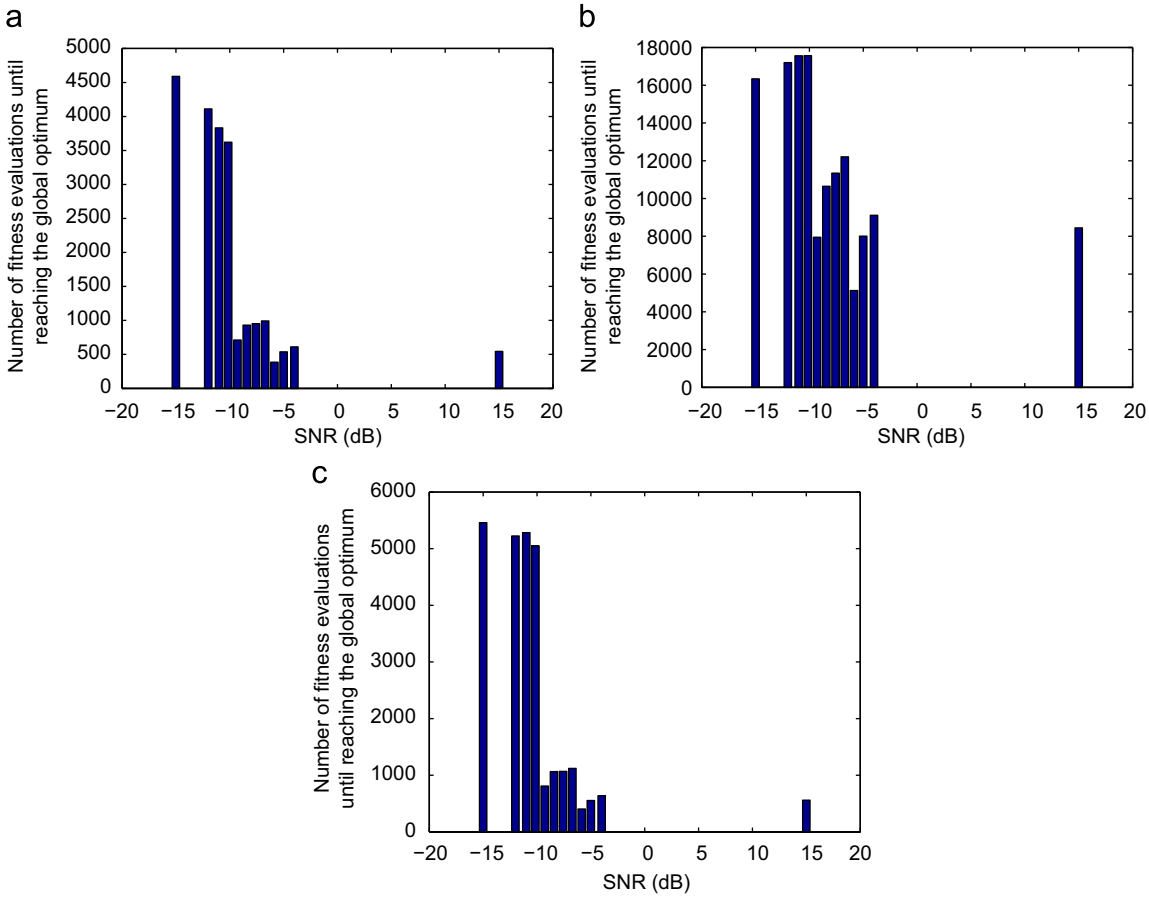

Fig. 4. Number of fitness evaluations versus the SNR. (a) DE. (b) CLONALG. (c) PS.

