# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**4,800**
Open access books available

**122,000**
International authors and editors

**135M**
Downloads

**154**
Countries delivered to

Our authors are among the

**TOP 1%**
most cited scientists

**12.2%**
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

## Interested in publishing with us? Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

# 8

# Robust Underdetermined Algorithm Using Heuristic-Based Gaussian Mixture Model for Blind Source Separation

Tsung-Ying Sun, Chan-Cheng Liu, Tsung-Ying Tsai, Yu-Peng Jheng, Jyun-Hong Jheng
*National Dong Hwa University*
*Taiwan, R.O.C.*

## 1. Introduction

Blind source separation (BSS) involves recovering unobserved source signals from several mixed observations, typically obtained at the output of a set of sensors. Each sensor receives a different combination of the source signals. The adjective "blind" emphasizes the fact that: first, the source signals are not observed; and next, no information is available about the mixture. The assumption is often held physically that the source signals are mutually independent.

Recently, BSS in signal processing has received considerable attention from researchers, due to its numerous promising applications in the areas of biomedical signal processing, digital communications and speech signal, sonar, image processing, and monitoring (Cichocki & Unbehauen, 1996), (Tangdiongga et al, 2001), (Yilmaz & Rickard, 2004), (Herault & Juten, 1986). A number of blind separation algorithms have been proposed based on different separation models. These algorithms play increasingly important roles in many applications. Since the pioneering work of Jutten and Herault (Herault & Juten, 1986), a variety of algorithms have been proposed for BSS. In general, the existing algorithms can be divided into five major categories: neural network-based algorithms (Cichocki & Unbehauen, 1996), (Zhang & Kassam, 2004), density model-based algorithms (Amari et al, 1997), (Lee et al, 1999a), algebraic algorithms (Belouchrani et al, 1997), (Li & Wang, 2002), information-theoretic algorithms (Pajunen, 1998), (Pham & Vrins, 2005) and space-based algorithms (Yilmaz & Rickard, 2004), (Lee et al, 1999b).

A source signal with sparse representation means that at most one value of the signal isn't zero at an instant, making the vector of sensor signals (mixtures) equivalent to some mixing vector. Therefore, the sparse-based BSS problem could be solved by searching for mixing vectors; moreover, recovering source signals. Like the mixtures with sparse representation, each base vectors of the unknown mixing matrix will be displayed on a 2-D plane coordinate as a directional line when two sensors are used. The sparse representation is first introduced in underdetermined BSS by Lee et al. (Lee et al, 1999b). After its introduction, several related methods have been proposed continuously for solving underdetermined BSS

cases. Bofill and Zibulevsky proposed a potential function to produce an approximate curve, which is able to describe the histogram of mixtures (Bofill & Zibulevsky, 2001). Shi et al. proposed the generalized exponential mixture model to approximate the distribution of mixtures without any predefined parameters (Shi et al, 2004). Besides, clustering methods such as fuzzy clustering, *k*-means, and other extensions were proposed to search for the mixing matrix (Grady & Pearlmutter, 2004), (Liu et al, 2006), (Vaerenbergh & Santamaria, 2006). These aforementioned algorithms provide good performance in well-conditioned mixing matrices which include identifiable mixing vectors.

Generally speak, the efficiency of signal recovering is dependent upon the precision of mixing matrix identification in a BSS problem; however, some practical and difficult conditions occur in an outside of the lab environment. First, when source signals are not sparse enough, the non-sparseness of the signals has affects the identification with problems like noise. Second, distances between mixing vectors are not far enough to distinguish them; therefore, two or more vectors would be regarded as one. Here, these two problems are termed an ill-conditioned BSS case. So far conventional algorithms produce unsatisfactory performance in such a case since many sub-solutions arise from these interferences. In this study, these difficult will be aimed by a heuristic method and a well-known statistic model.

Heuristic learning has been utilized to tackle similar problems encountered in other BSS categories. For example, the neural network-based BSS algorithms use the Genetic Algorithm (GA) (Yue & Mao, 2002) or Particle Swarm Optimization (PSO) (Song et al, 2007) to deal with linear mixing matrix or nonlinear mixing matrix identification; however, space-based BSS algorithms have never adopted such a heuristic learning process except that is published in (Liu et al, 2007).

Recently Gaussian mixture model (GMM) has been suggested to learn or model a set has multiple mixing data through the maximum-likelihood (ML) estimator or the expectation-maximization (EM) algorithm. And, the validity has been demonstrated in many research fields (Hedelin & Skoglund, 2000), (Todros & Tabrikian, 2007), (Nikseresht & Gelgon, 2008). Since there is not only single mixing vector in a BSS problem, GMM with multiple Gaussian models should be associated. However, the question then arises about the effect of initial parameters and falling into a local optimum from training by ML or EM. At first, the most information is unobservable in BSS problem; thus, the hint is too short to give proper initial parameters. Second, mixture outliers or ill-conditioned mixing vectors would cause a lot of local optimum; therefore, we think that an ability of widely exploring is weighty enough to affect the separate performance.

According to above analyses, this study develops a flexible GMM whose parameters are trained by PSO. The fitness function of PSO is designed to evaluate the inverse of sum of densities of GMM. When centers of all Gaussian models is close to the directions of all mixing vectors, the fitness value would approximate to the low bound. In order to make PSO more efficient, the representation of mixtures are modified from 2-D to 1-D; meantime, the boundary is normalized into the interval $[0,1]$. The search range of PSO therefore becomes more compact. Further, the particle update function of PSO is improved by using a cluster information to replace the global best (*Gb)*. This improvement is according to the property of the sparse signal so it is helpful to speed convergence and enhance accuracy. The simulations involving well-conditioned mixing vectors, ill-conditioned mixing vectors and unknown number of source are designed. In order to present advantages of proposed

algorithm, some existing underdetermined BSS algorithms and GMM-based algorithms will be performed in the simulation for performance comparison.

The remainder of this study is organized as follows: Section II presents the fundamental of BSS consisting of mixing model and recovery methods. Section III introduces the standard PSO and GMM. Section IV presents details of the proposed algorithm. Section V presents several BSS simulations and displays the compared results. The validity of appended parameters are analyzed and confirmed in Section VI. Section VII draws a brief conclusion for this work.

## 2. Underdetermined Blind Source Separation

### 2.1 Mixtures in Sparse Representation

In a situation where the sparse source signals are unobservable, $\mathbf{s}(t)=[s_1(t),\ldots,s_n(t)]^T$ which is zero-mean and is mutually (spatially) statistically independent (or as independent as possible), where $n$ denotes the number of sources, $[\cdot]^T$ denotes the transpose operation and $t$=1, …, $N$ is the instant time of sampling. The term "Sparse" means that only a small number of the $s_i$ differs significantly from zero. The degree of sparsity is evaluated by the probability density function (PDF) as follows:

$$P_{saprse_i}(s_i)=\alpha_i\delta(s_i)+(1-\alpha_i)f_i(s_i),\quad i=1,2,\ldots,n \tag{1}$$

where $\alpha_i$ is the probability that a source is inactive, $\delta(\cdot)$ denotes Dirac's delta and $f_i(s_i)$ is the PDF of the $i$th source when it is active (Luengo et al, 2005). The actual acoustics have a higher degree of sparsity in the frequency domain than in the time domain. Consequently, this study addresses the source signals that fulfill the requirements of sparse in the frequency domain and not in the time domain.

