sensors

Article

DOA Estimation for Local Scattered CDMA Signals by Particle Swarm Optimization

Jhih-Chung Chang

Department of Information Technology, Ling Tung University, Taichung 408, Taiwan;
E-Mail: changjc@mail.ltu.edu.tw; Tel.: +886-4-2389-2088; Fax: +886-4-3600-2535

Abstract: This paper deals with the direction-of-arrival (DOA) estimation of local scattered code-division multiple access (CDMA) signals based on a particle swarm optimization (PSO) search. For conventional spectral searching estimators with local scattering, the searching complexity and estimating accuracy strictly depend on the number of search grids used during the search. In order to obtain high-resolution and accurate DOA estimation, a smaller grid size is needed. This is time consuming and it is unclear how to determine the required number of search grids. In this paper, a modified PSO is presented to reduce the required search grids for the conventional spectral searching estimator with the effects of local scattering. Finally, several computer simulations are provided for illustration and comparison.

1. Introduction

Adaptive array techniques have been developed for enhancing the performance of code-division multiple access (CDMA) systems. In the CDMA system, each user employs a unique pseudo-noise (PN) codeword to identify their data stream. Each user's transmission interferes with all the other users and causes multiple access interference (MAI). The use of antenna arrays as a tool for improving coverage, reducing interference, and increasing capacity in CDMA systems has attracted significant interest [1]. Multiple propagation is common in cellular systems, which may result from local scatters in the vicinity of the sources. In macrocellular environments, high base station antennas will receive these locally scattered signals from the mobile terminal, which are coherent and confined to a narrow

angular region [2]. In addition, the observation period is assumed to be short in comparison to the coherence time of the channel so that the channel may be modeled as time-invariant. A mobile terminal with local scattering can be well approximated by a single point source, as seen from the base station. The model can express the mobile and coherent interference from the surrounding environment under non line-of-sight conditions as another arithmetical model. In cellular radio systems, antenna array processing is considered to be a useful method to solve problems such as multipath and co-channel interference and increase capacity [1]. There have been some studies appraising the impact of a local scattering channel model on CDMA system, which propose uniform linear array (ULA) geometry [3]. They use a first-order Taylor expansion to linearize the nonlinear conventional array manifold produced by scatters and develop the generalized array manifold (GAM) model which can obtain better nominal DOA estimation for mobiles. Here, the multiple signal classification (MUSIC) searching function derived by the steering vector of the GAM model is termed as GMUSIC. However, in all these studies, estimation of direction of arrival (DOA) of the desired signals is required. Array outputs aligned with code-matched filter can make the DOA estimation of multiple sources equivalent to that for single source localization problem in a noisy environment. With the advantage of code-matched filter inherent in the CDMA system, it has been proved that the multiple signal classification estimator [4] can obtain an unbiased DOA estimation with low mean-square-error (MSE) [5]. It also contributes to solving the limitation that the number of array elements must be more than the number of impinging sources, but for a conventional spectral searching DOA estimator such as MUSIC, its searching complexity and estimating accuracy strictly depend on the number of search grids used during the search. This is time consuming and it is unclear how to determine the required number of search grids. Thus, in this paper, we will focus on the GAM model of ULA to employ the DOA estimation in multipath reflection scenario from scatters.

Particle swarm optimization (PSO) is a population-based stochastic optimization paradigm, in which each agent, named particle, of the population, named swarm, is thought of as a collision-proof bird and used to represent a potential solution [6]. Like a genetic algorithm (GA), PSO starts by initializing a population of random solutions and searches for optima by updating generations. But, PSO does not use any evolution operators. In PSO, particles fly through the problem space by following their own experience and the best experience attained by the swarm as a whole. In contrast to analytical or general heuristic methods, PSO is computationally efficient and has great capability of escaping local optima [7,8]. In addition, a key characteristic of PSO is that the algorithm itself is highly robust yet remarkably simple to implement, while processing similar capabilities as other evolutionary algorithms such as GA [9]. A maximum likelihood (ML) criterion with hard-constraint PSO based solution applied to DOA estimation is proposed in [10], that explores the potentially superior performance at less computational costs. The same PSO algorithm based on the ML methodology is also derived in [11], where the cost function is an extension of the ML criteria that were originally developed for angle estimation with some of the sensors that are perfectly calibrated, while others are uncalibrated. Recently an adaptive PSO algorithm has been successfully applied in ML optimization solutions [12]. In another report a combined fuzzy adaptive PSO and differential evolution are also used to solve economic dispatch problems [13]. In this paper, a modified adaptive inertia weight PSO (APSO) is proposed to rationally balance the global exploration and local exploitation abilities of the particle to the GMUSIC criteria function [3] for CDMA signals. The

resulting estimator is called APSO-GMUSIC, where a simple and effective measure, individual search ability, is defined to indicate whether each particle lacks global exploration ability in each dimension. By employing APSO to treat our optimization problem of the desired signal direction angle θ. The DOA, obtained by the APSO-GMUSIC estimator, converges to the optimal or near optimal solution. Simulation results show that the proposed APSO-GMUSIC estimator is very suitable to treat the DOA estimation for CDMA signals in a local scattered scenario.

2. System Model

2.1. Array Data Model

Consider a DOA estimation scenario in a CDMA system with P users. The data bit $b_p(t) \in \{-1, 1\}$ of the pth user is spread by a pseudo-noise (PN) signature waveform $c_p(t)$ for $p = 1, 2, \ldots, P$. The signature waveform of the pth user is composed of a spreading sequence of L chips, *i.e.*, $c_p(t) = \sum_{l=1}^{L} g_{p,l} p_{T_c}[t - (l-1)T_c]$, where $g_{p,l} \in \{-1/\sqrt{L},\ +1/\sqrt{L}\}$ is the lth spreading chip for the pth user and $p_{Tc}(t)$ is the spreading pulse of duration T_c. Thus, the transmitted signal of the pth user during a bit interval T_b can be represented by:

$$s_p(t) = \varepsilon_p b_p(t) c_p(t) \tag{1}$$

where ε_p is the signal amplitude. Let the processing gain of the spreading code be $L = T_b/T_c$. To simplify analysis, the PN code during a bit interval at a sample rate $1/T_c$ can be expressed as an $L \times 1$ vector of discrete sequence $\mathbf{c}_p = [c_{p1}, c_{p2}, \ldots, c_{pL}]^T$ where the superscript T is the transpose operator.