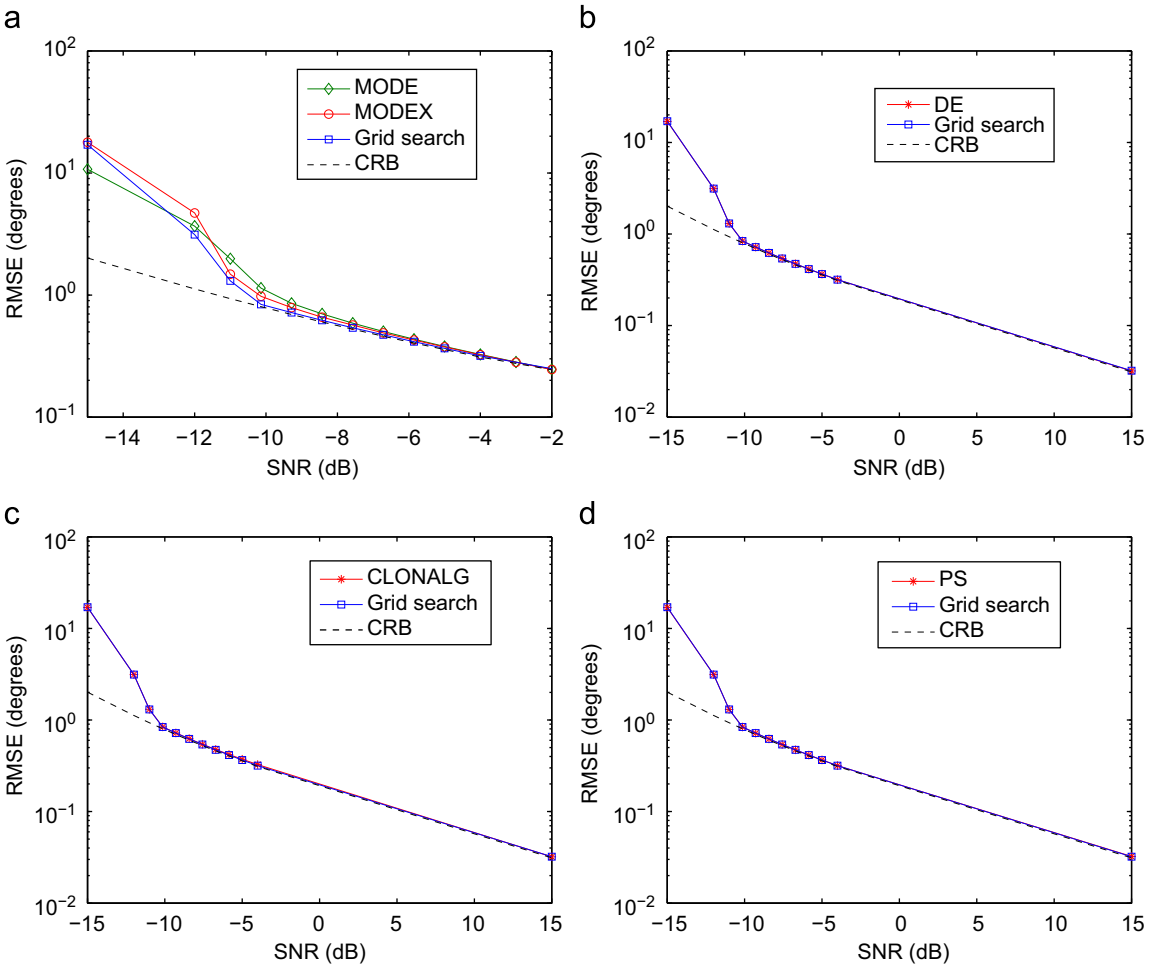

Fig. 5. RMSE curves versus the SNR. (a) MODE and MODEX. (b) DE. (c) CLONALG. (d) PS.

Meeting our expectation, we can observe that the RMSE curves obtained with DE, CLONALG and PS are very close to that associated with the grid search, which implies better performances than those reached via MODE and MODEX.

Finally, we conclude that this analysis considering once more the same DOA scenario used in the previous section, but with $\boldsymbol{\theta} = [70^\circ\ 75^\circ]^T$. Here, the angular resolution is preserved, whereas the angles are closer to 90°. This condition complicates the estimation process, since the discrimination between the respective frequencies becomes more inaccurate as we approach the parallel condition (90°), when, at last, the array stops working.

We present in Fig. 6 the RMSE curves versus the SNR corresponding to DE, CLONALG, and PS. Also, we include the performances of MODE, MODEX and the grid search, as well as the CRB.

As can be seen from Fig. 6, the performances of the three natural computing algorithms are almost equivalent to that related to grid search, with DE suffering a slight divergence for a specific SNR within the threshold domain. Moreover, the three algorithms evaluated a number of solutions significantly smaller than grid search, since no changes have been introduced in their parameters in both cases treated in this section.

Therefore, we obtained promising indicatives that the performances of the natural computing algorithms are not significantly influenced by changes in the true angles. In this sense, the application of these methods in the ML-DOA estimation problem is once more encouraged.

5.3. Third scenario

The experiments performed so far were based on the assumption of independence between the sources. However, it is known that the introduction of correlation may pose additional difficulties to the DOA estimation. For instance, the method MUSIC [46,47] suffers a notable degradation of performance as the correlation coefficient is increased [47].