The available sensor vector $\mathbf{x}(t)=[x_1(t),\ldots,x_m(t)]^T$, where $m$ is the number of sensors, is given by

$$\mathbf{x}(t)=\mathbf{A}\mathbf{s}(t) \tag{2}$$

where $\mathbf{A}\in\mathbf{R}^{m\times n}$ is an unknown mixing matrix and is nonsingular. The definition of an underdetermined case is one that satisfies $n\geq m$. Because two is the most applicatory number of sensors to such a BSS problem, $m=2$ is considered in this study. Therefore, eq. (2) could be rewritten as

$$\mathbf{x}(t)=\begin{bmatrix}x_1(t)\\x_2(t)\end{bmatrix}=\begin{bmatrix}a_{11} & a_{12} & \cdots & a_{1n}\\a_{21} & a_{22} & \cdots & a_{2n}\end{bmatrix}[s_1(t)\ \ s_2(t)\ \ \cdots\ s_n(t)]^T \tag{3}$$

where the components of mixing matrix could be presented as

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \end{bmatrix} \tag{4}$$

The feasibility of applying such an algorithm to identify sparse representation is affected by the sparsity of source signals and the density of mixing vectors. Then, the assumption that the distance between two arbitrary mixing vectors is less than the doubled sum of variances of distribution for the corresponding mixtures is held in this study.

The process of BSS can be divided into two steps: the first is the unknown mixing matrix identification which will be discussed in Section IV. The second is source signals recovery by the estimation of mixing matrix, described in the next subsection.

### 2.2 Source Signal Recovery

According to the estimated mixing matrix, sparse source signals can be recovered by maximizing the posterior distribution that is formed as (Shi et al, 2004)

$$P(\mathbf{s}|\mathbf{x}, \mathbf{A}) = \prod_{t=1}^{N} P(\mathbf{s}(t)|\mathbf{x}(t), \mathbf{A}) \tag{5}$$

According to eq. (2) and Bayes' rule, the log-likelihood can be obtained by taking the logarithm of eq. (2):

$$L(\mathbf{s}) = \sum_{t=1}^{N} \left\{ -0.5(\mathbf{x}(t) - \mathbf{A}\mathbf{s}(t))^T \Sigma^{-1} (\mathbf{x}(t) - \mathbf{A}\mathbf{s}(t)) + \log\{P(\mathbf{s}(t))\}\right\} + \beta \tag{6}$$

where $\Sigma^{-1}$ indicates the noise covariance matrix and $\beta$ is a constant irrelevant to $\mathbf{s}(t)$, and $(\cdot)^T$ denotes the transpose operation. In order to maximize eq. (6), the gradient of eq. (6) is derived with respect to $\mathbf{s}(t)$ as

$$\nabla_{\mathbf{s}(t)} L(\mathbf{s}(t)) = \mathbf{A}^T \Sigma^{-1} (\mathbf{x}(t) - \mathbf{A}\mathbf{s}(t)) + \nabla_{\mathbf{s}(t)} \log\{p(\mathbf{s}(t))\} \tag{7}$$

Therefore, the original signals can be recovered gradually by the following iteration:

$$\mathbf{s}^j(t) = \mathbf{s}^{j-1}(t) + \eta \nabla_{\mathbf{s}(t)} L(\mathbf{s}^{j-1}(t)) \tag{8}$$

where the superscript of **S** indicates the iteration index.

## 3. Introduced Techniques

### 3.1 Gaussian Mixture Model

A Gaussian mixture PDF for d-dimensional random vectors **X** is a weighted sum of

densities

$$f_{\mathbf{X}|\mathbf{\Theta}}\left(\mathbf{X}|\mathbf{\Theta}\right)=\sum_{i=1}^{M}\rho_i f_{\mathbf{X}|\theta_i}\left(\mathbf{X}|\theta_i\right) \tag{9}$$

where $\rho_i$ are the component weights, $M$ is the number of class and the component densities are Gaussian

$$f_{\mathbf{X}|\theta_i}\left(\mathbf{X}|\theta_i\right)=f_{\mathbf{X}|\mu_i,\mathbf{\Sigma}_i}\left(\mathbf{X}|\mu_i,\mathbf{\Sigma}_i\right)=\frac{1}{\left(2\pi\right)^{d/2}\left|\mathbf{\Sigma}_i\right|^{1/2}}e^{-\frac{1}{2}(\mathbf{X}-\mu_i)^T\mathbf{\Sigma}_i^{-1}(\mathbf{X}-\mu_i)} \tag{10}$$

with mean vectors $\mu_i$ and covariance matrices $\mathbf{\Sigma}_i$. The weights are constrained by $\rho_i>0$ and $\sum_{i=1}^{M}\rho_i=1$. The parameters of the Gaussian mixture density is represented by the set

$$\mathbf{\Theta}=\left\{\rho_1,\ldots,\rho_M,\mu_1,\ldots,\mu_M,\mathbf{\Sigma}_1,\ldots,\mathbf{\Sigma}_M\right\} \tag{11}$$

Generally, the expectation-maximization (EM) algorithm is a widely used procedure for maximum-likelihood (ML) estimation. It is an iterative algorithm where in each iteration over the same database a monotonic increase in the log-likelihood, $L(\mathbf{\Theta})$, is guaranteed, i.e., $L\left(\mathbf{\Theta}^{(k+1)}\right)\geq L\left(\mathbf{\Theta}^{(k)}\right)$, where $\mathbf{\Theta}^{(k)}$ is the value of the parameter set $\mathbf{\Theta}$ at iteration $k$ (Hedelin & Skoglund, 2000), (Nikseresht & Gelgon, 2008). A poor initialization of set $\mathbf{\Theta}$ would have great effect upon final performance; however, some elements are hard to give suitable initial values by experience of user. Consequently, this paper replaces the iterative method by PSO to obtain a more precise solution.

### 3.2 Heuristic Learning

The PSO is a population based optimization technique proposed by (Eberhart & Kennedy, 1995). The population is referred to as a *swarm*. The particles move and fast converge to local and/or global optimal position(s) over a small number of generations.

A swarm in PSO consists of a number of particles. Each particle represents a potential solution to the optimization task. All of the particles iteratively explore potential solutions through evolution. Each particle moves to a new position according to the new velocity which includes its previous velocity, and the directional vectors according to its own past best solution and global best solution. The best solution is then kept; each particle accelerates in the directions of not only the local best solution but also the global best position. If a particle discovers a new solution better than the global best solution, other particles will move closer to it in order to explore the region with more depth (Gudise & Venayagamoorthy, 2003).