Employing the code-matched filter and DOA estimation algorithm, we propose a receiving base station of CDMA communication system with a uniform linear array (ULA). The antenna array consists of M identical omidirectional elements and receives line-of-sight signals from P users that arrive at the array from different bearing angles $\theta_1, \theta_2, \ldots, \theta_p$ with respect to the broadside of array in a cell/sector. Assume that the array element space is d. The direction vector associated with the pth user is given by:

$$\mathbf{a}(\theta_p) = [a_1(\theta_p), a_2(\theta_p), \ldots, a_M(\theta_p)]^T \tag{2}$$

where $a_m(\theta_p) = \exp[-j\pi d(m-1)\sin(\theta_p)/\beta]$ denotes the response of the mth sensor array to a signal with unit amplitude arriving from the direction angle θ_p, where β is the propagation speed of plane wave. Thus, the received baseband signal across the array can be written in vector form:

$$\mathbf{x}(t) = \sum_{p=1}^{P} s_p(t)\mathbf{a}(\theta_p) + \mathbf{n}(t) \tag{3}$$

where $s_p(t) = \sum_{i=-\infty}^{\infty} \varepsilon_p b_p(i) c_p(t - iT_b - \tau_p)$, τ_p is the time delay, and the observation noise vector $\mathbf{n}(t)$ represents the spatially and temporally zero mean white Gaussian noise vector.

2.2. MUSIC Algorithms for Local Scattering Channels

To show how the spatial signatures depend on the spatial distribution of the multipath propagation, we assume that the time dispersion produced by the multipath propagation is small compared to the reciprocal of the signal's bandwidth, the time delay may be approximated as a phase shift $s_p(t - \tau_{ph}) \cong e^{-j2\pi f_c \tau_{ph}} s_p(t)$, where f_c is the carrier frequency, and τ_{ph} is the time delay associated with the hth scattered signal from the pth source arriving at the array. Let $s_p(t)$ be the signal scattered by the pth user. Due to the multipath propagation near the pth user, its contribution to the array is modeled as a superposition of N_p scattered rays. Then, the data received at the array is given by $M \times 1$ vector form:

$$\mathbf{x}(t) = \sum_{p=1}^{P} \sum_{h=1}^{N_k} \beta_{ph} s_p(t) e^{-j\,2\pi f_c \tau_{ph}} \mathbf{a}_{ph}(\theta_p + \tilde{\Delta}_{ph}) + \mathbf{n}(t) \tag{4}$$

where the reflection coefficient β_{ph} is the complex amplitude of the hth scattered signal from the pth user, $\mathbf{a}_{ph}$ is the response vector of the hth scattered signal from the pth user, and $j = \sqrt{-1}$. $\tilde{\Delta}_{ph}$ is spread angle of the pth user due to local scattering. It is assumed that the power is normalized so that all rays have equal power $1/N_p$. Then $\beta_{ph} = e^{j\chi_{ph}} / \sqrt{N_p}$, where χ_{ph} is i.i.d. and uniform over $[0, 2\pi]$ [3]. Then, the received data vector $\mathbf{x}(t)$ can be expressed as:

$$\mathbf{x}(t) = \mathbf{U}\mathbf{s}(t) + \mathbf{n}(t) \tag{5}$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p]$ and $\mathbf{s}(t) = [s_1(t), s_2(t), \ldots, s_p(t)]^T$. The pth column of $\mathbf{U}$, which is denoted as $\mathbf{u}_p$, represents the spatial signature associated with the signal $s_p(t)$, where $\mathbf{u}_p = \sum_{h=1}^{N_p} \gamma_{ph} \mathbf{a}_{ph}$ and $\gamma_{ph} = \beta_{ph} e^{-j2\pi f_c \tau_{ph}}$. In order to pick out the pth user's signal, a code-matched filter containing the specified PN code vector $\mathbf{c}_p$ is applied to $\mathbf{x}(t)$. For simplicity, the bit sampling index is dropped to make the equations more readable in the following description. Therefore, the pth user's despread and sampled array vector signal $\mathbf{y}_p$ can be represented as:

$$\mathbf{y}_p = \mathbf{x}\mathbf{c}_p = Ls_p \mathbf{u}(\theta_p) + \sum_{q=1,\, q \neq p}^{P} s_q \mathbf{u}(\theta_q) + \mathbf{n}_p \tag{6}$$

where $s_p = \varepsilon_p b_p$ and $s_q = \varepsilon_p b_p \kappa_{qp}$. Hence, $\pi_p = E\{|s_p|^2\}$ and $\pi_q = E\{|s_q|^2\}$ denote the average power of the desired pth user and each interferer user, respectively. The notation $E\{\bullet\}$ is used to denote the expectation operator. κ_{qp} is the inner product of two different PN code vectors $\mathbf{c}_q^T \mathbf{c}_p$, and the $M \times 1$ vector $\mathbf{n}_p$ with zero mean and variance π_n is the code-matched filter output of the noise vector $\mathbf{n}$. It was shown in Appendix A of [3] that κ_{qp} approaches a real Gaussian random variable with zero mean and unit variance for the normalized PN codes. Consequently, the signals of the undesired users through the code-matched filter, *i.e.*, MAI, can be viewed as a noise vector $\mathbf{n}_{MAIp}$ with zero mean and variance π_{MAIp} and $\mathbf{n}_{MAI_p} = \sum_{q=1,\, q \neq p}^{P} s_q \mathbf{u}(\theta_q)$. The correlation matrix of Equation (6) in the observation interval is given by:

$$\mathbf{R}_p = E\{\mathbf{y}_p \mathbf{y}_p^H\} = L^2 \pi_p \mathbf{u}(\theta_p) \mathbf{u}^H(\theta_p) + \pi_{np} \mathbf{I}_M \tag{7}$$

where the subscript H denotes conjugate transpose, $\pi_{np} = \pi_{MAIp} + \pi_n$, and $\mathbf{I}_M$ is an $M \times M$ identity matrix.

For high angular resolution subspace methods, the MUSIC algorithm is a kind of DOA estimation technique based on eigenvalue decomposition (EVD), which is also called the noise subspace-based method. The eigendecomposition of Equation (7) can be expressed as:

$$\mathbf{R}_p = \sum_{m=1}^{M} \eta_m \mathbf{e}_{pm} \mathbf{e}_{pm}^H = \eta_{p1} \mathbf{e}_{p1} \mathbf{e}_{p1}^H + \mathbf{E}_{pn} \mathbf{\Lambda}_{pn} \mathbf{E}_{pn}^H, \tag{8}$$

where $\eta_{p1} \geq \eta_{p2} = \eta_{p3} = \cdots = \eta_{pM} = \pi_{np}$ are the eigenvalues of $\mathbf{R}_p$ and $\mathbf{e}_{pm}$ denotes the eigenvector associated with η_{pm} for $m = 1, 2, ..., M$. Moreover, $\mathbf{e}_{p1}$ and $\mathbf{E}_{pn} = [\mathbf{e}_{p2}, \mathbf{e}_{p3}, \cdots, \mathbf{e}_{pM}]$ are orthogonal and span the signal and noise subspace corresponding to $\mathbf{R}_p$, respectively. $\mathbf{\Lambda}_{pn} = \pi_{np} \mathbf{I}_{M-1}$ is the noise eigenvalue matrix. Furthermore, $\mathbf{e}_{p1}$ spans the same signal subspace as that spanned by $\mathbf{a}(\theta_p)$. Thus, we have $\mathbf{E}_{pn}^H \mathbf{a}(\theta_p) = \mathbf{0}$ and $\mathbf{a}^H(\theta_p) \mathbf{E}_{pn} = \mathbf{0}$. The MUSIC estimates the DOA of the pth user from the highest peak of the following spectrum [3]:

$$S_p(\theta) = \max_{\theta} \frac{1}{|\mathbf{a}^H(\theta) \mathbf{E}_{pn} \mathbf{E}_{pn}^H \mathbf{a}(\theta)|}, \quad p = 1, 2, \cdots, P \tag{9}$$

where the largest maxima of $S_p(\theta)$ is taken to be the estimate of the DOA of the pth user.