In order to analyze how the natural computing algorithms cope with the correlated sources situation, we adopted the following scenario: $N=10$, $M=2$, $K=100$, $\boldsymbol{\theta} = [10^\circ\ 15^\circ]^T$. As in [2], the correlation coefficient is equal

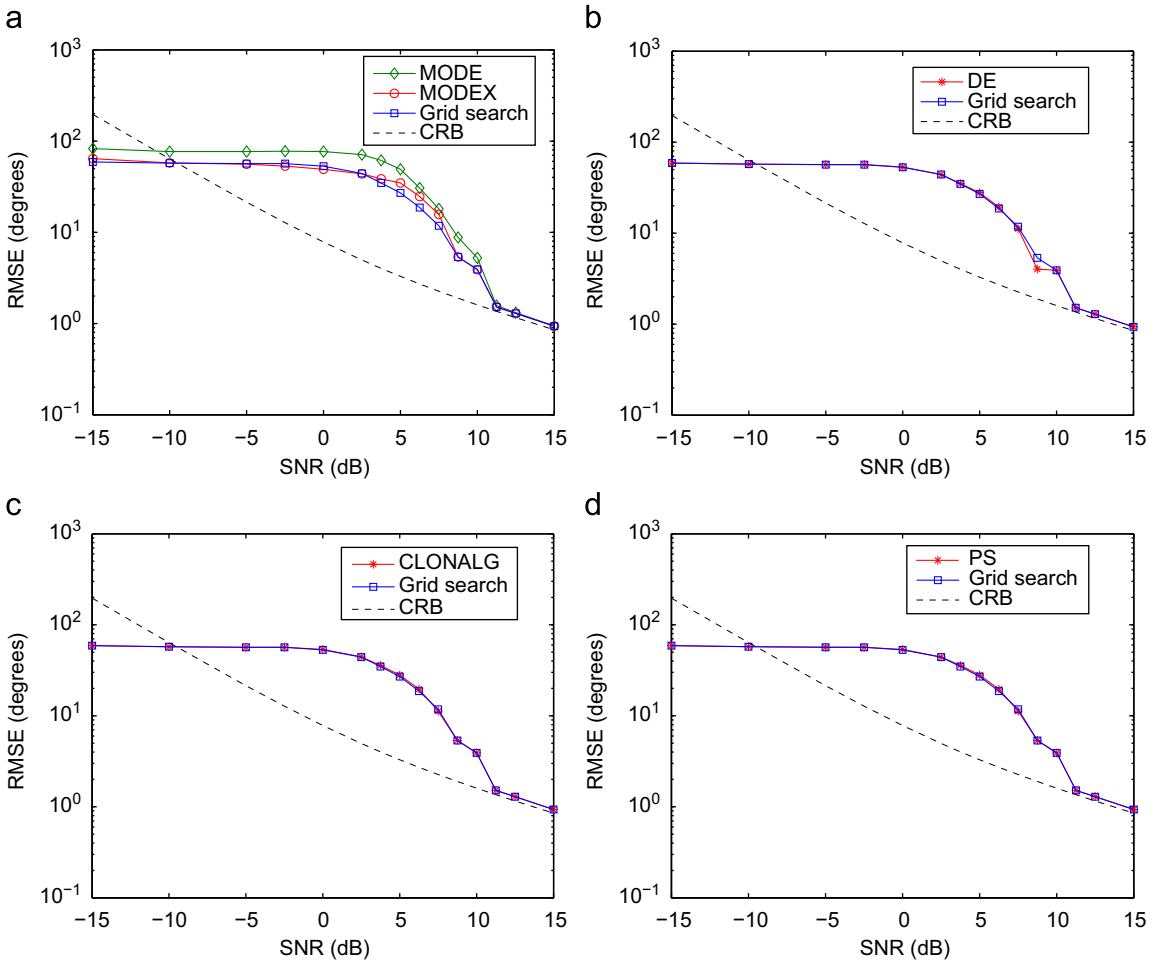

Fig. 6. Performance of the natural computing algorithms. (a) MODE and MODEX. (b) DE. (c) CLONALG. (d) PS.

to 0.98, so the source covariance matrix is given by

$$\mathbf{C} = \begin{bmatrix} 1 & 0.98 \\ 0.98 & 1 \end{bmatrix}.$$

Fig. 7 exhibits the performances of DE, CLONALG and PS, as well as the RMSE curves related to MODE, MODEX, grid search and CRB.

The results shown in Fig. 7 indicate that DE, CLONALG and PS reached performances very similar to that associated with the grid search, overcoming both the MODE and MODEX methods. Hence, we can conclude that the introduction of correlation between the sources does not compromise the DOA estimation performed by these algorithms.

5.4. Fourth scenario

The previous sections were devoted to the study of the performance of the natural computing algorithms in different scenarios that shared an important characteristic: the presence of only two sources impinging on the sensor array. This restriction allowed us to employ a grid search to approximately obtain the location of the global optima of ML cost function. The knowledge about where the global optima really is made possible to verify whether the performance of the considered algorithms actually corresponded to that of the ML estimator.

However, it is necessary to analyze how an increase in the number of sources affects the performance of the natural computing algorithms in DOA estimation. Unlike grid search, we wish that these algorithms continue to correctly estimate the direction of arrival without suffering with the curse of dimensionality.[2]

But, in these situations, we are forced to relinquish the knowledge about the exact location of the global optima, since the grid search cannot be applied anymore, which means that there is no longer a RMSE curve as the reference to be reached. Nonetheless, we still can make inferences and extract indicatives of how good was the DOA estimation based on some expected behaviors related to the ML estimator.

