

A Blind Source Separation Algorithm Based on Dynamic Niching Particle Swarm Optimization

Hongjie Li¹, Zhen Li^{1,2} and Hongyi Li^{1,2,*}

¹LMIB, School of Mathematics and Systems Science, Beihang University, Beijing 100191, China

²School of Software Engineering, Beihang University, Beijing 100191, China

*Corresponding author (E-mail: Hongyili_buaa@163.com)

Abstract. In this paper, the dynamic niching particle swarm optimization (DNPSO) is proposed to solve linear blind source separation problem. The key point is to use the DNPSO rather than particle swarm optimization (PSO) and fast-ICA as the optimization algorithm in Independent Component Analysis (ICA). By using DNPSO, which has global superiority, the performance of ICA will be improved in accuracy and convergence rate. The idea of sub-population in DNPSO leads to the greater efficiency compared with other methods when solving high dimensional cost functions in ICA. The performance of ICA based on DNPSO is investigated by numerical experiments.

1 Introduction

Blind source separation (BSS) is an advanced method in signal processing. In recent years, BSS has gained increasing interest in various fields of applications. The problem of linear BSS, in nature, is the problem of separating and estimating waveforms of the observed sources from their linear mixture, without knowing the characteristics of the transmission channels [1,8-10]. And the traditional method of linear BSS is Independent Component Analysis (ICA).

The main objective of ICA is to estimate an un-mixing matrix by using optimization algorithm to achieve the optimal solution. Since the traditional optimization algorithm, including gradient descent algorithm and natural gradient algorithm, are easily trapped into local optimum and have slow convergence rate, researchers proposed Particle Swarm Optimization (PSO) and the its improved algorithm as the optimization algorithm for ICA and have achieved remarkable experimental result [1]. However, the standard PSO algorithm also has the probability that being trapped into local optimum, and its convergence rate turns to be slow at the later time [6]. The Chaos-PSO algorithm proposed by Ding [7] improved the searching efficiency and accuracy of the standard PSO and shows high performance in optimizing low-dimensional functions, however, its accuracy still need to be improved in dealing with high-dimensional functions. [3] All that above demonstrates a requirement of an optimization algorithm with greater global superiority for BSS.

In this paper, we proposed ICA based on Dynamic Niching Particle Swarm Optimization (DNPSO), which has greater global superiority in optimization progress of ICA algorithm. Dynamic Niching Particle Swarm Optimization (DNPSO) maintains the diversity of its particle swarm by applying the idea of niche technique.

The widely using of Niche Technique in Genetic Algorithm (GA) in recent years has interpreted its remarkable performance in global optimization of high-dimensions functions [5]. In DNPSO, the particle swarm is divided into several sub-swarms, which form independent searching spaces dynamically so as to ensure every extreme point being examined. Meanwhile, the free particles are always searching for new optimums, which can help the algorithm avoid being trapped into local optimum [6].

The remainder of this paper is organized as follows. Section 2 presents an overview of the BSS problem based on ICA. Section 3 describes ICA based on standard PSO. Section 4 presents the proposed ICA based on DNPSO. The simulation results will be presented in Section 5.

2 Blind Source Separation (BSS) Problem

2.1 Problem description of BSS

The problem of linear BSS can be modeled as follows:

$$s(t) = (s_1(t), s_2(t), \dots, s_n(t))^T$$

Assuming that there exists n source signals, which are mutually statistically independent and unknown at each time instantly. The BSS mixing model can be presented as

$$x(t) = As(t)$$

where A is an unknown $n * n$ mixing matrix with full rank, $x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$ are the observed signals. The task of BSS is to recover the source vector $s(t)$ by using the observed signals $x(t)$ and possible prior knowledge. The process of separation can be

*Corresponding author (E-mail: Hongyili_buaa@163.com)

formulated that the $n * n$ separating matrix W has the following structure, which is called the un-mixing equation as follows

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t) = \mathbf{W}\mathbf{A}\mathbf{s}(t),$$

where $\mathbf{y}(t) = (y_1(t), y_2(t), \dots, y_n(t))^T$ is the estimation of the source signal $\mathbf{s}(t)$, W is the separating matrix. Without considering the additive noise, the basic thought of BSS is to separate the independent sources from the mixtures by estimating the separating matrix so as to realize Blind Source Separation.

2.2 Independent Component Analysis (ICA) and its optimization algorithm

Independent Component Analysis (ICA) is currently the major method to solve BSS problem. The most widely used algorithm based on ICA is a kind of adaptive optimization algorithm, which find the separation matrix by using optimization algorithm to a determined cost functions after utilizing the measure of independence between separated signals. In the process of using ICA to solve BSS problem, the choosing of cost function, which estimate the similarity between source signals and separation signals, and the optimization algorithm for determining separation matrix will greatly influence the separation effect [1].

The core strategy of improving ICA method lies on the global optimization of the cost function. In traditional ICA, gradient descent algorithm and natural gradient algorithm are used as optimization algorithm, but these methods based on gradient exists the disadvantage of being easily trapped in local optimum[2] In 2007, Vrin, Pham and etc. proved that the cost function of non-gaussian maximum principle has a large amount of local optimums[4]. This demonstrates that when using the optimization algorithm with poor global superiority, the separation precision of ICA method may not be ideal enough as the algorithm based on gradient will be easily trapped into local optimums.