As the angular spread $\tilde{\Delta}_{ph}$ of the hth scattered ray from the pth source arriving at the array is relatively small, a first-order Taylor expansion of $\mathbf{u}_p = \sum_{h=1}^{N_p} \gamma_{ph} \mathbf{a}_{ph}$ in Equation (4) may be approximated as follows [14]:

$$\tilde{\mathbf{u}}_p \cong \mathbf{a}_p + \rho_p \mathbf{d}_p \tag{10}$$

where $\rho_p = \sum_{h=1}^{N_p} \gamma_{ph} \tilde{\Delta}_{ph} / \sum_{h=1}^{N_p} \gamma_{ph}$ and $\mathbf{d}_p = \partial \mathbf{a}_p / \partial \theta_p$ is the first derivative of the steering vector. Equation (10) is a linear model and regards as the generalized array manifold (GAM) model with scattering, which is the superposition of the pth source steering vectors and their derivatives. Then, the received data vector $\tilde{\mathbf{x}}(t)$ can be expressed as:

$$\tilde{\mathbf{x}}(t) = \tilde{\mathbf{U}} \mathbf{s}(t) + \mathbf{n}(t) \tag{11}$$

Therefore, the pth user's despread and sampled array vector signal $\tilde{\mathbf{y}}_p$ can be represented as:

$$\tilde{\mathbf{y}}_p = \tilde{\mathbf{x}} \mathbf{c}_p = L s_p \tilde{\mathbf{u}}(\theta_p) + \sum_{q=1, q \neq p}^{P} s_q \tilde{\mathbf{u}}(\theta_q) + \mathbf{n}_p \tag{12}$$

The correlation matrix of Equation (12) in the observation interval is given by:

$$\tilde{\mathbf{R}}_p = E\{\tilde{\mathbf{y}}_p \tilde{\mathbf{y}}_p^H\} = L^2 \pi_p \tilde{\mathbf{u}}(\theta_p) \tilde{\mathbf{u}}^H(\theta_p) + \pi_{np} \mathbf{I}_M \tag{13}$$

The eigendecomposition of Equation (13) can be expressed as:

$$\tilde{\mathbf{R}}_p = \sum_{m=1}^{M} \eta_m \tilde{\mathbf{e}}_{pm} \tilde{\mathbf{e}}_{pm}^H = \eta_{p1} \tilde{\mathbf{e}}_{p1} \tilde{\mathbf{e}}_{p1}^H + \tilde{\mathbf{E}}_{pn} \mathbf{\Lambda}_{pn} \tilde{\mathbf{E}}_{pn}^H, \tag{14}$$

With the GAM model, the MUSIC searching function of Equation (9) can be expanded into a MUSIC with local scattering. The MUSIC searching function derived by the steering vector of the GAM model in Equation (10) , which is termed as GMUSIC and can be expressed as:

$$GS_p(\theta, \rho) = \max_{\theta, \rho} \frac{1}{|\xi^H \overline{\mathbf{A}}^H(\theta) \tilde{\mathbf{E}}_{pn} \tilde{\mathbf{E}}_{pn}^H \overline{\mathbf{A}}(\theta)\xi|}, \quad p = 1, 2, \cdots, P \tag{15}$$

where $\overline{\mathbf{A}}(\theta) = [\mathbf{a}(\theta) \quad \mathbf{d}(\theta)]$ and $\xi = [1 \quad \rho]^T$. The parameter of ρ can be estimated by using the method of [3]. And, the $\tilde{\mathbf{E}}_{pn}$ in Equation (15) is the noise subspace, which is the eigendecomposition of the autocorrelation matrix derived by the linearized steering vector in Equation (10). We can estimate DOA by searching the array manifold for direction vector and the largest peak of the function $GSp(\theta, \rho)$ denotes the DOA of the *p*th user.

2.3. Problem Formulation

Recently, a much more computationally efficient and more accurate parameter estimating approach via polynomial rooting method has been proposed to improve the spectral searching approaches for reducing the computational load. Due to the fact the scanning vector $\overline{\mathbf{A}}(\theta)$ of the GMUSIC is not Vandermonde structure, $GSp(\theta, \rho)$ cannot be implemented by using the polynomial rooting approach. The computational complexity of the classical subspace approach of MUSIC estimator is about $12M^3$ complex multiplications (CM) for computing EVD of a $M \times M$ dimension matrix. Let the search number be B. Therefore, the total required CM for computing Equation (15) of the GMUSIC estimator is about $12M^3 + BM^2$. The performance of the abovementioned spectral searching approach is governed by the scanning grid size and the number of search grids while implementing the high-resolution DOA estimation. A major problem of GMUSIC type algorithms is the heavy computational load incurred with spatial spectral search when a smaller grid size is adapted. However, it is time consuming and the optimum search grid size is not clear. Smaller grid size can improve estimate accuracy, but the required computational load also becomes relatively larger (e.g., if grid sizes are set to 1°, 0.1° and 0.01°, then the required search numbers B = 181, 1,801, and 18,001, respectively.). With the number of particle population P_N and the maximum number of iterations k_{max}, the computational complexity of our APSO approach is about $P_N \times k_{max}$. Hence, the computational complexity of APSO is smaller than the GMUSIC. Therefore, in order to reduce computational load, this paper investigates the feasibility of applying the PSO to replace the spectrum searching approach. As a result, the proposed PSO-based searching GMUSIC estimator does not increase the computational complexity, but significantly reduces the searching process requirement compared with the spectrum searching GMUSIC.

3. The Proposed PSO Based Searching Algorithms

Due to the fact that the performance of the abovementioned spectral searching GMUSIC estimator is governed by the scanning grid size and the number of search grids while implementing the high-resolution DOA estimation, it is time consuming and the search grid is not clear. In this section, we present PSO based searching approaches by maximizing the fitness function of GMUSIC at each iteration.

