

Research Article

General Improvements of Heuristic Algorithms for Low Complexity DOA Estimation

Haihua Chen ¹, Haoran Li,² Mingyang Yang,³ Changbo Xiang,³ and Masakiyo Suzuki⁴

¹College of Ocean and Spatial Information, China University of Petroleum, Qingdao, Shandong 266580, China

²Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Science, Beijing, China

³National Key Laboratory of Science and Technology on Electronic Test and Measurement,

The 41st Research Institute of China Electronic Science and Technology Group Corporation, Qingdao, China

⁴Graduate School of Engineering, Kitami Institute of Technology, Kitami, Hokkaido 090-8507, Japan

Correspondence should be addressed to Haihua Chen; chenhaihua@upc.edu.cn

Received 11 October 2019; Accepted 25 November 2019; Published 11 December 2019

Academic Editor: Sotirios K. Goudos

Copyright © 2019 Haihua Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Heuristic algorithms are considered to be effective approaches for super-resolution DOA estimations such as Deterministic Maximum Likelihood (DML), Stochastic Maximum Likelihood (SML), and Weighted Subspace Fitting (WSF) which are involved in nonlinear multi-dimensional optimization. Traditional heuristic algorithms usually need a large number of particles and iteration times. As a result, the computational complexity is still a bit high, which prevents the application of these super-resolution techniques in real systems. To reduce the computational complexity of heuristic algorithms for these super-resolution techniques of DOA, this paper proposes three general improvements of heuristic algorithms, i.e., the optimization of the initialization space, the optimization of evolutionary strategies, and the usage of parallel computing techniques. Simulation results show that the computational complexity can be greatly reduced while these improvements are used.

1. Introduction

The estimation of directions-of-arrival (DOA) is a basic and central issue in sensor array signal processing, and it can be applied in many fields such as radar, sonar, communications, and so on. According to the DOA estimation problem, although now more and more attention is attracted by some specific cases for special signals, noises, and application scenarios [1–3], the basic model of DOA is still the far-field narrow-band DOA estimation which is the issue to be dealt with in this context.

For far-field narrow-band signal DOA estimation, there are two processes in general. The first step is to get the criteria of DOA estimation according to different algorithms from observed data. So far, a set of algorithms have been proposed, such as Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [4], Multiple Signal Classification (MUSIC) [5, 6], Maximum Likelihood

(ML, including Deterministic ML (DML) [7] and Stochastic ML (SML) [8, 9]), compressed sensing [10, 11], and Weighted Subspace Fitting (WSF) [12]. Among these algorithms, it is well known that ML and WSF have the highest accuracy of DOA estimation [13–15]. However, it is also well demonstrated that the computational complexity of them is also the highest because the criteria of them are nonlinear multi-dimensional optimization [13–16].

The second step is how to get the DOAs from the criteria, i.e., the estimation process. Among the algorithms mentioned above, except for ESPRIT, the other algorithms are involved in optimization problems. MUSIC involves one-dimensional optimization, and ML (DML, SML) and WSF involve nonlinear multi-dimensional optimization. Unlike the one-dimensional optimization, the computational complexity of the estimation of nonlinear multi-dimensional optimization is usually extremely high. That is also the reason why DML, SML, and WSF could not be applied in

real systems [17, 18] in spite of their excellent performance in accuracy. This paper attempts to reduce the computational complexity of the estimation of these algorithms so that they can potentially be applied in practical systems.

To promote the nonlinear multi-dimensional optimization, usually there are two categories of solving algorithms. The first type is Neural Networks [19]. Usually, this kind of algorithm needs a large number of training samples. The second type is heuristic algorithms for simulating biological behaviors in nature such as Genetic Algorithm (GA) [20], Ant Colony Algorithm [21], and Particle Swarm Optimization (PSO) algorithm [16, 22, 23]. These algorithms are all population-based iterative techniques. To avoid falling into local optimum, these algorithms usually need a large number of particles which are randomly initiated in the whole solution space. In the convergence process of these particles, a large number of iterations are also needed. As a result, the computational complexity of these algorithms is still a bit high. This paper is trying to reduce the computational complexity of heuristic algorithms for the estimation of DML, SML, and WSF which involve nonlinear multi-dimensional optimization.

To reduce the computational complexity of heuristic algorithms for DOA estimation, this paper proposes three improvements. The first improvement is to optimize the initialization space. Traditional algorithms usually randomly initiate a large number of particles in the whole solution space, which is to ensure the diversity of the population. However, these randomly initiated particles are too scattered and far from the optimum. Then, a large number of iteration times are also needed. As a result, the computational complexity is high. This paper proposes a two-stage optimization strategy. In the first step, find an initial value which is near around the optimum through a low computational complexity algorithm, such as ESPRIT. Then, an optimized initialization space can be constructed around it. Since the initialization space is just around the optimum, less particles and iteration times are needed. Simulation results show that this improvement can greatly reduce the number of initial particles and iteration times. As a result, computational complexity can be reduced. The second improvement is to optimize the evolutionary strategy. Generally, one algorithm has only one evolutionary strategy. For example, the main characteristics of the genetic algorithm are heredity, crossover, and mutation. The evolutionary strategy determines the process of convergence of all the particles. Simple evolutionary strategy usually causes a long converge process. In this paper, we propose to fuse the evolutionary strategies of both GA and PSO so that the updating of the particles is affected not only by heredity, crossover, and mutation but also by local and global optimum in the evolutionary process. Simulation results show that this improvement can reduce the number of iteration times. As a result, the computational complexity can be reduced. The third improvement is the usage of Parallel Computing techniques. Obviously, the iterative process satisfies the condition that Single Program uses Multiple Data (SPMD), so it can be applied to parallel computing. To be honest, this

improvement is not a theoretical innovation, but it is very effective in practical calculation.