[2] A term coined by Richard Bellman that refers to the problem caused by the exponential increase in volume associated with adding extra dimensions to a mathematical space [48].

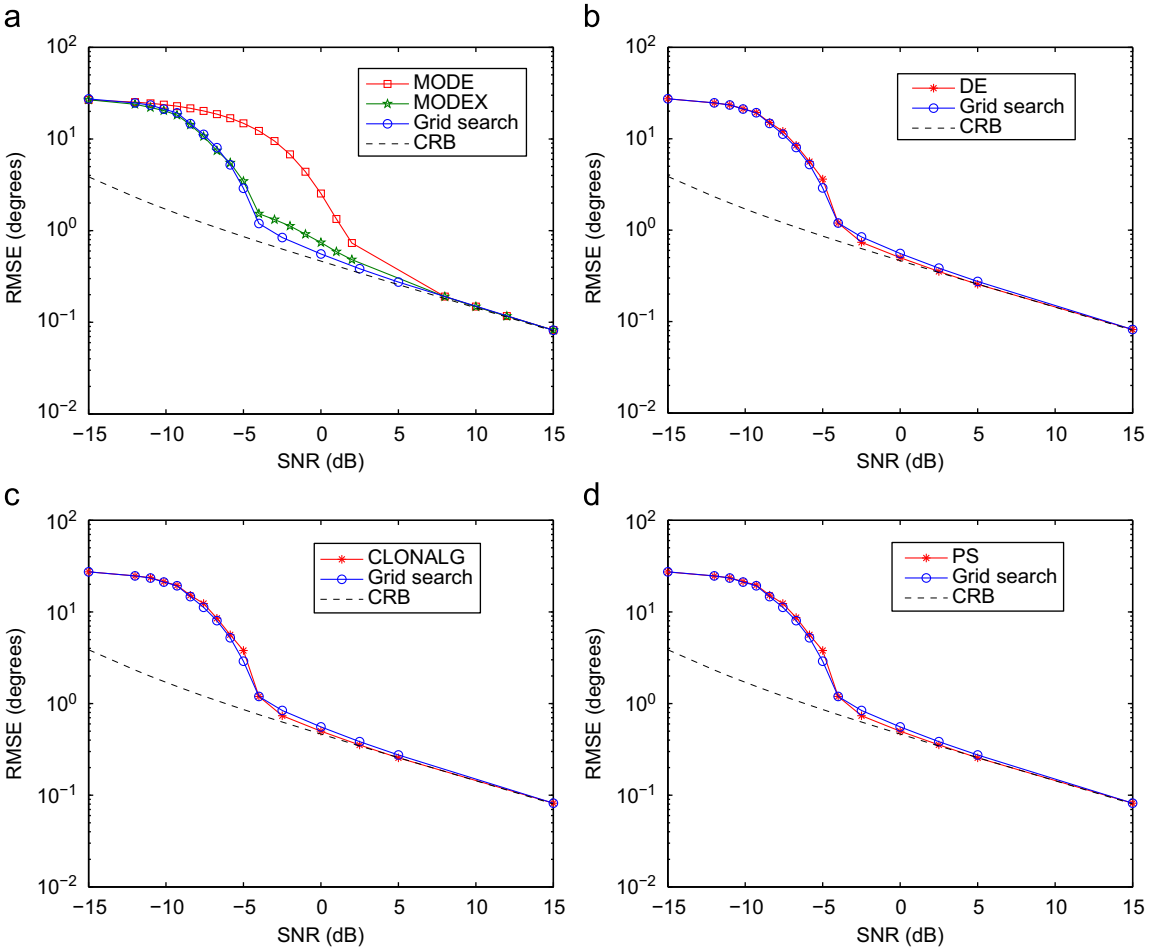

Fig. 7. RMSE curves versus SNR. (a) MODE and MODEX. (b) DE. (c) CLONALG. (d) PS.

Consider the following DOA scenario: $N=10$, $M=3$, $K=500$, $\mathbf{C}=\mathbf{I}$, $\boldsymbol{\theta}=[10^\circ\ 15^\circ\ 20^\circ]^T$. Fig. 8 shows the performances of the three natural computing algorithms employed in this work. It also displays the RMSE curves obtained with MODE, MODEX and the CRB.

Firstly, it is possible to observe that, asymptotically, the performance of DE and PS converged to the CRB. On the other hand, as the SNR decreases, the RMSE values obtained with these algorithms deteriorated with respect to the CRB. However, in the threshold domain, they remained below those related to MODE and MODEX, which suggests that the former algorithms overcome the latter ones in DOA estimation.