Let $sz$ denotes the swarm size. In general, there are three attributes, the particles' current position $p_i$, current velocity $v_i$, and local best position $Pb_i$, for particles in the search space to present their features. Each particle in the swarm is iteratively updated according to the

aforementioned attributes. Assuming that the fitness function $f(\cdot)$ is to be minimized, the new velocity of every particle is updated as

$$v_{ij}(g+1) = v_{ij}(g) + \alpha_1 \cdot r_1(g)\left[Pb_{ij}(g) - p_{ij}(g)\right] + \alpha_2 \cdot r_2(g)\left[Gb_j(g) - p_{ij}(g)\right] \tag{12}$$

where $Pb_{ij}$ denotes the local best position of the $i$th particle and $Gb_j$(g) denotes the global best position at the $g$th generation. For all the index of dimension, $j = 1,2,\ldots,k$, $v_{ij}$ is the velocity of the $j$th dimension of the $i$th particle, $\alpha_1$ and $\alpha_2$ denote the *acceleration coefficients*, $r_1$ and $r_2$ are elements from two uniform random sequences in the range (0, 1), $g$ is the number of generations and has to be bounded in [$Vmin, Vmax$]. The new position of a particle is calculated as follows:

$$p_i(g+1) = p_i(g) + v_i(g+1) \tag{13}$$

The local best position of each particle is updated by

$$Pb_i(g+1) = \begin{cases} Pb_i(g), & \text{if } f(p_i(g+1)) > f(Pb_i(g)) \\ p_i(g+1), & \text{otherwise} \end{cases} \tag{14}$$

And the global best position $Gb$ found from all particles during the previous three steps is defined as

$$Gb(g+1) = \arg\min_{Pb_i}\left\{f(Pb_i(g+1)),\ 1 \le i \le sz\right\} \tag{15}$$

Since Kennedy and Eberhart (Eberhart & Kennedy, 1995) introduced PSO in 1995, many researchers have worked on improving its performance in various ways. One of the variants called the standard PSO (Lin &. Feng, 2007), introduced by Shi and Eberhart (Shi & Eberhart, 1998), incorporates a parameter called inertia weight of velocity $\alpha_0$ into the original PSO. The new velocity update algorithm is shown as follows:

$$v_{ij}(g+1) = \alpha_0 \cdot v_{ij}(g) + \alpha_1 \cdot r_1\left[Pb_{ij}(g) - p_{ij}(g)\right] + \alpha_2 \cdot r_2\left[Gb_j(g) - p_{ij}(g)\right] \tag{16}$$

This plays the role of balancing the global search and local search. It can be a positive constant or even a positive linear or nonlinear function of time. This value is typically setup to vary linearly from 1 to near 0 during the course of a training run. Note that this is reminiscent of the temperature adjustment schedule found in Simulated Annealing algorithms. The inertia weight is also similar to the momentum term in a gradient descent neural network training algorithm.

Although there are numerous variants for PSO, these methods spend too much time finishing fitness evaluations, and will present similar results in the early parts of convergence. To reach requirements of on-line separation, only a limited amount of computational time is available to produce a reasonable solution. These limitations will require an efficient and simple method. Hence, the variant of PSO with inertia weight is chosen (Shi & Eberhart, 1998).

## 4. Identify Mixing Vector by PSO-based GMM

### 4.1 PSO-based GMM

Suppose that only two sensors are available (i.e. $m = 2$) here, and then received mixtures are represented as $[x_1(t), x_2(t)]^T$, $t = 1,2,\ldots,N$,. Section II mentions that source signals are sparse and the mixtures center around the mixing vectors on the $x_1 - x_2$ coordinate plane. Thus, unobservable mixing vectors could emerge from these clusters of mixtures. Because $n$ source signals will form $n$ clusters of mixtures (mixing vectors), GMM with $n$ densities is capable of expressing the distribution of mixtures. Further, the position of mixing vectors can be obtained from the centers of Gaussian model.

In order to find out the optimal parameters of GMM, the particles of PSO are regarded as a parameter set of GMM, $p = [\mu_1, \mu_2, \cdots, \mu_n]$. Because the estimate of each mixing vector should have equal importance in BSS case, the complexity of PSO could be reduced by omitting the estimate of weights, $\rho_i$. And, since covariance can be evaluated from classed mixtures, the estimate of covariance matrices $\mathbf{C}_i$ also could be omitted.

About the search space of PSO, there are several disadvantages when each element of particles is encoded into 2-D vector representation. First, there is no exact boundary for searching. Second, different elements would be mapped to the same solution. For instance $\mu = [2\times10^5, 10^5]$ is equivalent to $\mu = [1, 0.5]$. It would cause that particles explore in an infinite range; moreover, many equivalent solutions would pin down to each other. In order to enhance efficiency of optimization, the 2-D vector representation is replaced by 1-D angle representation; further, the angles are normalized in the interval from -1 to 1, i.e. $p \in [-1,1]^n$. The mixtures are therefore rewrote as

$$\hat{x}(t) = \frac{2\arctan\left(x_1(t)/x_2(t)\right)}{\pi}, \text{ where } \forall\ \hat{x} \in [-1, 1] \tag{17}$$

*t=1,2,...,N*; then, the search space of PSO become compact.

### 4.2 Improved PSO

Some potential improvements can be made to the exploration of PSO. In Fig. 1, which presents the distribution of mixtures, it could be seen that the two directions formed by the gathering mixtures imply two real mixing vectors. According to PSO's evolution, $Gb$ would slowly approach these directions. In Fig. 1, the solid lines placed far from these directions indicate the $Gb$ in the initial generation. The mixtures are further divided into

two clusters according to their distances to $Gb$; and then, cluster centers denoted by the dotted lines in Fig. 1 could be calculated. The result should reveal that the dotted lines are closer to the real mixing vectors than the solid lines.

Since mixtures gather toward the mixing vectors, cluster centers are more likely to produce a better solution than $Gb$. Moreover, it not only substantially improves $Gb$ during initial generations, but also fine tunes $Gb$ during final generations. Consequently, cluster information is the more preferable guide for particles compared to $Gb$. The factor $Gb$ is replaced with cluster centers $\mathbf{C}_{bv}$ in eq.(12), which could be rewritten as:

$$\begin{aligned} v_{ij}(g+1) = \alpha_0 \cdot v_{ij}(g) + \alpha_1 \cdot r_1 \left[ Pb_{ij}(g) - p_{ij}(g) \right] \\ + \alpha_2 \cdot r_2 \left[ c_j(g) - p_{ij}(g) \right] \end{aligned} \tag{18}$$

where $\{c_j | j = 1,2,\ldots,k\} \in \mathbf{C}_{bv}$ is a set of cluster centers according to $Gb$, and each component is evaluated by:

$$c_j = \frac{\sum_{i=1}^{cn_j} \xi_i^2 \times \hat{x}_i}{\sum_{i=1}^{cn_j} \xi_i^2} \tag{19}$$

where $\xi_t = \sqrt{x_1^2(t) + x_2^2(t)}$, $j$ denotes the index of the cluster, $cn_j$ denotes the number of mixtures which belongs to the $j$th estimated vector, and $i$ is the index of mixtures. Since the involved signals are sparse, $\xi_i$ could be regarded as a weight to the angle of the $i$th mixture. In other words, mixtures with a larger $\xi$ have a greater effect upon the cluster center that it belongs to, whereas others are noisy or even voiceless.