Note that these improvements of heuristic algorithms are general and independent to each other, and they can be applied to DML, SML, or WSF estimation of DOA alone or together. In this paper, we apply them to the SML estimation of DOA as an example. Simulation results show that these improvements can greatly reduce the computational complexity compared to the original algorithms.

The rest of this paper is organized as follows. In Section 2, we introduce the problem of DOA and the formulation of the SML algorithm. In Section 3, we show the improvements of heuristic algorithm. Simulation results are shown in Section 4, and conclusion is drawn in Section 5.

2. System Model and Problem Formulations

The basic model of far-field narrow-band DOA estimation is depicted as shown in Figure 1. Suppose that there is an array with m sensors. All the sensors are not coupled and are omnidirectional. Also, there are n narrow-band sources far from the array. All the sources with a known frequency impinge on the array from distinct directions $\theta_1, \theta_2, \dots, \theta_n$, $n < m$, with respect to a reference point, respectively. Note that the array configuration can be arbitrary, in the real system, though it is usually set to be a specific array configuration such as a uniform linear array or a uniform circle array. The sources can be coherent [8, 12] which happens, for example, in multipath propagations.

2.1. Problem Formulation of DOA. Using the basic model depicted above, the p -dimensional output vector received by the array is

$$\begin{aligned} \mathbf{z}(t) &= \mathbf{A}(\Theta)\mathbf{x}(t) + \mathbf{y}(t), \\ \mathbf{A}(\Theta) &= [a(\theta_1)a(\theta_2)\cdots a(\theta_n)], \\ \mathbf{a}(\theta) &= [a_1(\theta)e^{-j\omega_0\tau_1(\theta)}, \dots, a_p(\theta)e^{-j\omega_0\tau_p(\theta)}]^T, \end{aligned} \quad (1)$$

where $\mathbf{x}(t)$ is the signal vector, $\mathbf{y}(t)$ is the noise vector, and $\mathbf{A}(\Theta)$ is the steering vector in which $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$. The superscript T denotes the transpose of a matrix. $a_i(\theta)$ is the amplitude response of the i -th sensor to a wave front impinging from the direction θ . $\tau_i(\theta)$ is the propagation delay between the i -th sensor and the reference point. Then, take N snapshots of the observed data:

$$\mathbf{Z} = [z(t_1)z(t_2)\cdots z(t_N)]. \quad (2)$$

The DOA estimation is to get

$$\hat{\Theta} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_n\}, \quad (3)$$

for given observed data \mathbf{Z} .