3.1. HPSO-GMUSIC Estimator

In order to reduce the scanning accurate angle problems of high computation cost, we use the PSO to replace scan in the ULA. For time-space signal processing, each particle can be treated as a point in the P-dimensional problem space and a swarm consisting of D random particles and then searches for best position (solution) by updating generations until exceeding the limit of iteration number. The position of the ith particle is represented as $\boldsymbol{\theta}_i = [\theta_{i1}, \theta_{i2}, ..., \theta_{iP}]$, the rate of the position change (velocity) for ith particle is represented as $\mathbf{v}_i = [v_{i1}, v_{i2}, ..., v_{iP}]$. The best previous position of the ith particle, which gives the best fitness value, is recorded as $\mathbf{p}_i = [p_{i1}, p_{i2}, ..., p_{iP}]$. And, the index of the best particle among all the particles in the population is represented by $\mathbf{p}_g = [p_{g1}, p_{g2}, ..., p_{gp}]$ and called global best location. In every iteration, the local bests and global bests are determined through evaluating the fitness values of the current population of particles. Two factors characterize a particle status on the search space: its position and velocity. The P-dimensional position for the ith particle in the kth iteration can be denoted as $\boldsymbol{\theta}_i(k) = [\boldsymbol{\theta}_{i1}(k), \boldsymbol{\theta}_{i2}(k), ..., \boldsymbol{\theta}_{iP}(k)]$. Similarly, the velocity for the ith particle in the kth generation can be described as $\mathbf{v}_i(k) = [v_{i1}(k), v_{i2}(k) ..., v_{iP}(k)]$. The velocity and the position of the ith particle at the $(k+1)$th iteration for $i = 1, 2, ..., D$ and $k = 1, 2, ..., k_{\max}$ are updated according to the following equations:

$$\mathbf{v}_i(k+1) = w(k)\mathbf{v}_i(k) + c_1\mathbf{r}_1(k) \otimes (\mathbf{p}_i(k) - \boldsymbol{\theta}_i(k)) + c_2\mathbf{r}_2(k) \otimes (\mathbf{p}_g(k) - \boldsymbol{\theta}_i(k)) \tag{16}$$

$$\boldsymbol{\theta}_i(k+1) = \boldsymbol{\theta}_i(k) + \mathbf{v}_i(k+1) \tag{17}$$

where $\otimes$ denotes element-wise product and the positive acceleration constants c_1 and c_2 are two positive constants named learning factors. $\mathbf{r}_1(k)$ and $\mathbf{r}_1(k)$ are P-dimensional vectors consisting of independent random numbers uniformly distributed between 0 and 1, which are used to stochastically vary the relative pull of $\mathbf{p}_i$ and $\mathbf{p}_g$ in order to simulate the unpredictable component of natural swarm behavior. The inertia weight $w(k)$ based on the linear decreasing strategy is considered critically for the convergence behavior of PSO [10], which is selected to decrease during the optimization process. Thus, PSO tends to have more global search ability at the beginning of the run while having more local search ability near the end of the optimization. Given a maximum value $w_{\max}$ and a minimum value $w_{\min}$, $w(k)$ is updated as:

$$w(k+1) = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}}k \tag{18}$$

where $k_{\max}$ is the maximum number of iterations, $w_{\max}$ and $w_{\min}$ are chosen as 0.9 and 0.4 respectively in this work as in [10]. Because the PSO particle represents a series of priorities that range from $-90°$ to $90°$, all parameters of the P-dimensional particle positions, either initialized or updated during search, must be confined within $[\theta_{\min}, \theta_{\max}] = [-90°, 90°]$, avoiding infeasible particle positions that can lead to slow PSO searches. The new particle position is calculated using Equation (17).

A particle position vector is converted to a candidate solution vector in the problem space through a suitable mapping. The score of the mapped vector evaluated by a GMUSIC function is regarded as the fitness of the corresponding particle. For time-space signal processing, the DOA GMUSIC estimation

problem in this paper is to find the *i*th particle position or estimate angle $\theta_{ip}(k)$ of the *p*th user to maximize the following fitness function:

$$\text{fitness}_{ip}(k) = \frac{1}{\xi^H \overline{\mathbf{A}}^H(\theta_{ip}(k))\tilde{\mathbf{E}}_{pn}\tilde{\mathbf{E}}_{pn}^H \overline{\mathbf{A}}(\theta_{ip}(k))\xi}, \quad p = 1, 2, \cdots, P \tag{19}$$

At the end of iteration, global best location $p_{gp}(k_{\max})$ is the final estimated angle defined as $\hat{\theta}_p = p_{gp}(k_{\max})$. The final global best position $\mathbf{p}_g(k_{\max})$ is taken as the GMUSIC estimates of users. As a result, the PSO with linear decreasing inertia weight, particle velocity limitation, and particle position clipping is termed as hard-constraint PSO GMUSIC (HPSO-GMUSIC). Unfortunately, the HPSO-GMUSIC is not so facile to implement for its overhead on computing the search space mapping, particle velocity limitation and particle position clipping. In this paper, we present a more feasible and efficient modified PSO algorithm for the HPSO-GMUSIC estimator.

3.2. The Proposed Multiple Adaptive Inertia Weight

The inertia weight is critical for the performance of PSO, which balances global exploration and local exploitation abilities of the swarm. In other words, the inertia weight $w(k)$ is employed to control the impact of the previous history of velocities on the current velocity, thereby influencing the trade-off between global and local exploration abilities of the "flying points". A large inertia weight tends to facilitate searching new area and global exploration. Conversely, a small inertia weight facilitates local exploitation in the current search area. Suitable selection of the inertia weight provides a balance between global and local exploration abilities and thus requires less iteration on average to find the optimum. During the search every particle dynamically changes its position, so every particle locates in a complex environment and faces different situation. Therefore, each particle along every dimension may have different trade off between global and local search abilities. According to [15], it has been shown that the performance of the PSO algorithm with linearly decreasing inertia weight has the ability to quickly converge, but the PSO may lack global search ability at the end of a run and may fail to find the required optima in cases when the problem to be solved is too complicated and complex. But to some extent, this can be overcome by employing a self-adapting strategy for adjusting the inertia weight. In this subsection, inertia weight is dynamically adapted for every particle along every dimension. A measure, individual search ability, which characterizes the faced situation for every particle is defined. Basing on this measure, the particle could decide to whether to increase or decrease the values of inertia weight by means of the transfer function. The fine strategy of dynamically adjusting inertia weight could lead to improvement in performance of PSO.