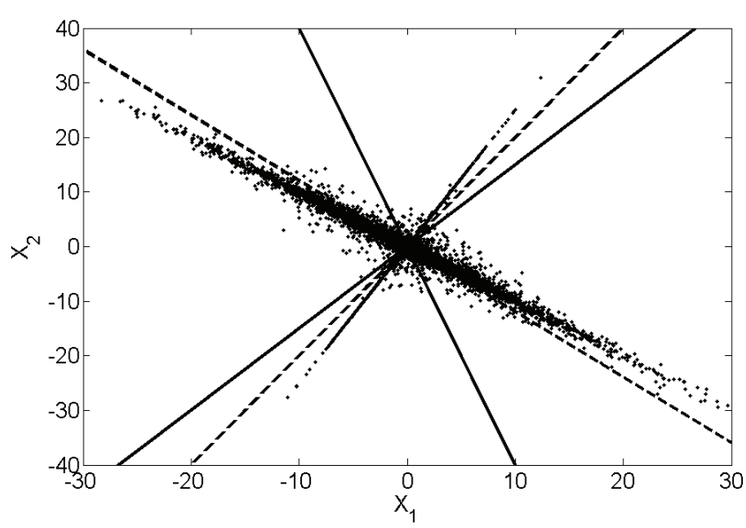

Fig. 1. The cluster vectors (dotted line) are derived from vectors of current *Gb* (solid line). The two directions constructed by gathered mixtures indicate the real mixing vectors.

### 4.3 Objective Function

As for the fitness function of the PSO, the property of sparse mixture distribution is introduced into our design. Since mixtures respectively gather toward mixing vectors and the vector length of mixtures is in respect to the energy of the signal, the fitness function is defined as

$$fit = \left( \sum_{i=1}^{M} f_{\hat{\mathbf{x}}|\mu_i}\left(\hat{\mathbf{X}}|\mu_i\right)\right)^{-1} \tag{20}$$

where the differential angle between the *i*th mixture vector and the nearest estimated vector is calculated by:

$$\Delta\hat{x}_i = \min\left\{\left|\hat{x}_i - \mu_j\right|, i = 1,2,\ldots,N \text{ and } j = 1,2,\ldots,n\right\} \tag{21}$$

Consequently, a small fitness value implies a more accurate estimate to mixing vectors.

### 4.4 Disturbance

Additionally, in order to prevent the search from falling into a local optimum, a disturbance operation is added to PSO. Every current particle is allotted a random value between 0 and 1. A particle will carry out the disturbance sequence if its random value is less than a disturbance rate $P_d$ . A disturbance particle is produced by:

$$p_{dis} = \left[\mu_1, \mu_2, \cdots, \mu_r + \varepsilon, \ldots, \mu_n\right] \tag{22}$$

where $\mu_r$ is a randomly selected dimension of the particle, and $\varepsilon$ is a tiny disturbance factor. PSO begins with more uniformly scattered particles during initial generations, but incorporates more gathered particles during final generations, in the distribution of particles. Therefore, the value of $P_d$ should be dependent upon the current evolutionary state of PSO. A floating $P_d$ was decided to serve this purpose, its change is in respect to linear monotonic increase:

$$P_d(g) = \frac{P_D \times g}{G} \tag{23}$$

where $P_D$ is the maximal $P_d$ choosing from interval $[0,1]$, $g$ is the current generation of PSO, and $G$ is the maximal generation for evolution of PSO.

### 4.5 Algorithm Procedure

The procedure which uses the proposed GMM-PSO to deal with the underdetermined BSS problem is explained in Fig. 2, and the detail is described as following steps:

**Step 1.** Two mixtures mixed by eq. (3) is received.

**Step 2.** The both mixtures are transformed into frequency domain.

**Step 3.** (Start PSO) The initial particles are randomly produced from interval $[0,1]$.

**Step 4.** Fitness values are calculated by eq. (20) for each particle.

**Step 5.** *Pb* and *Gb* are updated by eq. (14) and eq. (15) respectively.

**Step 6.** The cluster centers $\mathbf{C}_{bv}$ are calculated by eq. (19) according the *Gb*; and then, replaces ***Gb***.

**Step 7.** New particles are produced according to eq. (18).

**Step 8.** Disturbance is carried out according a rate referred from eq. (23).

**Step 9.** If terminational condition is reached, i.e. $g = G$, then procedure goes to next step (End PSO); otherwise, it goes back to the Step 4.

**Step 10.** According to the final *Gb* to restructure mixing matrix; and then, recover signals are evaluated by eq. (8).

Further, there are several sub-steps in the Step 4 explaining how to evaluate fitness values:

**Sub-step 1.** Particle are inputted one by one.

**Sub-step 2.** The mixtures are classified into $n$ classes by eq. (21) according to a particle.

**Sub-step 3.** The fitness value is calculated by eq. (20).

**Sub-step 4.** The fitness values of all particles are outputted.

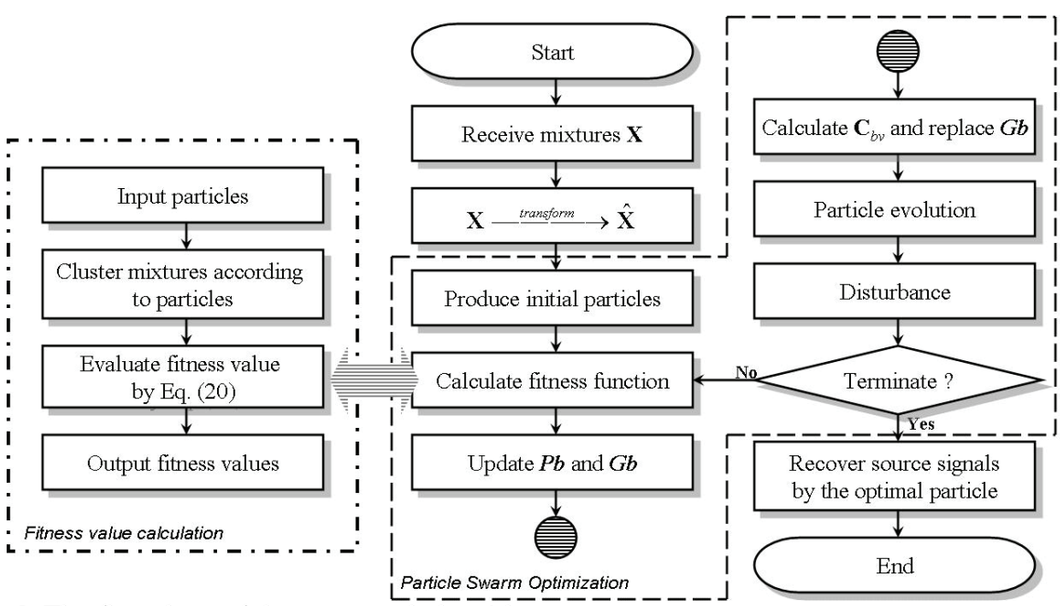


Fig. 2. The flowchart of the proposed algorithm

## 5. Simulations and Results

### 5.1 Description and Parameters Setting

Three other underdetermined algorithms with state of the art are tested in the following simulations to compare with the proposed algorithm. Here, the first one is named PF proposed in (Bofill & Zibulevsky, 2001), the second one is named GE proposed in (Shi et al, 2004), and the last one is our previous work which named FC proposed in (Liu et al, 2006).