2.2. SML Algorithm. The SML criterion is shown as follows. For more detailed derivation, one should refer [8, 9].

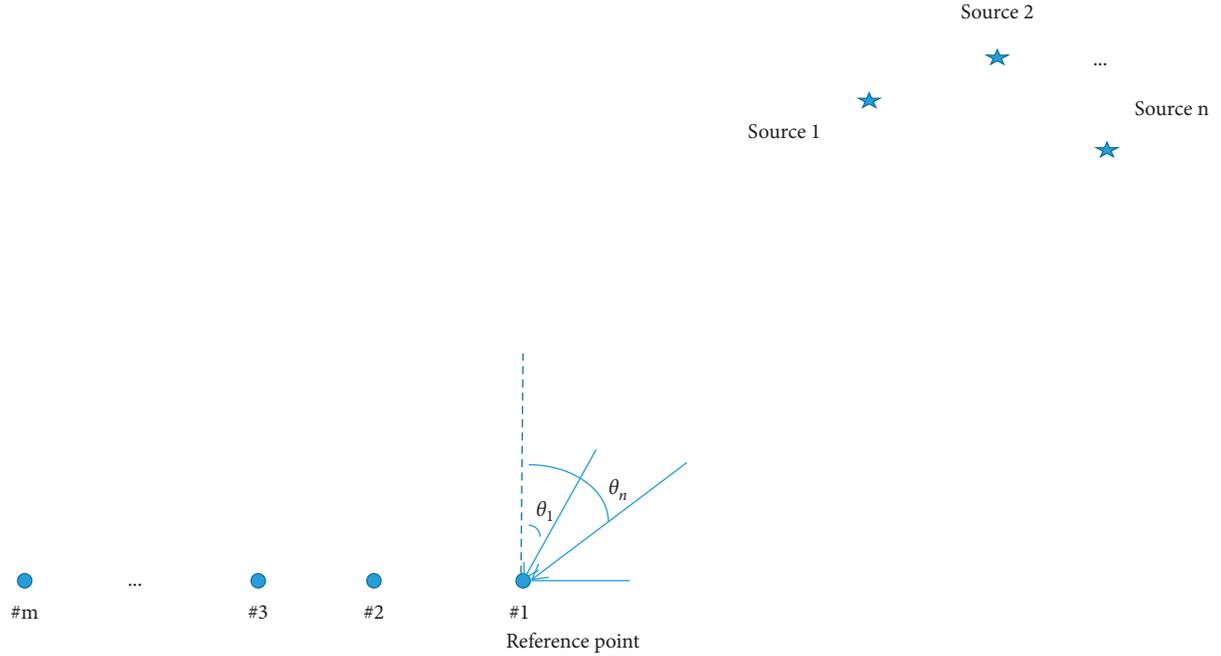


FIGURE 1: The solution of SML, ESPRIT, and the rough search DOA.

$$\hat{\Theta}_S = \arg \min_{\Theta} L_S(\Theta), \quad (4)$$

$$L_S(\Theta) = \det\{R_S\} \times \left(\frac{1}{m-n} \text{tr}\{R_N\} \right)^{m-n}, \quad (5)$$

$$R_N = T_N^H(\Theta) \hat{R} T_N(\Theta), \quad (6)$$

$$R_S = T_S^H(\Theta) \hat{R} T_S(\Theta), \quad (7)$$

$$\hat{R} = \frac{1}{N} Z Z^H. \quad (8)$$

In the above, $()^H$ denotes the complex conjugate transpose. \hat{R} is the sample covariance matrix of the observed data Z . $T_S(\Theta)$ is an $m \times n$ matrix composed of an orthonormal system of the signal subspace spanned by $A(\Theta)$. $T_N(\Theta)$ is an $m \times (m-n)$ matrix composed of an orthonormal system of the noise subspace. $\hat{\Theta}_S$ is the estimated vector of directions of signals. To get $\hat{\Theta}_S$, it involves a multi-dimensional nonlinear optimization problem as shown in formula (4) which is the SML criteria.

3. General Improvements of Heuristic Algorithms

Heuristic algorithms simulating biological behaviors in nature such as Genetic Algorithm (GA) [20], Ant Colony Algorithm [21], and Particle Swarm Optimization (PSO) algorithm [22, 23] are considered to be effective approaches for multi-dimensional nonlinear optimization problems. These algorithms are all population-based iterative techniques. To further understand the characteristics of heuristic

algorithms, we take the PSO algorithm for SML estimation as an example.

The core idea of the PSO algorithm is that in the updating process, each particle is affected by both the local best position found by itself and the global best position found by the swarm until the optimum solution is found. It can be applied to SML estimation as follows:

- (1) Randomly initiate a number of particles in the whole solution space. The solution space is a set of all possible directions of n signals. For example, for a uniform linear array, possible incident angles range from -90 degree to 90 degree for each signal. Therefore, the solution space is huge especially when the signal number n is large. To ensure the diversity and find the unique solution of SML effectively in the huge solution space, the population size is usually set to be relatively large, possibly 30 or more.
- (2) Each particle uses the local best position found by itself and the global best position found by the swarm for updating. The smaller it makes the value of formula (5), the better its position.
- (3) Continue in step 2 until all the particles do not move or the maximum iteration number is attained.

To quantitatively understand the computational complexity, we did simulation for the PSO algorithm for SML with comparison to MUSIC and ESPRIT as shown in Table 1. The scenario is that $m = 4$, $n = 2$, $N = 100$, and $\text{SNR} = 0$ dB, and all the signals are independent. The Monte Carlo is 100 times.

From Table 1, it is obvious that the computational complexity of ESPRIT is the smallest because it can calculate the directions explicitly [4]. The computational complexity

TABLE 1: Comparison of computational complexity of ESPRIT, MUSIC, and PSO applied for SML for one time DOA estimation.

	PSO for SML	MUSIC	ESPRIT
Population size	30	—	—
Average iteration times	128	—	—
Total calculating time (seconds)	0.45	0.011	0.0017

of MUSIC is also acceptable because it just involves in one-dimensional optimization [5]. The computational complexity of the estimation of PSO for SML is the highest because it is a multi-dimensional nonlinear optimization problem.

To get the solution of SML while using the PSO algorithm, 30 particles take 128 iteration times, i.e., calculating the criteria $L_S(\Theta)$ in formula (5) $128 * 30 = 3840$ times. To evaluate $L_S(\Theta)$, for a given particle whose position is a vector of directions, for example $\hat{\Theta} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_n\}$, the evaluation process is as follows. (1) Evaluate steering vector $\mathbf{A}(\hat{\Theta})$; (2) evaluate the observation data \mathbf{Z} and the sample covariance matrix of the observed data $\hat{\mathbf{R}}$; (3) evaluate $\mathbf{T}_S(\hat{\Theta})$, $\mathbf{T}_N(\hat{\Theta})$, and \mathbf{R}_S , \mathbf{R}_N according to formulas (6) and (7); and (4) evaluate $L_S(\hat{\Theta})$. It is clear that to get $L_S(\hat{\Theta})$, a set of matrix operations are needed. Since there are too many particles and a large number of iteration times, the computational complexity is high.