For velocity updating, the last two parts of Equation (16) can be viewed as the accelerations parts and can be defined as $a_{ip}(k) = c_1 \cdot rand(\bullet) \cdot (p_{ip}(k) - \theta_{ip}(k))$ and $a_{gp}(k) = c_2 \cdot rand(\bullet) \cdot (p_{ip}(k) - \theta_{ip}(k))$, where $rand(\bullet)$ is independent random number with uniformly distributed between [0,1]. So we could consider that the particle is moving with velocity of $v_{ip}(k)$ and acceleration of $a_{ip}(k)$ and $a_{gp}(k)$. But, $a_{gp}(k)$ is the dominant term for improving convergence rate. Suppose that the mass of the *i*th particle in the *p*th dimension is normalized to 1 kg. According to the principle of mechanics, $a_{gp}(k) = f_{ip}(k)$, where $f_{ip}(k)$ is an outside force, which is put on the particle comes from the "pulling" of $p_{ip}(k)$ and $p_{gp}(k)$. For ULA, the DOA is a one-dimensional searching problem. In order to make the particle fly towards optimal

region quickly, $v_{ip}(k)$ should turn to the direction of $f_{ip}(k)$ as soon as possible. Define the error between $p_{ip}(k)$ and $\theta_{ip}(k)$ is $z_{ip}(k) = p_{gp}(k) - \theta_{ip}(k)$. Let $m_{ip}(k) = |z_{ip}(k)|/\theta_{\max}$ which is a number between 0 and 1. The velocity update problem of the *i*th particle on the *p*th dimension can be divided into two classes:

(1) Firstly, $v_{ip}(k)$ and $z_{ip}(k)$ are in the same direction, $z_{ip}(k)/v_{ip}(k) \geq 0$. If $m_{ip}(k)$ is relatively large, it means the particle is in the right direction, but the velocity is too small. Therefore, the particle needs to speed up, and the inertia weight $w_{ip}(k)$ needs to be set larger. If $m_{ip}(k)$ is relatively small, it means the particle has come to the location, that is near the optimal region. So the velocity of this particle should slow down and the neighborhood of the state should be searched carefully, and $w_{ip}(k)$ should be set smaller.

(2) On the other hand, consider $v_{ip}(k)$ and $z_{ip}(k)$ on different direction, $z_{ip}(k)/v_{ip}(k) < 0$. If $m_{ip}(k)$ is relatively large, it means that the particle's state is far from the optimal region. So the particle needs to change its velocity as soon as possible, and the inertia weight $w_{ip}(k)$ needs to be set smaller. If $m_{ip}(k)$ is relatively small, it means that it's not urgent for the particle to change its direction on this dimension, and $w_{ip}(k)$ could be set a large value.

Based on the aforementioned analysis, an adaptive inertia weight strategy is proposed and is shown in Table 1. The individual normalized search ability of the *i*th particle along the *p*th dimension is defined as $m_{ip}(k)$. It is noted that the value of $w_{ip}(k)$ is a function of $m_{ip}(k)$. To get a balance of global search and local search ability, $w_{ip}(k)$ cannot be too large or too small, thus $w_{ip}(k)$ is limited in the range of $[w_{\min}, w_{\max}]$, which is like the process of normalization. Therefore, we used μ-*law* algorithm to achieve our strategy for every particle along every dimension, normalization of the $m_{ip}(k)$ to $\overline{w}_{ip}(k)$. The μ-*law* algorithm is a companding scheme used in telephone network [16]. Increasing the value of *u*, the dynamic range capability of μ-*law* can be improved and defined by:

$$\overline{w}_{ip}(k) = \frac{\log(1 + \mu m_{ip}(k))}{\log(1 + \mu)}, \quad 0 \leq m_{ip}(k) \leq 1 \tag{20}$$

where $m_{ip}(k)$ and $\overline{w}_{ip}(k)$ are normalized input and output values respectively, and the positive constant μ is the compression parameter. The reciprocal slop of this curve is given by the derivative of $m_{ip}(k)$ with respect to $\overline{w}_{ip}(k)$. For case 1, the value for small $m_{ip}(k)$ increases at the expense of the value for large $m_{ip}(k)$. To accommodate these conflicting requirements (*i.e.*, a reasonable values for both small and large $m_{ip}(k)$), a compromise is usually made in choosing the value of parameter μ for μ-*law*. The requirement of case 2 is opposite. Finally, based on our strategy and Equation (20), we can define multiple adaptive inertia weight $w_{ip}(k)$ as:

$$w_{ip}(k) = \begin{cases} w_{\min} + \overline{w}_{ip}(k), & \dfrac{z_{ip}(k)}{v_{ip}(k)} \geq 0 \\ w_{\min} + (1 - \overline{w}_{ip}(k)), & \dfrac{z_{ip}(k)}{v_{ip}(k)} < 0 \end{cases} \tag{21}$$

In Equation (21), $w_{\min}$ be added to avoid particles from stopping moving. The curves of $w_{ip}(k)$ with $w_{\min} = 0$ using different μ can be plotted in Figure 1. Then, we also investigate the sensitivity for APSO-GMUSIC with different values of μ However, it accords with our strategy for different μ. Thus, μ-law algorithm with $\mu = 100$ is chosen. Note that for every particle in population, $w_{ip}(k)$ is unique and

can be computed individually. Therefore, the single inertia weight $w_{ip}(k)$ can be replaced by a multiple adaptive inertia weight $w_{ip}(k)$. The proposed APSO-GMUSIC seems to be robust to control parameters due to the intrinsic advantages of the algorithm and the separation of the problem-independent PSO kernel from newly introduced problem-specific features in our design for adaptive multiple inertia weight. Finally, the steps for implementing the APSO-GMUSIC are shown in Figure 2 and described in the list that follows.

Table 1. Adaptive inertia weight strategy.

Value of $w_{ip}(k)$		Value of $m_{ip}(k)$		
		Small	Middle	Large
Directions of $v_{ip}(k)$ and $z_{ip}(k)$	Same	small	middle	large
	Opposite	large	middle	small