But, as mentioned before, we cannot assure that the natural computing algorithms actually reached the performance of the ML estimator. This question becomes more evident if we focus our attention to the case where the SNR is -5 dB: although the RMSE values related to DE and PS are smaller than those of MODE and MODEX, neither can we affirm which of them offers a better performance, nor if any method really achieved the performance of the ML estimator.

Nevertheless, the RMSE curves yielded by DE and PS resemble what we would expect to be the performance of the ML estimator for the whole range of SNR values. Hence, the results shown in Fig. 8 indicate that these natural computing algorithms may be able to adequately estimate the direction of arrival when the number of sources increases.

On the other hand, we can observe in Fig. 8(b) that CLONALG could not reach the expected performance in the asymptotic domain, since the RMSE values obtained with this method were greater than the CRB. But this observation can mislead us into thinking that CLONALG failed to correctly estimate the angles in all experiments, when, actually, the deterioration verified in the performance of CLONALG may be due to only a few experiments in which the algorithm did not locate the global optima of the ML cost function, thus yielding DOA estimates that are more distant to the true angles than such points. This argumentation is corroborated by the histogram of the estimates of the first angle obtained with CLONALG considering the SNR of 15 dB, as shown in Fig. 9. Since the same behavior was observed for the other angles, we decided to omit the respective histograms.

We can notice in Fig. 9 that the vast majority of the estimates is concentrated near the true angle ($\theta_1=10^\circ$), which suggests that CLONALG could find the global

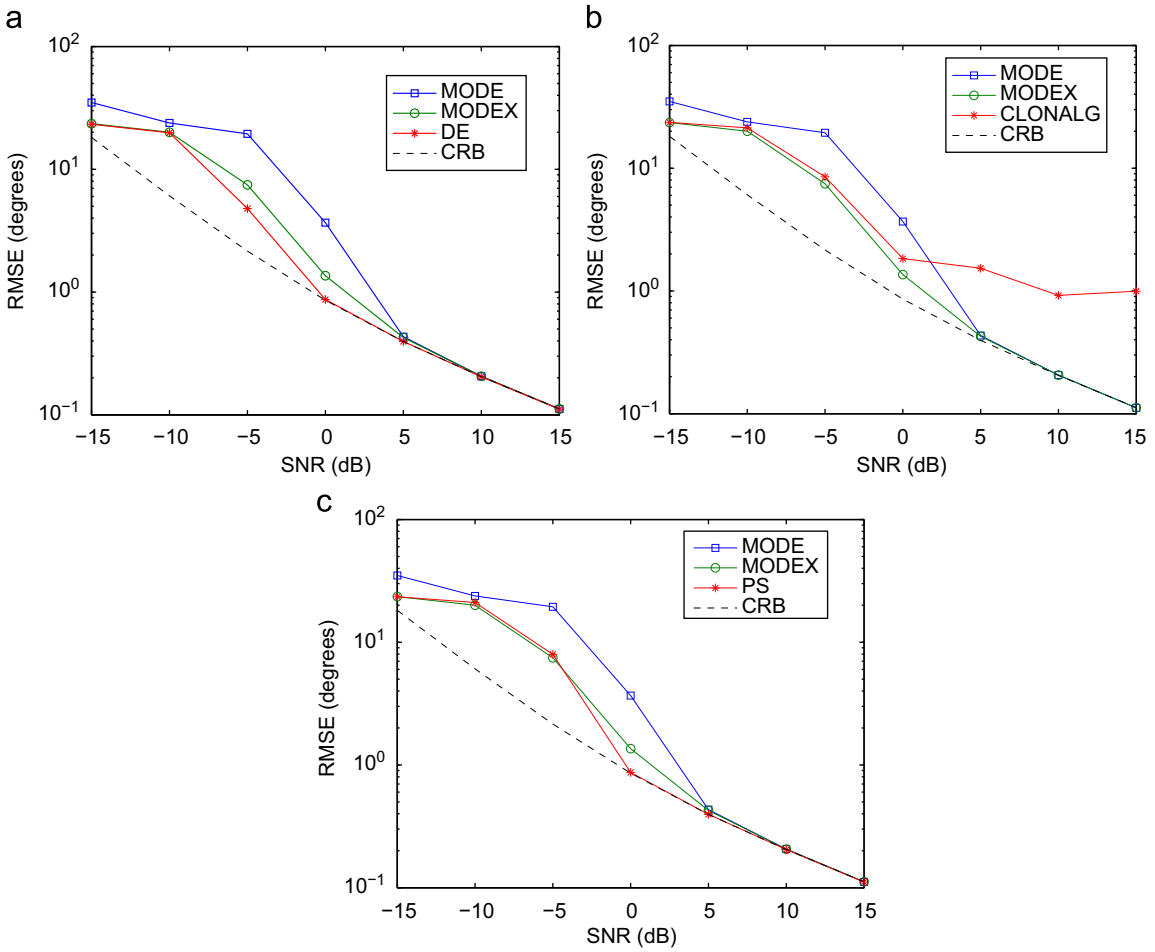

Fig. 8. Performance of the natural computing algorithms. (a) DE. (b) CLONALG. (c) PS.