Through the analysis of the PSO algorithm, the following defects are revealed:

- (1) The initialization space which is equivalent to the whole solution space is too large. The randomly initialized particles are too scattered and are far away from the optimal solution. As a result, a relatively large number of particles and iteration times are needed for convergence.
- (2) The evolutionary strategy is too simple, causing a long convergence process.

To overcome these defects and reduce computational complexity further, this paper proposes three general improvements of heuristic algorithms.

3.1. Improvements of Initialization Space. According to the first defect, if we can find a value which is just near around the solution of SML, then all the particles are randomly initiated around this value. It can be predicted that less particles and less iteration times are needed for these particles to find the solution of SML. The computational complexity can be reduced consequently. Therefore, the key problem is how to find such a value which is close to the solution of SML with extremely low computational complexity. Here, we provide two methods.

3.1.1. The First Method. Apply the ESPRIT algorithm to the observed data \mathbf{Z} to get the solution of ESPRIT, $\hat{\Theta}_{ES}$. Although it is well known that the estimation accuracy of ESPRIT is much lower than that of SML, it is the fact that the solution of ESPRIT is near around the solution of SML.

Furthermore, the computational complexity of ESPRIT is extremely low because it can calculate the DOAs explicitly. Therefore, the solution of ESPRIT provides a rather good initial value.

Note that the classic ESPRIT algorithm requires that all the signals are independent and the array configuration is uniform linear array. If there are coherent signals, the preprocessing techniques such as the spatial smoothing [24] or matrix reconstruction [25] method are needed. If the array configuration is not a uniform linear array, this method is not valid.

3.1.2. The Second Method. This method provides a rough search DOA result, and it does not need to consider the array configuration. It is a hypothesis technique. Let $L_S(\hat{\Theta}_{(l)})$ be a cost function of $L_S(\Theta)$ in (5) where l is the assumed number of signals and $\hat{\Theta}_{(l)} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_l\}$.

- (1) Suppose that $l = 1$ and get the cost function $L_S(\hat{\Theta}_{(1)})$. Find $\hat{\theta}_1$ which minimizes $L_S(\hat{\Theta}_{(1)})$.
- (2) Suppose that $l = 2$ and $\hat{\theta}_1$ is fixed as calculated above. Find $\hat{\theta}_2$ which minimizes $L_S(\hat{\Theta}_{(2)})$.
- (3) Continue in this way until all the directions $\{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_n\}$ are calculated.

Note that the validity of this method was demonstrated in [7] which provides a rather good initial value for the AM (Altering Minimization) algorithm [7]. The AM algorithm is also an iterative technique for solving a nonlinear multi-dimensional minimization problem. It contains two parts. The first part is the initialization phase, which is the same as what we have presented above. The second part is the convergence phase. In the convergence phase, at each updating process, let one parameter be variable and let all other parameters be held fixed. Find the value of this variable parameter minimizing the criterion of the corresponding cost function by one-dimensional global search. Fix this value and then let another parameter be variable. Continue in this fashion until all the parameters are converged. The core idea of this algorithm is that in each updating process, this is the only parameter.

Simulation results show that this approach also provides a rather good initial value which is close to the solution of SML, and it does not need to consider coherent signals and array configuration. For consistency, we also name this rough search DOA value as $\hat{\Theta}_{ES} = \{\hat{\theta}_{es1}, \hat{\theta}_{es2}, \dots, \hat{\theta}_{esn}\}$.

After getting $\hat{\Theta}_{ES}$, we define an empirical function as a “scale” to construct the initialization space:

$$T(\text{SNR}) = \begin{cases} \frac{2}{e^{\text{SNR}/10}}, & \text{all the signals are independent,} \\ \frac{5}{e^{\text{SNR}/10}}, & \text{there are coherent signals.} \end{cases} \quad (9)$$

This “scale” is only related to SNR. It is smaller for the noncoherent case or when SNR is higher. That is because $\hat{\Theta}_{ES}$ is more close to the solution of SML when all the signals

are independent or SNR is higher, which was well demonstrated in many literatures. The initialization space is then defined as a set of $\tilde{\Theta}$ in which

$$\begin{aligned}\hat{\theta}_1 &\in [\hat{\theta}_{es1} - T(\text{SNR}_1), \hat{\theta}_{es1} + T(\text{SNR}_1)], \\ \theta_2 &\in [\hat{\theta}_{es2} - T(\text{SNR}_2), \hat{\theta}_{es2} + T(\text{SNR}_2)], \dots, \\ \theta_n &\in [\hat{\theta}_{esn} - T(\text{SNR}_n), \hat{\theta}_{esn} + T(\text{SNR}_n)],\end{aligned}\quad (10)$$

where SNR_i is the SNR of the i -th signal. The best result of this improvement is that the initialization space is not too large and just contains the solution of SML. Note that even if the solution of SML is not included in the initialization space, it will not have any effect on the effectiveness of the algorithm.

The validity of the improvement of the initialization space will be demonstrated in the section of simulation.