Figure 1. Normalized output $\overline{w}_{ip}(k)$ *versus* normalized input $z_{ip}(k)/\theta_{\max}$ under $w_{\min}=0$.
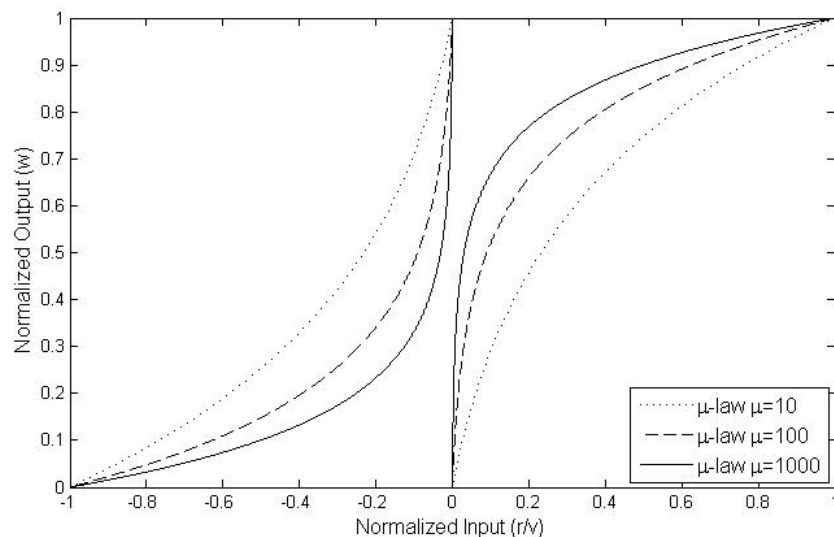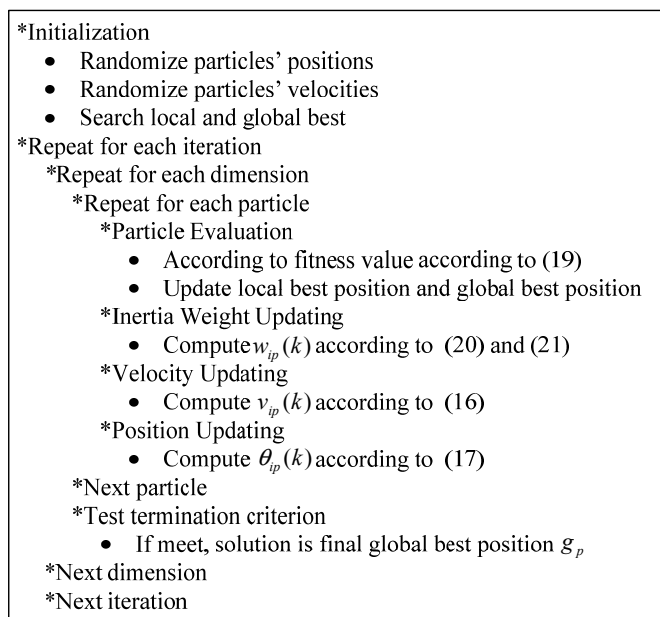

Figure 2. Flowchart illustrating the main steps of the APSO-GMUSIC.

```
*Initialization
    • Randomize particles' positions
    • Randomize particles' velocities
    • Search local and global best
*Repeat for each iteration
    *Repeat for each dimension
        *Repeat for each particle
            *Particle Evaluation
                • According to fitness value according to (19)
                • Update local best position and global best position
            *Inertia Weight Updating
                • Compute wip(k) according to (20) and (21)
            *Velocity Updating
                • Compute vip(k) according to (16)
            *Position Updating
                • Compute θip(k) according to (17)
        *Next particle
        *Test termination criterion
            • If meet, solution is final global best position gp
    *Next dimension
*Next iteration
```

4. Computer Simulations

Several computer simulations for illustration and comparison are presented in this section. We use a 6-element ULA with half wavelength for simulation. Consider an asynchronous CDMA system with PN codes of length 31 and BPSK modulation. The spatial signature $\mathbf{u}_p$ is generated by $N_p = 30$ independent identically distributed local scatters with uniform random angular spread of width $2\tilde{\Delta}_p$. It is assumed that $\tilde{\Delta}_p$ is uniformly distributed over the interval $[-\Delta p, \Delta_p]$. Each scatter has equal power and is randomly, uniformly distributed over $[0, 2\pi]$. The value of parameter ρ is assumed to be known. Eight users are impinging on the array with random impinging angles of uniform distribution in the interval $[-90°, 90°]$. All signal powers are set to be 20 dBW. The additive background noise is assumed to be white Gaussian distribution with zero-mean and unit Watts of power. For GMUSIC, the search resolution (grid) is set to 1°, 0.1°, and 0.01° under the search range of $[-90°, 90°]$. The root-mean-squared error (RMSE) is calculated in an average manner as RMSE $= [(1/FP) \sum_{j=1}^{F} \sum_{p=1}^{P} (\hat{\theta}_{jp} - \theta_p)^2]^{0.5}$, where F indicates the number of independent simulation runs, P is the number of users, $\hat{\theta}_{jp}$ is the pth estimate DOA achieved in the jth run, and θ_p is the true DOA of the pth user. The PSO parameters chosen for all the experiments are summarized in Table 2. All PSO algorithms start with random initializations and are terminated if the maximum iteration $k_{\max}$ is reached or the global best particle position is not updated in 20 successive iterations. Every simulation result is presented after 200 data bits were processed and it is averaged by $F = 10^3$ independent Monte Carlo runs with independent noise samples for each run.

Table 2. Selected PSO Parameters.

	c_1	c_2	The number of particles	The number of iterations	Constraints	$w_{\max}$	$w_{\min}$	Inertia weight feature
HPSO	2	2	20	20	Hard constraint	0.9	0.4	Single and linearly
APSO						1.1	0.1	Multiple and used μ–law

Comparison results with other estimators, including the GMUSIC, HPSO-GMUSIC and APSO-GMUSIC with $\mu = 100$ to DOA estimation error are presented. Figure 3 depicts the convergence in terms of DOA RMSE *versus* the number of iterations. As a result, the HPSO-GMUSIC requires more iterations to achieve convergence. Note that the proposed APSO-GMUSIC achieves fast convergence with the selected parameters, which means that it needs less iterations to approach the desired global extreme. Figure 4 shows the required number of calculating fitness function (*B*) *versus* number of particles. For the number of particles in the population, more particles may increase success in searching for optima due to sampling state space more thoroughly. However, more particles require more evaluation cost. The HPSO-GMUSIC needs more particles to approach the desired global extreme. It is confirmed that the proposed PSO-based searching approaches can reduce the computational complexity of the GMUSIC due to the searching process. As expected, this figure also provides a great improvement over the convergence rate on optimization problems. In fact, additional adaptive multiple inertia weight operation can improve the searching speed and RMSE performance further. Figure 5 presents the RMSE of DOA estimation *versus* varying angular spreads. We note that