In order to confirm validation and robustness of these algorithms, four sparse signals recorded from real sounds are taken for the source signals whose waveforms are shown in Fig. 3 and Fig. 4. In the first BSS case, the first three source signals are mixed by a well-conditioned mixing matrix as

$$\mathbf{A}_{well} = \begin{pmatrix} 0.3714 & 0.4472 & 0.7071 \\ 0.9285 & -0.8944 & 0.7071 \end{pmatrix} \tag{24}$$

It involves distinguishable mixing vectors whose normal angles are $\tilde{\mu}_{well}[0.2422, -0.2952, 0.5000]$ respectively. The distribution of mixtures is plotted in Fig. 5. In the second BSS case, the four source signals are mixed by an ill-conditioned mixing matrix as

$$\mathbf{A}_{ill} = \begin{pmatrix} 0.7071 & 0.7071 & 0.6097 & 0.3714 \\ 0.7071 & -0.7071 & 0.7926 & 0.9285 \end{pmatrix} \tag{25}$$

It involves undistinguishable mixing vectors whose normal angles are $\tilde{\mu}_{ill} = [0.5000, -0.5000, 0.4175, 0.2422]$ respectively since the first vector and the third vector are quite close. The distribution of mixtures is plotted in Fig. 6.

The parameters of compared algorithms are referenced from the original setting of their articles. For example, the grid scale is given 720 and $\lambda$ is entered as 55 in PF. The improved PSO, through the experience of numerous previous experiments, are given the suitable parameters, $sz = 20$ , $P_D = 0.5$ , $\alpha_0 = 0.12$ , $\alpha_1 = 0.3$ and $\alpha_2 = 0.4$ . For generation number, $G = 100$ is given in first case and $G = 200$ is given in second case. For all algorithms, the every simulation will be tested by 30 independent runs. And, the performance is evaluated by mean square error (MSE) as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\tilde{\mu}_i - \mu_i\right)^2 \tag{26}$$

where $\tilde{\mu}_i$ denotes the *i*th real mixing vector, and $\mu_i$ denotes the *i*th estimation. In final, the average of MSE by 30 independent runs will be presented. An estimated set of mixing vector having a small MSE implies a excellent source separation.

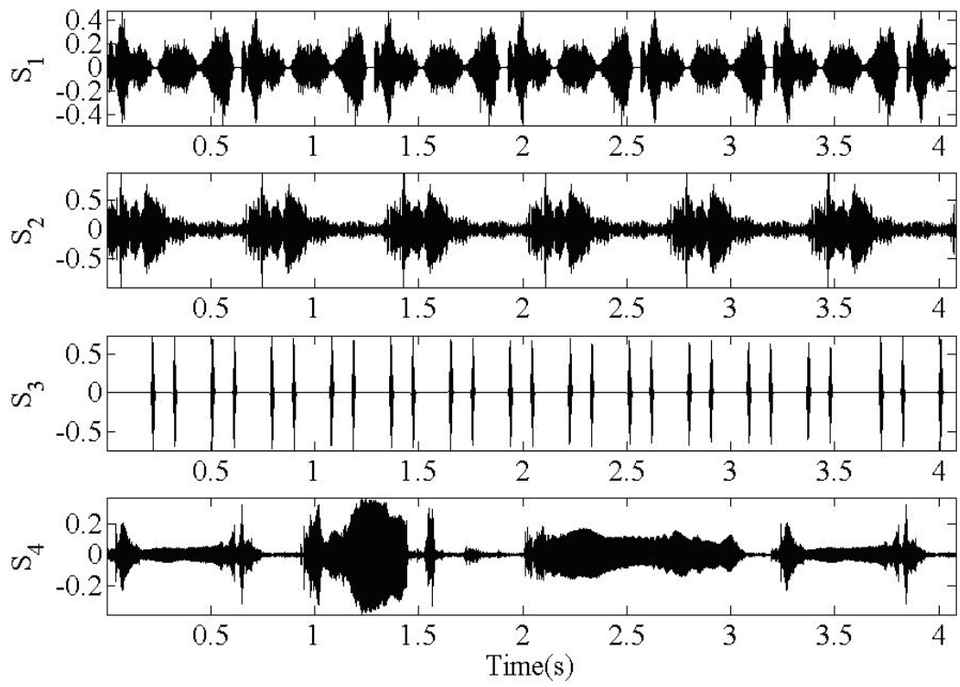


Fig. 3. The waveforms of source signals represented in time domain.

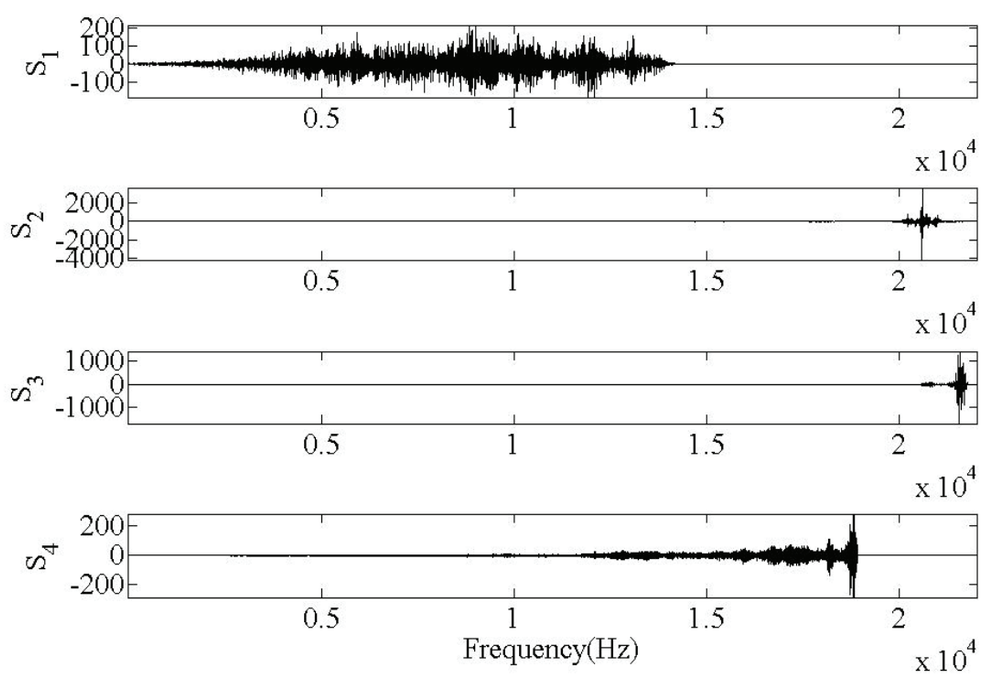


Fig. 4. The waveforms of source signals represented in frequency domain.