3.2. Improvement of Evolutionary Strategy. Usually, one heuristic algorithm has one evolutionary strategy. As we have analyzed above, simple evolutionary strategy may result in a long convergence process. Here, we provide a fusion of the genetic algorithm and the PSO algorithm. The proposed algorithm combines the characteristics of cross-mutation of the genetic algorithm, and its particle is also affected by the local best position found by itself and the global best position found by the swarm.

The proposed algorithm applied for SML estimation can be described as follows:

- (1) Randomly initiate p particles in the initialization space. Define $\tilde{\Theta} = \{\tilde{\Theta}_1, \tilde{\Theta}_2, \dots, \tilde{\Theta}_p\}$, where $\tilde{\Theta}_i = (\theta_{i1}, \theta_{i2}, \dots, \theta_{in})$ is the position of the i -th particles.
- (2) Let k denote the k -th iteration. $\tilde{\Theta}_i^k$ represents the i -th particle's position at the k -th iteration. Let F_i^k and G^k be the local best solution found by the i -th particle and the global best solution found by all the p particles at the k -th iteration, respectively.

$$\begin{aligned}F_i^0 &= \tilde{\Theta}_i^0, \\ F_i^1 &= \arg \min_{\tilde{\Theta}_i^0, \tilde{\Theta}_i^1} L_S(\Theta), \dots, \\ F_i^k &= \arg \min_{\tilde{\Theta}_i^{k-1}, \tilde{\Theta}_i^k} L_S(\Theta), \\ G^k &= \min\{F_1^k, F_2^k, \dots, F_p^k\},\end{aligned}\quad (11)$$

where $L_S(\Theta)$ is the cost function of SML criteria defined in equation (5).

- (3) The position of the i -th particle at the $(k+1)$ -th iteration is updated as follows:

$$\tilde{\Theta}_i^{k+1} = \tilde{\Theta}_i^k + \tilde{V}_i^{k+1}, \quad (12)$$

where

$$\tilde{V}_i^{k+1} = w\tilde{V}_i^k + h_1e_1(F_i^k - \tilde{\Theta}_i^k) + h_2e_2(G^k - \tilde{\Theta}_i^k). \quad (13)$$

\tilde{V}_i^k is the moving speed. w is the inertia factor. h_1 and h_2 are self- and swarm confidence factors, respectively. e_1 and e_2 are random value in $(0, 1)$. Generally,

$w = 0.5$, $h_1 = h_2 = 2$, $e_1 = e_2 = 0.5$, and $\tilde{V}_i^0 = 0.5$. [22, 23].

- (4) Select the best $\lfloor p/2 \rfloor$ particles from above. The smaller it makes the value of the cost function, the better its position. Then, change the value of each direction of each particle into a binary code.
- (5) Randomly select two particles selected in step 4 (for example, A and B) to make a new particle (for example, C) as follows. To generate C, the crossover and mutation processes are necessary. The crossover process is like using a part of the binary code of one direction of A exchanges with the corresponding binary part of the same direction of B to generate the new direction of C. The mutation process is like that each binary bit of one direction of C can be changed to a new value (for example, 0 becomes 1) with a certain probability. Note that the crossover process is also probabilistic. Usually, the crossover and mutation probability are set to be 0.9 and 0.01, respectively. Use $\lfloor p/2 \rfloor$ particles to generate p particles under this way. Then, change the binary code to the normal value (decimal).
- (6) Go to step 3 until the moving speed is zero or when the number of maximum iteration times is achieved.

Note that there are literatures discussing the value of parameters in GA and PSO algorithms [22, 23]. In order to make the paper more compact, these parts are omitted. Meanwhile, this proposed algorithm is just a fusion of GA and PSO, and its effectiveness can be demonstrated in simulations. Readers can try some other fusion of different heuristic algorithms.

3.3. Usage of Parallel Computing Technologies. From the analysis above, it is clear that the most of the computational cost of heuristic algorithms focuses on the iteration part. Fortunately, this part is Single Program with Multiple Data (SPMD). Therefore, parallel computing technologies can be used for heuristic algorithms. Although the parallel computing technology highly depending on the hardware of the system cannot reduce computational complexity, it can reduce the computation time, which is also important in real systems.

MATLAB supports a variety of parallel structures, including "parfor," "SPMD," and so on. Here, we recommend SPMD because it is more flexible. To be honest, this improvement is not a theoretical innovation, but it is very effective in practical calculation.