the subspace-based techniques show serious degradation when faced with local scatters. Local scattering may be viewed as a form of model error and gives rise to the perturbation of the noise subspace. Again, these figures show that the proposed APSO-GMUSIC method yields significantly superior performance over the other methods in the presence of local scatters. For comparison, the result of GA-GMUSIC estimator is also provided. The same parameters of GA-GMUSIC estimator are used in [17]. Figure 6 shows the RMSE *versus* different SNR of the desired user for angular spreads $2\triangle_p = 1°$. For the low SNR case, all of methods may produce highly biased estimates. Clearly, with the compatible searching resolution, the APSO-GMUSIC can save the required number of searching grids and improve the RMSE performance, as compared with the other estimators. The GA-GMUSIC has a poor performance than HPSO-GMUSIC and APSO-GMUSIC. It is well known that premature convergence degrades the performance of GA and reduces the search ability [18]. In addition, it has been shown that the performance of the PSO algorithm with linearly decreasing inertia weight has the ability to quickly converge, the PSO may lack global search ability at the end of a run and may fail to find the required optima in cases when the problem to be solved is too complicated and complex [19]. But to some extent, this can be overcome by employing the proposed adaptive multiple strategy for adjusting the inertia weight. Finally, in Figure 7, we compare the RMSE performance against the number of active user P, given SNR = 20 dBW and angular spread $2\triangle_p = 1°$. Basically, the RMSE is increased quite steadily with the increase of P. It can be observed that the APSO-GMUSIC obtain more performance improvement when the number of users P is reasonably increasing. Among them, the proposed APSO method achieves the lowest RMSE.
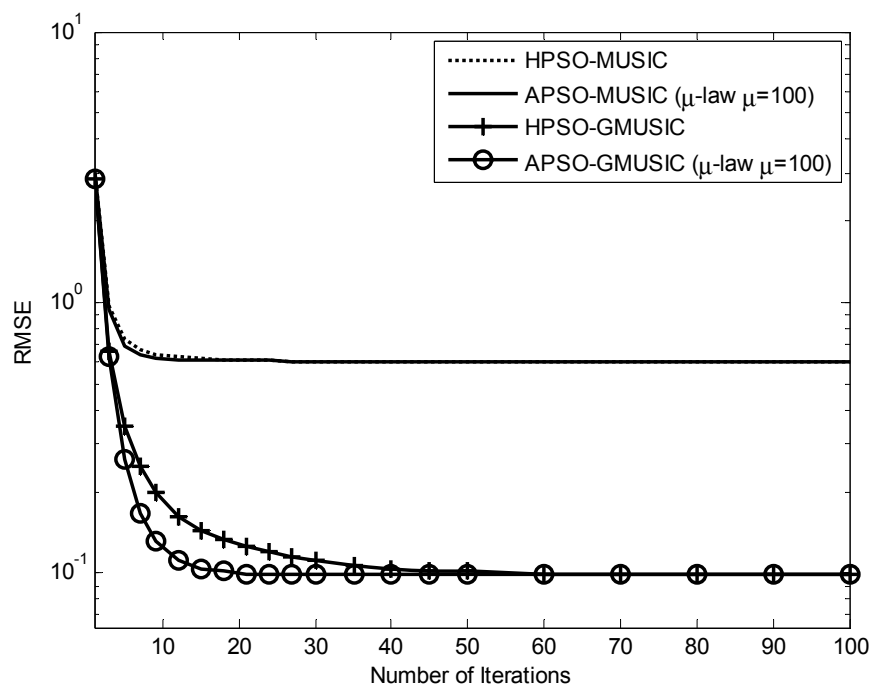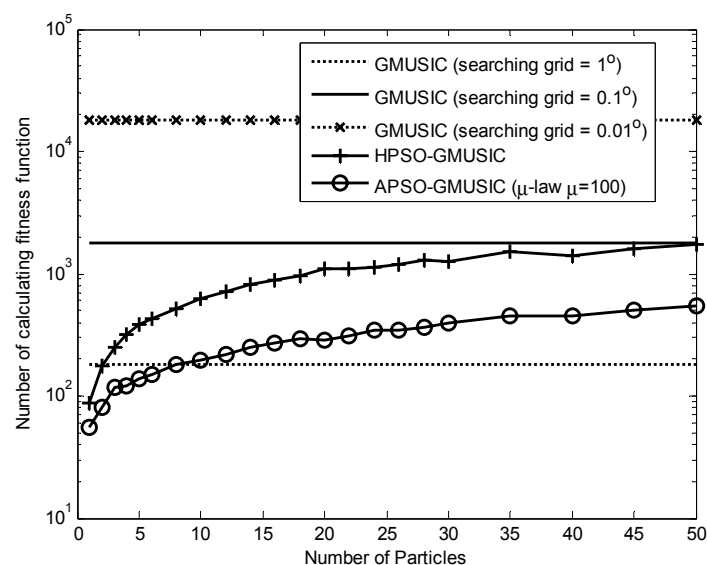
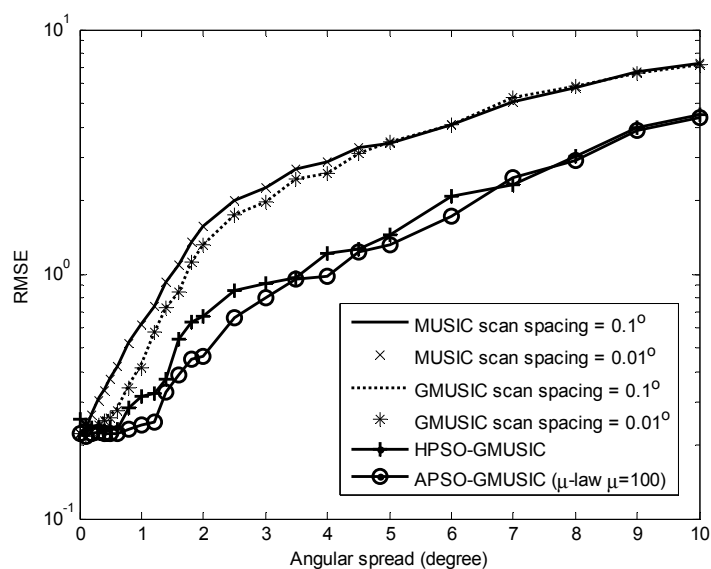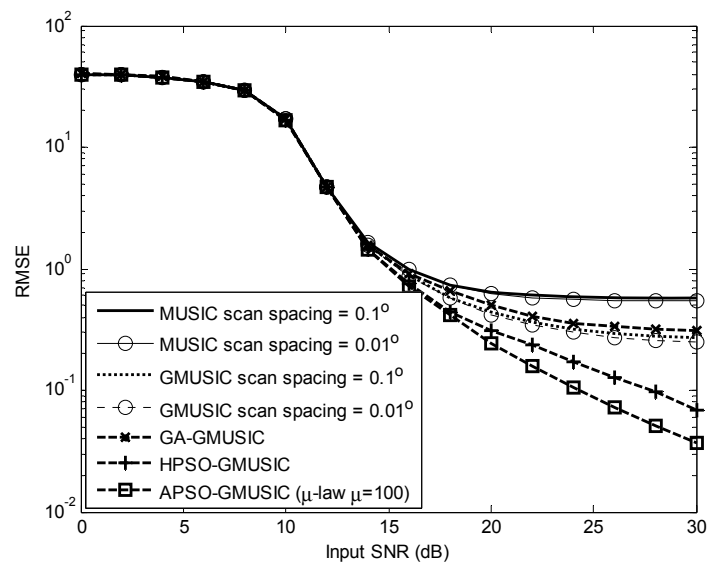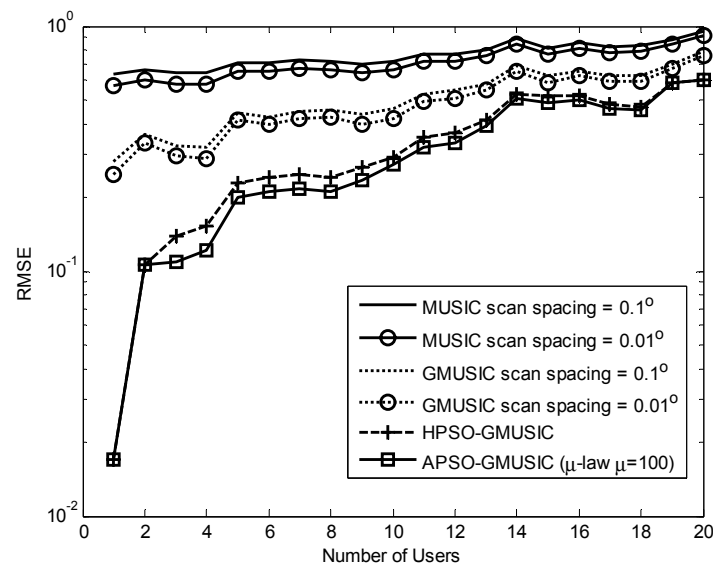**Figure 3.** DOA RMSE *versus* the number of iterations.