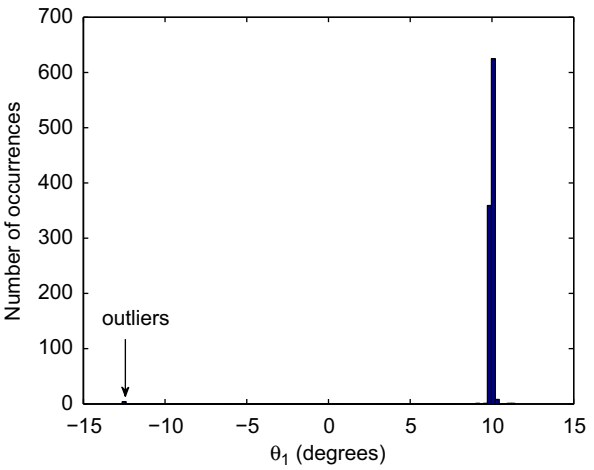

Fig. 9. Histogram of DOA estimates obtained with CLONALG.

optima of the ML cost function in these experiments. However, we can also verify the occurrence of estimates that are really distant to the true angle in a very small number of experiments. Indeed, these outliers occurred in only four of the 1000 experiments realized. Even though, they are the responsible for the observed degradation in the RMSE value associated with CLONALG. Therefore, the histogram highlights the fact that CLONALG correctly found the DOA estimates in almost all experiments. So, the outliers must be seen as extreme situations that occur in a very small number of experiments and do not reflect the real performance of the algorithm in ML-DOA estimation. In this sense, the results obtained with CLONALG, albeit they did not reach the same level of excellence as in the former scenarios, are satisfactory in the vast majority of the experiments, and the application of this algorithm may be still considered pertinent.

Another relevant aspect we must stress is that the natural computing algorithms still used the same values shown in Section 4 for their parameters. This means that, without any increase in the number of fitness evaluations, these methods achieved performances consistent with what we expected for the ML estimator.

To complete the study of the application of natural computing algorithms, consider the following case: $N=10$, $M=4$, $K=2000$, $\mathbf{C}=\mathbf{I}$, $\boldsymbol{\theta}=[10^\circ\ 15^\circ\ 20^\circ\ 30^\circ]^T$. Our intention with this scenario is to verify whether the number of points evaluated during the search for

Table 1
RMSE values.

Method	SNR	
	10 dB	15 dB
DE	0.2135	0.1125
Cramér–Rao bound	0.2066	0.1094

the global optima performed by the natural computing algorithms grows in the same manner as with the grid search.

We decided to focus our analysis on the asymptotic behavior, so that the CRB becomes the performance reference. Besides, we also restricted this study to the differential evolution algorithm for the sake of illustrating this issue. Table 1 displays the RMSE values obtained with DE, along with the CRB, for the SNR values of 10 dB and 15 dB.

We can notice in Table 1 that the RMSE values reached via DE are quite close to those associated with the CRB, which indicates that, at least in the asymptotic domain, DE continued to correctly estimate the angles. Moreover, since DE still employed the parameter values shown in Section 4, the number of points sampled and evaluated by DE is the same as that used in the scenarios with two sources, while the grid search requires a prohibitive amount of points to find the global optima of J_{ML} in this case.

Therefore, it has been shown that the natural computing algorithms can efficiently perform ML-DOA estimation even in the cases where the number of sources is increased.

6. Conclusion

This work addressed the application of meta-heuristics belonging to the research field called Natural Computing to the problem of estimating the direction of arrival of signals impinging on a sensor array. Firstly, we presented the fundamentals of DOA estimation as well as the challenging features of the corresponding optimization problem that arises with the ML approach. Then, we described the three algorithms considered in this paper, DE, CLONALG and PS, and discussed the parameter setting issue, highlighting the existence of several surfaces to be optimized with different characteristics as a consequence of the SNR effect. Finally, we proceeded to the performance analysis considering different DOA scenarios.

The results obtained in this work have shown that these algorithms can deal with the multimodal and signal-to-noise variant nature of the ML cost function, thus achieving adequate performances in all the considered cases. Moreover, these techniques were able to approximately implement the ML estimator using a number of points during the search that is significantly inferior to that required by a grid search. Indeed, we obtained promising indicatives that the natural computing algorithms yield DOA estimates very close to the global optima employing a moderate number of points. It is also pertinent to remark that these techniques have proved not to be sensitive to the angular resolution and source correlation. Besides, the array geometry poses no difficulty to these algorithms.