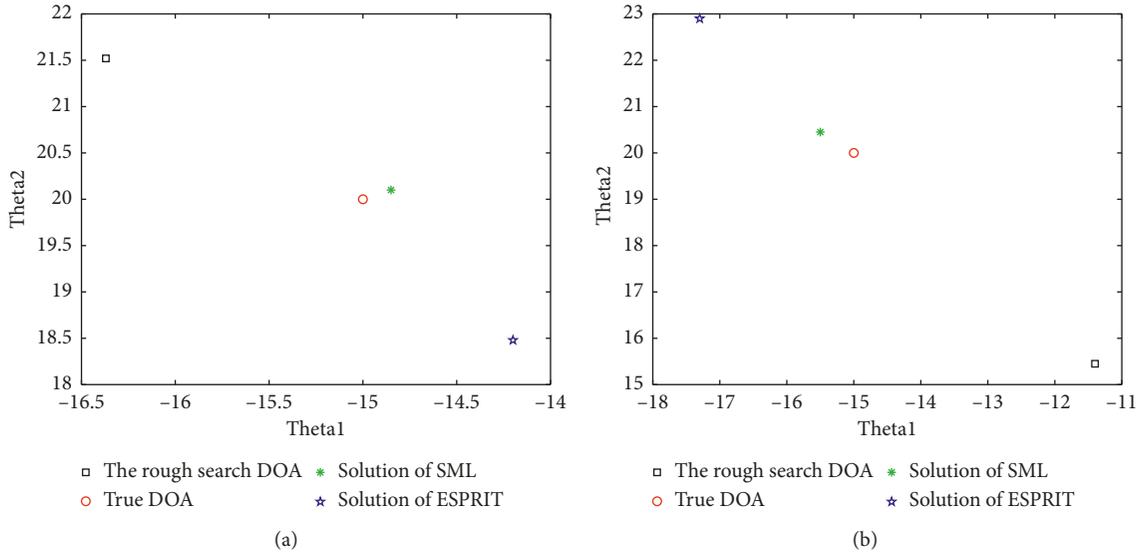


FIGURE 2: The solution of SML, ESPRIT, and the rough search DOA. (a) Noncoherent case: all the signals are independent. (b) Coherent case: two signals are fully correlated.

4. Simulations

In this section, the validity of three improvements of heuristic algorithms is demonstrated. These improvements are general and independent to each other. They can be used alone or together.

The simulations are done using “Matlab” in an ordinary laptop. The signal-to-noise ratio (SNR) is defined as follows:

$$\text{SNR}_k = 10 \log_{10} \frac{E[|s_k(t)|^2]}{\sigma^2}, \quad (14)$$

where $s_k(t)$ is the k -th signal and σ^2 is the power of noise. The root mean square error (RMSE) is defined as

$$\text{RMSE} = \sqrt{\frac{1}{nM} \sum_{k=1}^n \sum_{l=1}^M |\hat{\theta}_{k,l} - \theta_k|^2}, \quad (15)$$

where $\hat{\theta}_{k,l}$ is the estimation of θ_k at the l -th trial. $M = 100$ is the Monte Carlo number. Note that in the simulation, the comparison of the computational complexity of each heuristic algorithm for SML estimation is based on the condition that all the algorithms converge to the same value (i.e., the RMSE for each algorithm is the same for a certain SNR); otherwise, it is meaningless to compare the computational complexity.

4.1. Validity of the Improvements of the Initialization Space. The scenario is that $m = 4$, $n = 2$, $\text{SNR} = 0$ dB, and $N = 100$. The true DOA is -15 degree and 20 degree. The array is a uniform linear array.

Figure 2 shows the position of the true DOA, the solution of SML, and the results of method one (“solution of ESPRIT”) and method two (“rough search DOA”) for both noncoherent (Figure 2(a)) and coherent (Figure 2(b)) cases. It is clear that both method one and method two provide

rather good initial position because they are close enough to the solution of SML.

Figure 3(a) shows the “scale” function $T(\text{SNR})$ with the parameter SNR. It is an empirical function obtained through many experiments, and it is only related to SNR. It is smaller for the noncoherent case or when SNR is higher. That is because the result of method one or method two, $\hat{\Theta}_{ES}$, is more close to the solution of SML when all the signals are independent (as Figure 2(a) shows) or SNR is higher. Then, the initialization space is constructed. Since this initialization space is not large and is close to the solution of SML, less particles are needed. As Figure 3(b) shows, only 5 particles are initiated and they are close enough to the solution of SML.

In Table 2, improved PSO and improved GA represent the PSO and GA algorithms using the improvement of the initialization space. The scenario is the same as Figure 2(a). From Table 2, it can be seen that after using the improvement of the initialization space, both the number of initial particles and the average iteration times are greatly reduced. That is because the initial particles are already close enough to the solution of SML, so the convergence process is quick. As a result, the computational complexity (calculating time) can be reduced greatly.

Similar simulation results can be observed for other scenarios. The validity of the improvement of the initialization space is proved. Furthermore, it is a general technique which can be used for other population-based heuristic algorithms.

4.2. Validity of the Improvements of Evolutionary Strategy.

In Tables 3 and 4, “proposed-F” represents the algorithm proposed in Section 3.2 whose evolutionary strategy is the fusion of GA and PSO. In Table 3, the scenario is the same as Figure 2(a), while in Table 4, the scenarios are the same as Figure 2(b).