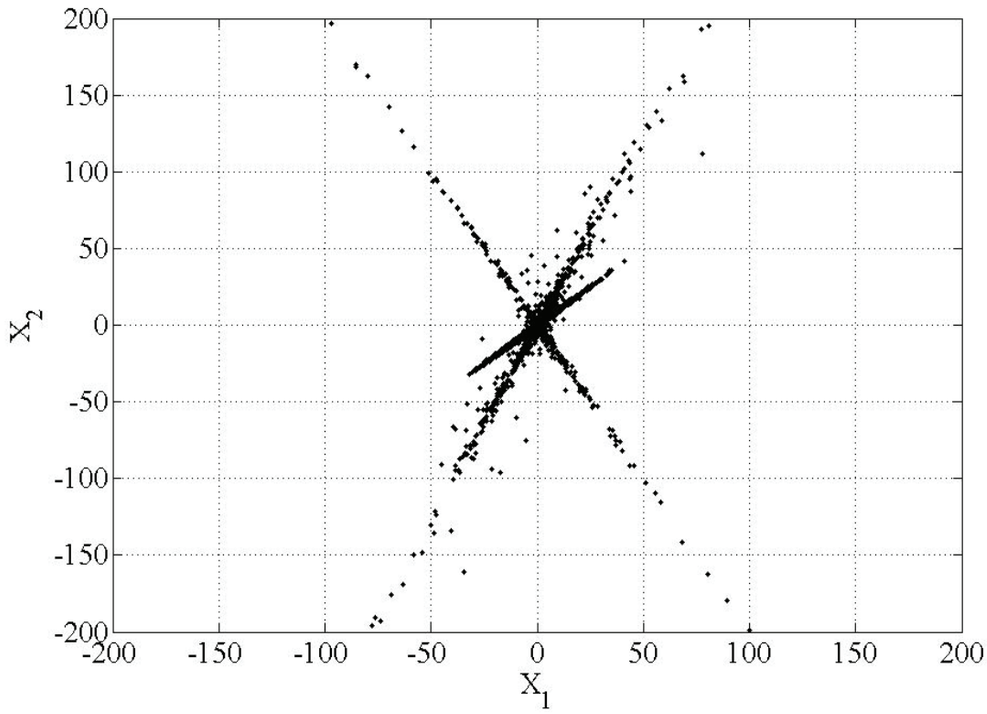


Fig. 5. The distribution of mixtures produced by well-conditioned mixing matrix.

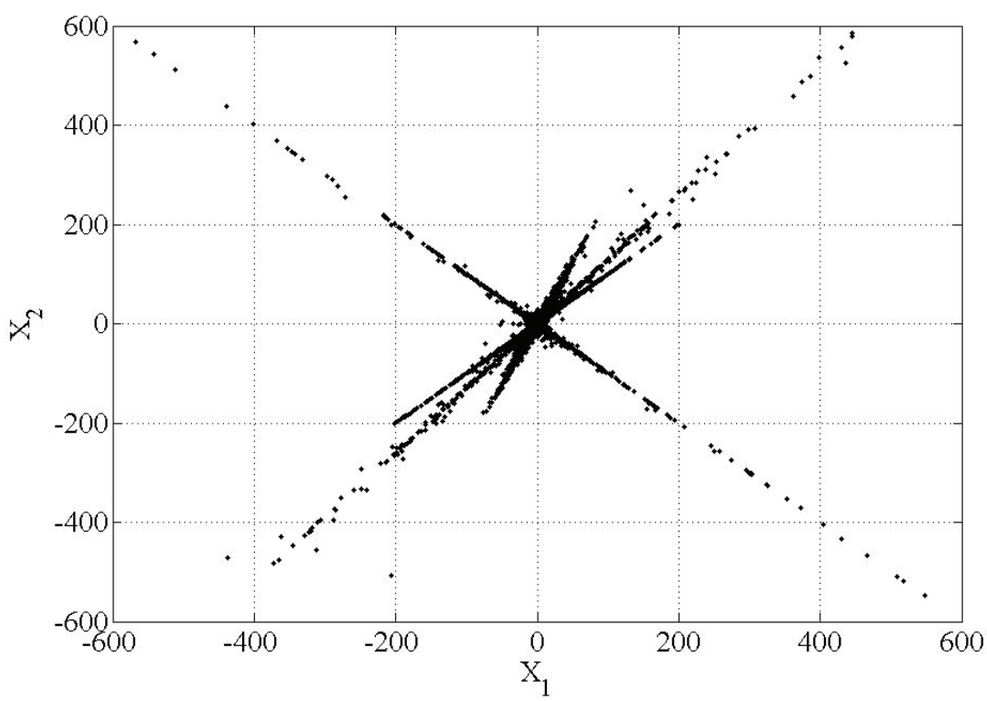


Fig. 6. The distribution of mixtures produced by ill-conditioned mixing matrix.

### 5.2 Results

After two simulations are implemented by the involved algorithms, the compared data about estimating accuracy are presented in Table 1 and Table 2. The both tables describe the real mixing vectors, the average of estimated mixing vectors and the MSE of the four algorithms for well-conditioned case and ill-conditioned case. From these tables, it could be observed that GE algorithm's performance is always unacceptable in all cases. PF algorithm just work acceptably in well-conditioned case, but it fail in ill-conditioned case. FC algorithm is valid in all cases, but its MSE is not better than that of proposed PSO-GMM algorithm.

In order to compare the improved PSO and standard PSO, their average fitness curves are shown in Fig. 7 (well-conditioned case) and Fig. 8 (ill-conditioned case). Form both figures, it could be observed that improved version has better convergent ability on speed and depth; particularly, that in Fig. 8.

| Compared algorithms | | PF | GE | FC | PSO-GMM |
|---|---|---|---|---|---|
| $\tilde{\mu}_1 = 0.2422$ | $\mu_1$ | 0.2497 | 0.2292 | 0.2498 | 0.2421 |
| $\tilde{\mu}_2 = -0.2952$ | $\mu_2$ | -0.2190 | -0.1958 | -0.2903 | -0.2896 |
| $\tilde{\mu}_3 = 0.5000$ | $\mu_3$ | 0.1627 | 0.1402 | 0.5134 | 0.4995 |
| MSE | | 0.0399 | 0.0465 | 8.7110e-05 | 1.0540e-05 |

Table 1. Comparison of results between the four algorithms in well-conditioned BSS case.

| Compared algorithms | | PF | GE | FC | PSO-GMM |
|---|---|---|---|---|---|
| $\tilde{\mu}_1 = 0.5000$ | $\mu_1$ | 0.5520 | 0.6856 | 0.5000 | 0.4998 |
| $\tilde{\mu}_2 = -0.5000$ | $\mu_2$ | -0.4895 | -0.6469 | -0.4982 | -0.4929 |
| $\tilde{\mu}_3 = 0.4175$ | $\mu_3$ | 0.5639 | 0.5687 | 0.3996 | 0.4176 |
| $\tilde{\mu}_4 = 0.2422$ | $\mu_4$ | 0.7494 | -0.0817 | 0.2426 | 0.2426 |
| MSE | | 0.0704 | 0.0460 | 8.0060e-05 | 1.2662e-05 |

Table 2. Comparison of results between the four algorithms in ill-conditioned BSS case.