Nevertheless, there still remains many different aspects to be considered in future works. For example, it may be worth investigating the application of different meta-heuristics to the ML-DOA estimation problem. Another interesting perspective is related to tuning the natural computing algorithms to this particular problem by providing additional information available within the data.

In this context, we have been studying the possibility of employing a noise filtering procedure to aid the natural computing algorithms to locate the global optima of the ML cost function. In a few words, the idea is to collect some knowledge about the true angles by looking at the frequency response of the FIR filter whose coefficients are given by the eigenvector of the covariance matrix $\hat{\mathbf{R}}_{\mathbf{y}}$ associated with its higher eigenvalue [49], and then use this information in the initialization of the repertoire of candidate solutions. The preliminary results are very promising as the algorithms tend to preserve their effectiveness, while they become more efficient, presenting faster convergence towards the global optima, thus requiring fewer fitness evaluations to reach the desired solution [3].

Finally, it seems to be pertinent to study the application of similar algorithms in scenarios where the DOA parameters dynamically change. The work in [50] represents a step towards this perspective and the results obtained there motivate deeper investigations.

Acknowledgments

This work was supported by FAPESP 2008/56937-2, CNPq and CAPES (Brazil).

References

[1] H.L. Van Trees, Optimum Array Processing. Part IV of Detection, Estimation and Modulation Theory, John Wiley & Sons, New York, USA, 2001.
[2] P. Stoica, K.C. Sharman, Novel eigenanalysis method for direction estimation, IEE Proceedings part F (Radar and Signal Processing) 137 (1) (1990) 19–26.
[3] L. Boccato, Aplicação de computação natural ao problema de estimação de direção de chegada, Master's Thesis, Faculdade de Engenharia Elétrica e Computação - UNICAMP, Campinas, SP, 2010.
[4] I. Ziskind, M. Wax, Maximum likelihood localization of multiple sources by alternating projection, IEEE Transactions on Acoustics, Speech and Signal Processing 36 (10) (1988) 1553–1560.
[5] M. Feder, E. Weinstein, Parameter estimation of superimposed signals using the EM algorithm, IEEE Transactions on Acoustics, Speech and Signal Processing 36 (4) (1988) 477–489.
[6] M. Miller, D. Fuhrmann, Maximum-likelihood narrow-band direction finding and the EM algorithm, IEEE Transactions on Acoustics, Speech and Signal Processing 38 (9) (1990) 1560–1577.
[7] J. Fessler, A. Hero, Space-alternating generalized expectation–maximization algorithm, IEEE Transactions on Signal Processing 42 (10) (1994) 2664–2677.
[8] P.J. Chung, J.F. Böhme, DOA estimation using fast EM and SAGE algorithms, Signal Processing 82 (11) (2002) 1753–1762.
[9] Y. Bresler, A. Macovski, Exact maximum likelihood parameter estimation of superimposed exponential signals in noise, IEEE