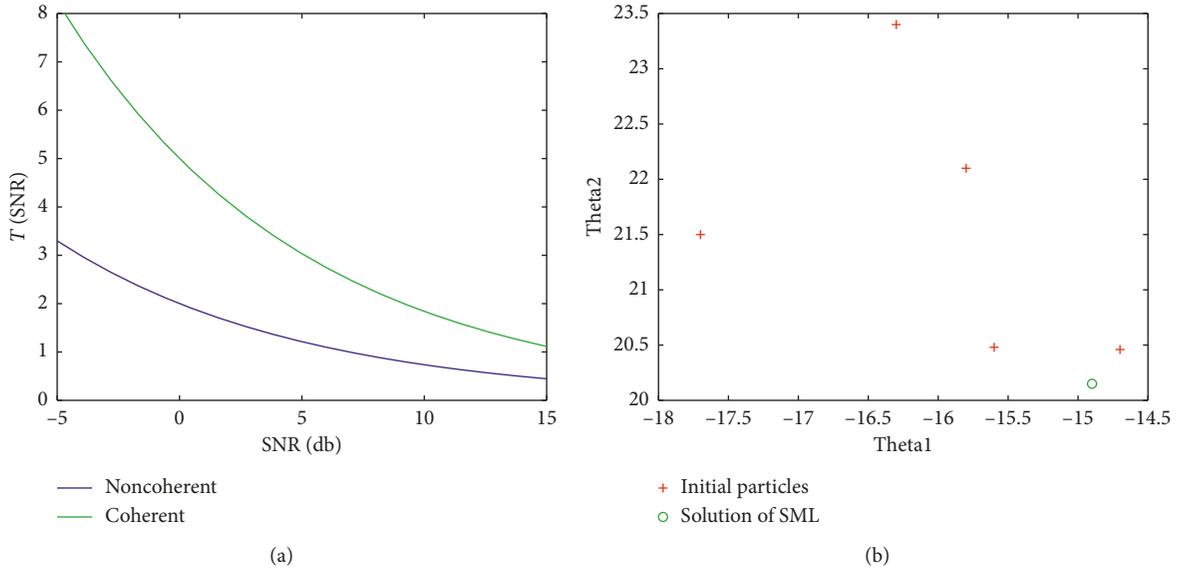


FIGURE 3: Improvement of the initialization space. (a) The “scale” function. (b) Positions of initial particles and the solution of SML (the scenario is the same as Figure 2(a)).

TABLE 2: Comparison of computational complexity of original GA, PSO, and the improved GA and PSO for SML estimation.

	Original PSO	Original GA	Improved PSO	Improved GA
Population size	30	30	5	5
Average iteration times	128	148	22	25
Total calculating time (seconds)	0.45	0.51	0.031	0.039

TABLE 3: Comparison of computational complexity of original GA, PSO, and the proposed fusion technique for SML estimation for the noncoherent case.

	Original PSO	Original GA	Proposed-F
Population size	30	30	30
Average iteration times	128	148	84
Total calculating time (seconds)	0.45	0.51	0.32

TABLE 4: Comparison of computational complexity of original GA, PSO, and the proposed fusion technique for SML estimation for the coherent case.

	Original PSO	Original GA	Proposed-F
Population size	30	30	30
Average iteration times	132	156	86
Total calculating time (seconds)	0.46	0.53	0.33

The “proposed-F” algorithm combines the characteristics of cross-mutation of the genetic algorithm, and its particle is also affected by the local best position found by itself and the global best position found by the swarm. Therefore, this algorithm inherits the advantages of both GA and PSO. From Tables 3 and 4, it can be seen that the number

of average iteration times is much less (about two-thirds) than that of the original GA and PSO for the same initial particles. As a result, the computational complexity is reduced. Similar simulation results can be observed for other scenarios.

4.3. Validity of Parallel Computing Technology. In the simulation, we use SPMD provided in Matlab for parallel computing with a four-core computer. Therefore, in SPMD, the number of workers is four. In Table 5, the scenario is the same as Figure 2(a). “P-PSO” and “P-GA” represent the PSO and the GA algorithm using parallel computing, respectively. From Table 5, it can be seen that for a 4-core parallel computing, the calculating time is about a half of the original one. Obviously parallel computing is effective.

Note that the three improvements aforementioned are independent to each other. At last, we do simulation using all the three improvements at the same time. In Table 6, the scenario is the same as Figure 2(a). The “Proposed” represents the algorithm using all the three improvements. It is clear that the calculating time is greatly reduced which is about one-thirty of the original PSO algorithm. The computational complexity of the proposed algorithm is even comparable to that of MUSIC. Therefore, the proposed improvements of heuristic algorithms have great practical value in real systems.

TABLE 5: Comparison of calculating time of original GA, PSO, and their application of parallel computing.

	Original PSO	Original GA	P-PSO	P-GA
Population size	30	30	30	30
Average iteration times	128	148	128	148
Total calculating time (seconds)	0.45	0.51	0.20	0.24

TABLE 6: Comparison of calculating time of original GA, PSO, and ESPRIT and the proposed algorithm.

	Original PSO	Original GA	Proposed	MUSIC	ESPRIT
Population size	30	30	5	–	–
Average iteration times	128	148	16	–	–
Total calculating time (seconds)	0.45	0.51	0.014	0.011	0.0017

5. Conclusion

This paper proposes three general improvements of heuristic algorithms, i.e., the optimization of the initialization space, the improvement of evolutionary strategies, and the usage of parallel computing techniques and took them for SML estimation of DOA as examples. The first improvement can greatly reduce the number of particles and iterative times. The second improvement fuses the evolutionary strategies of different algorithms so that the convergence process would accelerate. The third improvement is not a theoretical innovation but it is useful in real calculation. The contribution of this paper is that these improvements of heuristic algorithms are general and independent. They can be applied to different population-based heuristic algorithms for different nonlinear multi-dimensional optimization problems. Simulation results show that these improvements are of great practical value in real systems.