## 6. Discussion

In comparing the proposed PSO-GMM with related BSS algorithms, the performance of GE algorithm is sensitive to predefined parameters. Tt exhibited a large value in MSE because of the lack of perfect initiations. Unfortunately, there is no rule or criterion that can be referred to for choosing suitable initiations. The PF algorithm is available in well-conditional case, and it does not involves any random initiation. However, the PF algorithm is not robust enough to deal with a complex problem because its settings of parameters is not for general-purpose; moreover, there are no instructions to guide a user on how to adjust them to suit other specific cases. The FC algorithm and PSO-GMM algorithm are efficient and robust enough to handle whether a general toy BSS case or an advanced BSS case. For further comparison between the both algorithms, it can be discovered that PSO method explores variant potential solutions; therefore, its accuracy is more excellent than FC algorithm. For the different PSO versions, the improved PSO exhibits a better convergent curve because it have the additional mechanism which enhances and replaces the globel best solution to rapidly drag particles toward a solution with an exact direction and distance during whole generations.

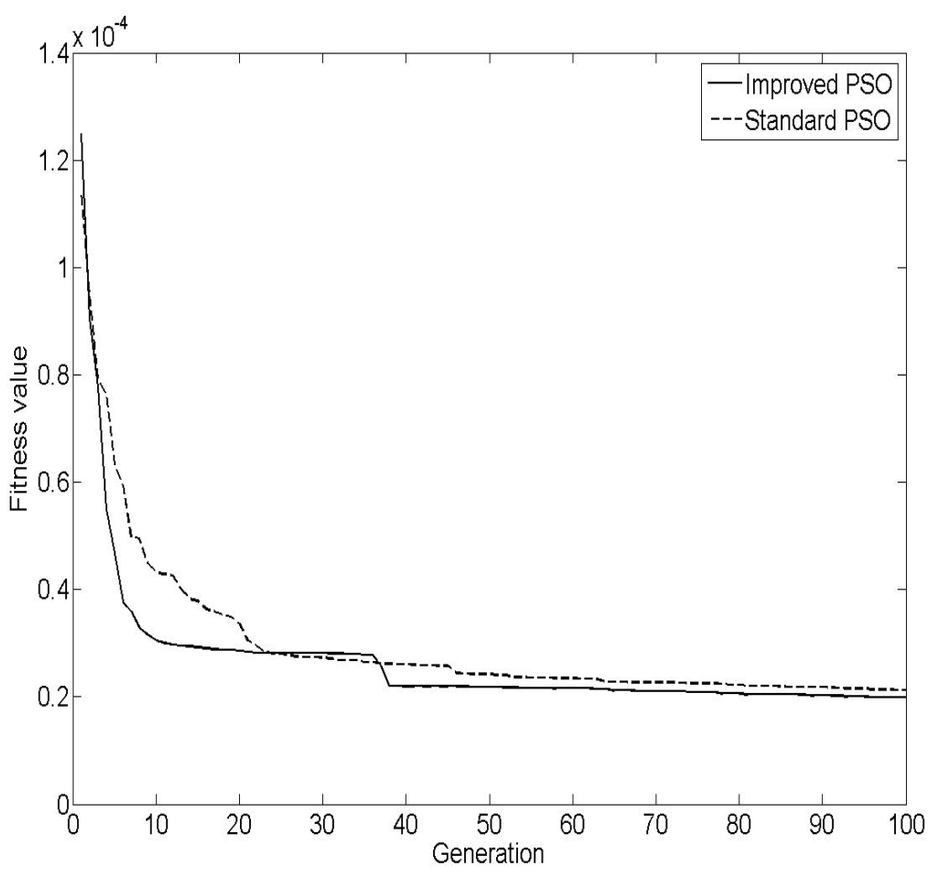


Fig. 7. Fitness convergence comparison between improved PSO and standard PSO for well-conditioned BSS case.

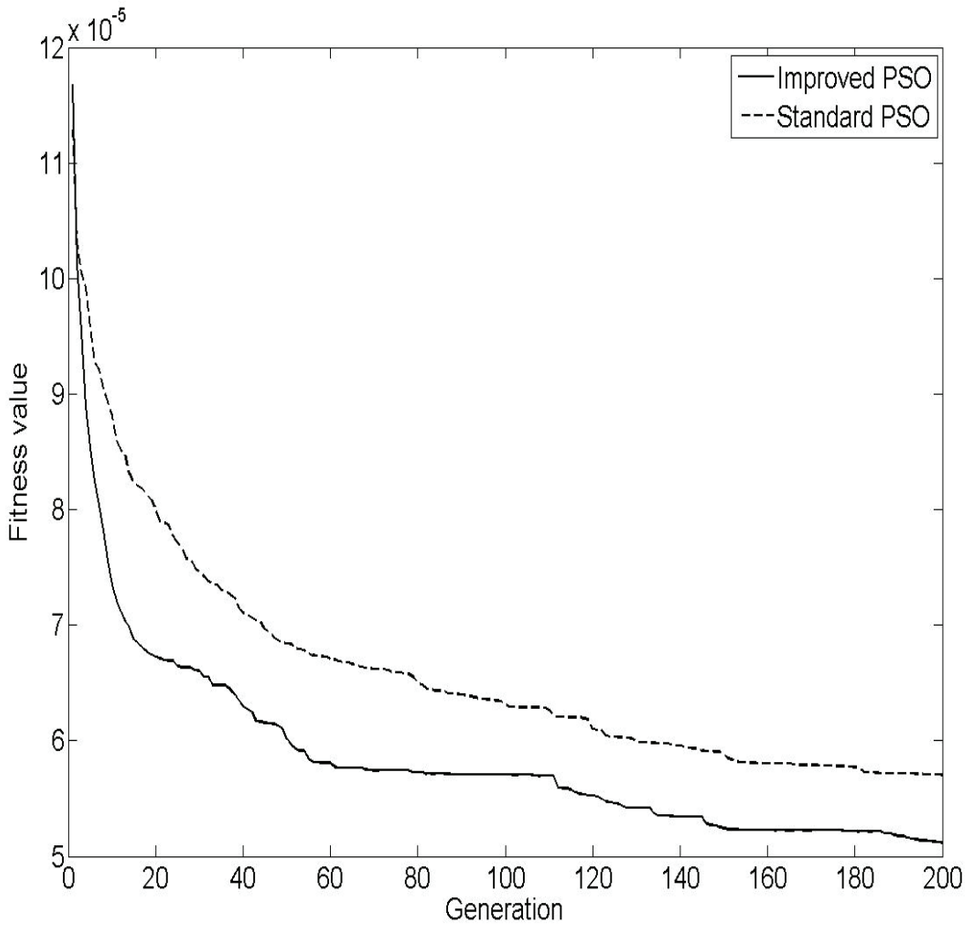


Fig. 8. Fitness convergence comparison between improved PSO and standard PSO for ill-conditioned BSS case.,

## 7. Conclusion

This study addresses on the BSS problem which involves sparse source signals, underdetermined linear mixing model. Some related algorithms have been proposed, but are only tested on toy cases. For robustness, GMM is introduced to learn the distribution of mixtures and find out the unknown mixing vectors; meantime, PSO is used to tune the parameters of GMM for expanding search range and avoiding local solutions as much as possible. Besides, a mechanism is proposed to enhance the evolution of PSO. For simulations, a simple toy case which includes distinguishable mixing matrix and a difficult case which includes close mixing vectors are designed and tested on several state of the art algorithms. Simulation results demonstrate that the proposed PSO-GMM algorithm has the best accuracy and robustness than others. Additionally, the comparison between standard PSO and improved PSO shows that improved PSO is more efficient than standard PSO.