Transactions on Acoustics, Speech and Signal Processing 34 (1986) 1081–1089.
[10] A.B. Gershman, P. Stoica, New MODE-based techniques for direction finding with an improved threshold performance, Signal Processing 76 (3) (1999) 221–235.
[11] A. Lopes, I.S. Bonatti, P.L.D. Peres, C.A. Alves, Improving the MODEX algorithm for direction estimation, Signal Processing 83 (9) (2003) 2047–2051.
[12] P. Stoica, A.B. Gershman, Maximum-likelihood DOA estimation by data-supported grid search, IEEE Signal Processing Letters (1999) 273–275.
[13] A.B. Gershman, P. Stoica, Data-supported optimization for maximum likelihood DOA estimation, in: Proceedings of the 2000 IEEE Sensor Array and Multichannel Signal Processing Workshop, 2000, pp. 337–341.
[14] K.C. Sharman, G.D. McGlurkin, Genetic algorithms for maximum likelihood parameter estimation, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1989, pp. 2716–2719.
[15] K.C. Sharman, Maximum likelihood parameter estimation by simulated annealing, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1988, pp. 2741–2744.
[16] M. Li, Y. Lu, A refined genetic algorithm for accurate and reliable DOA estimation with a sensor array, Wireless Personal Communications 43 (2007) 533–547.
[17] M. Li, Y. Lu, Improving the performance of GA-ML DOA estimator with a resampling scheme, Signal Processing 84 (2004) 1813–1822.
[18] P. Karamalis, A. Marousis, A. Kanatas, P. Constantinou, Direction of arrival estimation using genetic algorithms, in: Vehicular Technology Conference. VTC 2001 Spring. IEEE VTS 53rd, vol. 1, 2001, pp. 162–166.
[19] L. Boccato, R. Krummenauer, R. Attux, A. Lopes, Um estudo da aplicação de algoritmos bio-inspirados ao problema de estimação de direção de chegada, Revista Controle & Automação 20 (4) (2009) 609–626.
[20] R. Storn, K. Price, Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces, Technical Report TR-95-012, ICSI, 1995.
[21] L.N. de Castro, F.J. Von Zuben, Learning and optimization using the clonal selection principle, IEEE Transactions on Evolutionary Computation 6 (3) (2002) 239–251.
[22] M. Clerc, J. Kennedy, The particle swarm explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (1) (2002) 58–73.
[23] D. Rife, R. Boorstyn, Single tone parameter estimation from discrete-time observations, IEEE Transactions on Information Theory 20 (5) (1974) 591–598.
[24] H. Krim, M. Viberg, Two decades of array signal processing research: the parametric approach, IEEE Signal Processing Magazine 13 (4) (1996) 67–94.
[25] P. Stoica, K.C. Sharman, Maximum likelihood methods for direction-of-arrival estimation, IEEE Transactions on Acoustic, Speech and Signal Processing 38 (7) (1990) 1132–1143.
[26] P. Stoica, A. Nehorai, Performance study of conditional and unconditional direction-of-arrival estimation, IEEE Transactions on Acoustics, Speech, and Signal Processing 38 (10) (1990) 1783–1795.
[27] S.M. Kay, in: Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory, Prentice Hall Signal Processing Series, Englewood Cliffs, NJ, 1993.
[28] P. Forster, P. Larzabal, E. Boyer, Threshold performance analysis of maximum likelihood DOA estimation, IEEE Transactions on Signal Processing 52 (11) (2004) 3183–3191.
[29] J. Ziv, M. Zakai, Some lower bounds on signal parameter estimation, IEEE Transactions on Information Theory IT-15 (3) (1969) 386–391.
[30] A. Weiss, E. Weinstein, Fundamental limitations in passive time delay estimation—Part I: narrow-band systems, IEEE Transactions on Acoustics, Speech, and Signal Processing 31 (2) (1983) 472–486.
[31] L. Atallah, J.-P. Barbot, P. Larzabal, From Chapman–Robbins bound towards Barankin bound in threshold behaviour prediction, Electronics Letters 40 (4) (2004) 279–280.
[32] L.N. de Castro, Fundamentals of Natural Computing: Basic Concepts, Algorithms and Applications, Chapman & Hall, CRC, 2006.
[33] T. Bäck, D. Fogel, Z. Michalewicz, Evolutionary Computation 1: Basic Algorithms and Operators, Institute of Physics Publishing, Bristol, UK, 2000.
[34] R.R.F. Attux, M.B. Loiola, R. Suyama, L.N. de Castro, F.J. Von Zuben, J.M.T. Romano, Blind search for optimal wiener equalizers using an artificial immune network model, EURASIP Journal of Applied Signal Processing 2003 (6) (2003) 740–747.
[35] D.H. Wolpert, W.G. Macready, No Free Lunch Theorems for Search, Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995.
[36] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (2) (1997) 67–82.
[37] Z. Michalewicz, D.B. Fogel, in: How To Solve It: Modern Heuristics, 2nd ed., Springer, 2004.
[38] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.
[39] K. Price, R. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer, 2005.
[40] F.M. Burnet, Clonal Selection Theory of Acquired Immunity, Cambridge University Press, 1959.
[41] G.L. Ada, G.J.V. Nossal, The Clonal Selection Theory, Scientific American, 1987, pp. 50–57.
[42] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks,vol. 4, 1995, pp. 1942–1948.
[43] J. Kennedy, The particle swarm: social adaptation of knowledge, in: IEEE International Conference on Evolutionary Computation, 1997, pp. 303–308.
[44] J. Kennedy, R. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufmann Publishers, 2001.
[45] J. Kennedy, Particle swarms: optimization based on sociocognition, in: L.N. de Castro, F.J. Von Zuben (Eds.), Recent Developments in Biologically Inspired Computing, Idea Group Publishing, 2004, pp. 235–269 (Chapter X).
[46] R.O. Schmidt, A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation, Ph.D. Thesis, Stanford University, 1981.
[47] R.O. Schmidt, Multiple emitter location and signal parameter estimation, IEEE Transactions on Antennas and Propagation AP-34 (3) (1986) 276–280.
[48] R.E. Bellman, Dynamic Programming, Princeton University Press, 1957.
[49] R. Krummenauer, M. Cazarotto, A. Lopes, P. Larzabal, P. Forster, Improving the threshold performance of maximum likelihood estimation of direction of arrival, Signal Processing 90 (5) (2010) 1582–1590.
[50] L. Boccato, F.O. de França, R. Krummenauer, R. Attux, F.J. Von Zuben, A. Lopes, Otimização dinâmica imuno-inspirada para o problema de estimação de direção de chegada, in: Anais do XXVII Simpósio Brasileiro de Telecomunicações, 2009.