Data Availability

The Figures and Tables used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported financially by the National Natural Science Funds, China (61601519); Fundamental Research Funds for the Central University, China (18CX02135A); and Electronic Testing Technology Key Laboratory Fund Project, China (614200105010217).

References

- [1] F. Jin, T. Qiu, S. Luan, and W. Cui, "Joint estimation of the DOA and the number of sources for wideband signals using cyclic correntropy," *IEEE Access*, vol. 7, pp. 42482–42494, 2019.
- [2] H. Bai, M. F. Duarte, and R. Janaswamy, "Direction of arrival estimation for complex sources through l1 norm sparse bayesian learning," *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 765–769, 2019.
- [3] Y. Wang, X. Yang, J. Xie, L. Wang, and B. W.-H. Ng, "Sparsity-inducing DOA estimation of coherent signals under the coexistence of mutual coupling and nonuniform noise," *IEEE Access*, vol. 7, pp. 40271–40278, 2019.
- [4] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.
- [5] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.
- [6] B. D. Rao and K. V. S. Hari, "Performance analysis of root-music," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 1939–1949, 1989.
- [7] I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 10, pp. 1553–1560, 1988.
- [8] M. Wax, "Detection and localization of multiple sources via the stochastic signals model," *IEEE Trans. Signal Processing*, vol. 39, no. 11, pp. 2450–2456, 1991.
- [9] H. Chen and M. Suzuki, "Exact formulation for stochastic ML estimation of DOA," *IEICE Transactions on fundamentals of electronics, Communications and Computer Sciences*, vol. E93-A, no. 11, pp. 2141–2152, 2010.
- [10] P. Stoica, P. Babu, and J. Li, "SPICE: a sparse covariance-based estimation method for array processing," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 629–638, 2011.
- [11] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.
- [12] M. Viberg, B. Ottersten, and T. Kailath, "Detection and estimation in sensor array using weighted subspace fitting," *IEEE Transactions on Signal Processing*, vol. 39, no. 11, pp. 2436–2449, 1991.
- [13] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1783–1795, 1990.
- [14] B. Ottersten, M. Viberg, and T. Kailath, "Analysis of subspace fitting and ML techniques for parameter estimation from sensor array data," *IEEE Transactions on Signal Processing*, vol. 40, no. 3, pp. 590–600, 1992.
- [15] H. L. Van Trees, *Optimum Array Processing, Part IV of Detection, Estimation and Modulation Theory*, John Wiley & Sons, New York, NY, USA, 2002.
- [16] F. Gong, H. Chen, S. Li, J. Liu, Z. Gu, and M. Suzuki, "A low computational complexity SML estimation algorithm of DOA

- for wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–11, 2015.
- [17] K. Alexander, M. Tangemann, and B. Ernst, “A real-time DOA-based smart antenna processor,” *IEEE Transactions on Vehicular Technology*, vol. 51, no. 6, 2002.
- [18] T. B. Lavate, V. K. Kokate, and A. M. Sapkal, “Performance analysis of MUSIC and ESPRIT DOA estimation algorithms for adaptive array smart antenna in mobile communication,” in *Proceedings of the 2010 Second International Conference on Computer and Network Technology*, pp. 308–311, Bangkok, Thailand, April 2010.
- [19] Z.-M. Liu, C. Zhang, and P. S. Yu, “Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections,” *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7315–7327, 2018.
- [20] M.-H. Li and Y.-L. Lu, “Improving the performance of GAML DOA estimator with a resampling scheme,” *Signal Processing*, vol. 84, no. 10, pp. 1813–1822, 2004.
- [21] W. Shi, J. Huang, and Y. Hou, “Fast DOA estimation algorithm for MIMO sonar based on ant colony optimization,” *Journal of Systems Engineering and Electronics*, vol. 23, no. 2, pp. 173–178, 2012.
- [22] J. Wang, Y. Zhao, and Z. Wang, “Low complexity subspace fitting method for wideband signal location,” in *Proceedings of the 5th IFIP International Conference on Wireless and Optical Communications Networks*, pp. 1–4, Surabaya, Indonesia, May 2008.
- [23] L. Boccatto, R. Krummenauer, R. Attux, and A. Lopes, “Application of natural computing algorithms to maximum likelihood estimation of direction of arrival,” *Signal Processing*, vol. 92, no. 5, pp. 1338–1352, 2012.
- [24] R. T. Williams, S. Prasad, A. K. Mahalanabis, and L. H. Sibul, “An improved spatial smoothing technique for bearing estimation in a multipath environment,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 4, pp. 425–432, 1988.
- [25] G. Shi-Wei and B. Zheng, “Data-based matrix decomposition technique for high resolution array processing of coherent signals,” *Electronics Letters*, vol. 23, no. 12, pp. 643–645, 1987.



Hindawi

Submit your manuscripts at
www.hindawi.com