Figure 4. The required number of calculating fitness function (*B*) *versus* number of particles.

Figure 5. RMSE of DOA estimation *versus* angular spread.

Figure 6. RMSE of DOA estimation *versus* the input SNR of desired user for $2\Delta_p = 1°$.

Figure 7. RMSE of DOA estimation *versus* the number of users for SNR = 20 dBW and $2\Delta_p = 1°$.

5. Conclusions

In this paper, we have presented a PSO based searching DOA estimation method, named APSO, which uses the GMUSIC searching function for CDMA signals. With the code-matched filter, the MAIs after code-decorrelation appear as noises for CDMA signals. The rooting MUSIC method is suboptimal in the presence of the noise and MAI and the GMUSIC cannot be implemented by using the polynomial rooting approach. However, the proposed techniques reduce the required search grids for the conventional spectral searching estimators. In the previous work on DOA estimation using PSO algorithm [10], hard constraints have been taken, during each iteration of PSO algorithm. In this paper, multiple inertia weight has been incorporated in PSO. In conjunction with a modified PSO for angle searching, the proposed approach can reduce the required computational load for the conventional spectral searching MUSIC estimator with local scattering. Moreover, the convergence of the proposed approach is much faster. Computer simulations have demonstrated the effectiveness of the proposed approach. Furthermore, a common drawback with MUSIC-like technique is that it is a suboptimal estimator and tends to suffer from low performance due to low SNR, small sample size, and correlated sources. Thus, how to develop a cheaper way and work well with correlated or even coherent sources will be important in the future work.

Acknowledgment

References

1. Ottersten, B. Array Processing for Wireless Communication. In *Proceedings of 8th IEEE Signal Processing Workshop on Statistical Signal and Array Processing*, Corfu, Greece, 24–26 July 1996; pp. 466–473.

2. Astely, D.; Ottersten, B. The effects of local scattering on direction of arrival estimation with MUSIC. *IEEE Trans. Signal Proc.* **1999**, *47*, 3220–3234.

3. Astely, D.; Ottersten, B.; Swindlehurst, A.L. Generalised array manifold model for wireless communication channels with local scattering. *IEE Proc. Radar, Sonar Navig.* **1998**, *145*, 51–57.

4. Schmidt, R.O. Multiple emitter location and signal parameter estimation. *IEEE Trans. Antennas Propag.* **1986**, *34*, 276–280.

5. Chiang, C.T.; Chang, A.C. DOA estimation in the asynchronous DS-CDMA system. *IEEE Trans. Antennas Propag.* **2003**, *51*, 40–47.

6. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In *Proceedings of IEEE International Conference on Neural Networks*, Path, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.

7. Shi, Y.; Eberhart, R. A Modified Particle Swarm Optimizer. In *Proceedings of IEEE International Conference on Evolutionary Computation*, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.

8. Shi, Y.; Eberhart, R. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In *Proceedings of the Congress on Evolutionary Computation*, La Jolla, CA, USA, 16–19 July 2000; Volume 1, pp. 84–88.

9. Shi, Y.; Eberhart, R. Comparison between genetic algorithms and particle swarm optimization. *Lect. Notes Comput. Sci.* **1998**, *1447*, 611–616.

10. Li, M.; Lu, Y. Maximum likelihood DOA estimation in unknown colored noise fields. *IEEE Trans. Aerospace Electron. Syst.* **2008**, *44*, 1079–1090.

11. Li, M.; Lu, Y. Source bearing and steering-vector estimation using partially calibrated arrays. *IEEE Trans. Aerospace Electron. Syst.* **2009**, *45*, 1361–1372.

12. Zeng, J.; He, Z.; Liu, B. Maximum Likelihood DOA Estimation Using Particle Swarm Optimization Algorithm. In *Proceedings of CIE '06 International Conference on Radar*, Shanghai, China, 16–19 October 2006; pp. 1–4.

13. Niknam, T.; Mojarrad, H.D.; Nayeripour, M. A new hybrid fuzzy adaptive particle swarm optimization for non-convex economic dispatch. *Int. J. Innov. Comput. Inf. Control (IJICIC)* **2011**, *7*, 189–202.

14. Chang, A.C.; Jen, C.W. Subspace-based techniques for 2-D DOA estimation with uniform circular array under local scattering. *J. Chin. Inst. Eng.* **2006**, *29*, 663–673.

15. Shi, Y.; Eberhart, R.C. Empirical Study of Particle Swarm Optimization. In *Proceedings of the Congress on Evolutionary Computation*, Washington, DC, USA, 6–9 July 1999; Volume 3, pp. 1945–1950.

16. Haykin, S. *Communication Systems*, 4th ed.; John Wiley & Sons: New York, NY, USA, 2001.

17. Chang, A.C.; Hung, J.C. Combining GA and INWES approach to DOA estimation of CDMA signals. *Neural Comput. Appl.* **2010**, *19*, 1003–1011.

18. Fogel, D.B. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*; IEEE: New York, NY, USA, 1995.

19. Panigrahi, B.K.; Pandi, V.R.; Das, S. Adaptive particle swarm optimization approach for static and dynamic economic load dispatch. *Energy Convers. Manag.* **2008**, *49*, 1407–1415.