## 8. References


Amari, S.; Chen, T. P. & Cichocki, A. (1997). Stability analysis of learning algorithms for blind source separation, *Neural Networks*, vol. 10, issue: 8, pp. 1345-1351

Belouchrani, A.; Abed-Meraim, K.; Cardoso, J. F. & Moulines, E. (1997). A blind source separation technique using second-order statistics, *IEEE Trans. on Acoustics, Speech, and Signal Processing,*vol. 45, issue: 2, pp. 434-444

Bofill, P. & Zibulevsky, M. (2001). Underdetermined blind source separation using sparse representations, *Signal Processing*, vol. 81, pp. 2353-2362

Cichocki, A. & Unbehauen, R. (1996). Robust neural networks with on-line learning for blind identification and blind separation of sources, *IEEE Trans. on circuits and systems-I: fundamental theory and applications*, vol. 43

Eberhart, R. C. & Kennedy, J. (1995). A new optimizer using particle swarm theory," *Proc. of the Sixth International Symposium on Micro Machine and Human*, pp. 39-43

Grady, P. O. & Pearlmutter, B. (2004). Soft-LOST: EM on a mixture of oriented lines, *Proc. of ICA 2004*, ser. *Lecture Notes in Computer Science Granada*, pp. 430-436

Gudise, V. G. & Venayagamoorthy, G. K. (2003). Comparison of particle swarm optimization and Backpropagation as Training Algorithms for Neural Networks, *Proc. of IEEE Swarm Intelligence Symposium*, pp. 110-117

Hedelin, P. & Skoglund, J. (2000). Vector quantization based on gaussian mixture models, *IEEE Trans. on Speech and Audio Processing,* vol. 8, no. 4, pp. 385-401

Herault, J. & Juten, C. (1986). Space or time adaptive signal processing by neural network models, *Proc. of AIP Conf. Snowbird, UT, in Neural Networks for Computing*, J. S. Denker, Ed. New York: Amer. Inst. Phys., pp. 206-211

Lee, T. W.; Girolami, M. & Sejnowski, T. J. (1999a). Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources, *Neural Computation*, vol. 11, issue: 2, pp. 409-433

Lee,T. W.; Lewicki, M. S.; Girolami, M. & Sejnowski, T. J. (1999b). Blind source separation of more sources than mixtures using overcomplete representations, *Signal Processing Letters*, vol. 6, issue: 4, pp. 87-90

Li, Y. & Wang, J. (2002). Sequential blind extraction of instantaneously mixed sources, *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. 50, issue: 5, pp. 997-1006

Lin, C. & Feng, Q. (2007). The standard particle swarm optimization algorithm convergence analysis and parameter selection, *Proc. of the 3th International Conference on Natural Computation*, pp. 823-826

Liu, C. C.; Sun, T. Y.; Li, K. Y. & Lin, C. L. (2006). Underdetermined blind signal separation using fuzzy cluster on mixture accumulation, *Proc. of the International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 455-458

Liu, C. C.; Sun, T. Y.; Li, K. Y.; Hsieh, S. T. & Tsai, S. J. (2007). Blind sparse source separation using cluster particle swarm optimization technique, *Proc. of International Conference on Artificial Intelligence and Applications*, pp. 549-217

Luengo, D.; Santamaria, I. & Vielve, L. (2005). A general solution to blind inverse problems for sparse input signals: deconvolution, equalization and source separation, *Neurocomputing*, vol. 69, pp. 198-215

Nikseresht, A. & Gelgon, M. (2008). Gossip-based computation of a gaussian mixture model for distributed multimedia indexing, *IEEE Trans. on Multimedia,* vol. 10, no. 3, pp. 385-392

Pajunen, P. (1998). Blind source separation using algorithmic information theory, *Neurocomputing,* vol. 22, issue: 1-3, pp. 35-48

Pham, D. T. & Vrins, F. (2005). Local minima of information-theoretic criteria in blind source separation," *IEEE Signal Processing Letters*, vol. 12, issue: 11, pp. 788-791

Shi, Z.; Tang, H.; Liu, W. & Tang, Y. (2004). Blind source separation of more sources than mixtures using generalized exponential mixture models, *Neurocomputing*, vol. 61, pp. 461-469

Shi, Y. & Eberhart, R. C. (1998). A modified particle swarm optimizer, *Proc. of IEEE World Congress on Computational Intelligence*, pp. 69-73

Song, K.; Ding, M.; Wang, Q. & W. Liu, (2007). Blind source separation in post-nonlinear mixtures using natural gradient descent and particle swarm optimization algorithm, *Proc. of the 4th International Symposium on Neural Networks*, pp. 721-730

Tangdiongga, E.; Calabretta, N.; Sommen, P. C. W. & Dorren, H. J. S. (2001). WDM monitoring technique using adaptive blind signal separation, *IEEE Photonics Technology Letters,* vol. 13, issue: 3, pp. 248 - 250

Todros, K. & Tabrikian, J. (2007). Blind separation of independent sources using gaussian mixture model, *IEEE Trans. on Signal Processing,* vol. 55, no. 7, pp. 3645-3658

Vaerenbergh, S. V. & Santamaria, I. (2006). Spectral clustering approach to underdetermined postnonlinear blind source separation of sparse sources, *IEEE Trans. on Neural Networks,* vol. 17, issue: 3, pp. 811-814

Yilmaz, O. & Rickard, S. (2004). Blind separation of speech mixtures via time-frequency masking, *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. 52, issue: 7, pp. 1830-1847

Yue, Y. & Mao, J. (2002). Blind separation of sources based on genetic algorithm, *Proc. of the 4th World Congress on Intelligent Control and Automation*, pp. 2099-2103

Zhang, Y. C. & Kassam, S. A. (2004). Robust rank-EASI algorithm for blind source separation, *IEE Proceedings-Communications*, vol. 151, issue: 1, pp. 15-19


**Frontiers in Robotics, Automation and Control**

Edited by Alexander Zemliak

ISBN 978-953-7619-17-6

Hard cover, 450 pages

**Publisher** InTech

**Published online** 01, October, 2008

**Published in print edition** October, 2008

This book includes 23 chapters introducing basic research, advanced developments and applications. The book covers topics such us modeling and practical realization of robotic control for different applications, researching of the problems of stability and robustness, automation in algorithm and program developments with application in speech signal processing and linguistic research, system's applied control, computations, and control theory application in mechanics and electronics.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tsung-Ying Sun, Chan-Cheng Liu, Tsung-Ying Tsai, Yu-Peng Jheng and Jyun-Hong Jheng (2008). Robust Underdetermined Algorithm Using Heuristic-Based Gaussian Mixture Model for Blind Source Separation, Frontiers in Robotics, Automation and Control, Alexander Zemliak (Ed.), ISBN: 978-953-7619-17-6, InTech, Available from:
http://www.intechopen.com/books/frontiers_in_robotics_automation_and_control/robust_underdetermined_algorithm_using_heuristic-based_gaussian_mixture_model_for_blind_source_separ

INTECH
open science | open minds

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821