3. ICA based on particle swarm optimization

The Particle Swarm Optimization (PSO) is an optimization algorithm inspired by the process of searching food of bird flock and fish school, especially Swarm Theory. In PSO, each searching individual is regarded as a particle, which represents a potential solution in the D-dimension solution space. All the particles in the solution space combined are regarded as the Particle Swarm.

Assuming that there are m particles in the Particle Swarm, $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ is the D-dimensions position vector of the i -th particle. The superiority of the current position in solution space can be estimated by calculating the fitness of p_i with the cost function

determined. PSO disperses each particle in the whole solution space and adjust its velocity according to its previous velocity and independent cognition (the best solution position that this particle found) as well as the social cognition (the best solution position that the whole particle swarm found), so as to converge the particle swarm on the position of the optimal solution.

The PSO-based ICA algorithm can be implemented by the following iterative procedure [7]:

- 1) Set the cost function $f(\mathbf{x})$
- 2) Preprocess the source signals and randomly initialize particle swarm's positions and velocities.
- 3) The best previous position which giving the best fitness value of the i th particle is updated by the following expression, which also updates the best position among the whole particles:

$$p_i(t+1) = \begin{cases} p_i(t) & \text{if } f(x_i(t+1)) < f(p_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) \geq f(p_i(t)) \end{cases}$$

- 4) Update the position and velocity vector of each particle:

$$\begin{aligned} \mathbf{v}_{is}(t+1) &= \mathbf{v}_{is}(t) + c_1 r_1 (\mathbf{p}_{is}(t) - \mathbf{x}_{is}(t)) + c_2 r_2 (\mathbf{p}_{gs}(t) - \mathbf{x}_{is}(t)) \\ \mathbf{x}_{is}(t+1) &= \mathbf{x}_{is}(t) + \mathbf{v}_{is} \end{aligned}$$

where learning factors c_1 and c_2 are nonnegative real numbers, r_1 and r_2 are independent random numbers subject to the uniform distribution between [0, 1].

- 5) Repeat 2) and 3) until convergence.
- 6) Use the particle w with the best fitness value as the separation matrix and compute the separated signals

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t)$$

In traditional PSO algorithm, each individual particle shares the best position of the whole particle swarm, which leads to the high possibility that the whole particle swarm converge on the optimum found early with slow convergence rate [6]. These limitations will require an efficient and accurate method.

4. ICA based on dynamic niching particle swarm optimization

Dynamic niching Particle Swarm Optimization (DNPSO), proposed by Nickabadi in 2008, was based on the niche technique. [6], and was mainly applied as the optimization algorithm for multi-extreme value problems.

DNPSO uses the concept of sub-populations. In this method, the particles are divided into some sub-swarms and a group of free particles. The sub-swarms are created and destroyed dynamically at each generation and the number of the particles in each swarm changes. The best fitted neighbor is determined as *lbest* and is called the master of the particle. Any particle that is its own best neighbor can be a candidate solution or a niche in the objective function and is referred to as a self-master particle, the particle's master is replaced with the master of its master repeatedly until a self-master particle is reached. And if two masters are too close to each other, the particle with the better fitness value is set as the master of the other particle and form a sub-swarm. All of the particles that are not a member of any sub-swarm are set as free particles [6].

Owing to the using of GCPSO method in sub-swarm searching and the cognition-only model in free particles searching, the sub-swarms will form a niching search area near a promising solution and converge quickly to the optimum point while the free particles can find all of the extreme value points in searching space and form a new sub-swarm around it, which protects the algorithm from being trapped in local optimum. The DNPSO improves the exploration capability of the standard PSO and ensures both the accuracy and efficiency of the results while being applied to high-dimensional functions with multi local optimums.

The most widely used cost function in ICA method such as Non-gaussian maximum principle and mutual information estimation, has been proved to have multiple local optimums in practice using [4]. Considering how the multiple local optimums affects the performance of ICA, which have been discussed above, we use DNPSO in the ICA method for BSS. The ICA based on dynamic niching particle swarm optimization (DNPSO) can be implemented as follows:

1) Set the cost function $f(\mathbf{x})$; processing pretreatment and whitening process to the blind source signals; make the signals zero mean and unit covariance:

$$\mathbf{X} = \mathbf{X} - E[\mathbf{X}], \mathbf{H} = \mathbf{D}^{-\frac{1}{2}} \mathbf{F}^T \mathbf{X},$$

2) Initialize the parameters and maximum iteration times of DNPSO; randomly generate the initial positions z_{id}^0 and velocity v_{id}^0 based on the dimensions of the signal and calculate the fitness value. Each position of the particles indicates a potential solution of separation matrix W needed in ICA.

3) Use the cognition-only model PSO algorithm for free particles group (At the beginning all the particles are free particles), the positions and velocities is updated according to the following equations (p_{id}^k is the best position of the id_{th} particle, w is inertia weight, c is

the learning factor, r is a random parameter between [1,2])

$$v_{id}^{k+1} = w * v_{id}^k + c * r * (p_{id}^k - z_{id}^k)$$

$$z_{id}^{k+1} = z_{id}^k + v_{id}^{k+1}$$

4) For each particle, if it is not self-master, the particle's master is replaced with the master of its master repeatedly until a self-master particle is reached.

5) If two self-masters particles' Euclidian distance is smaller than the threshold value d_{min} , the particle with the better fitness value is set as the master of the other particle and form a sub-swarm.

6) Train the velocity and position of the particles of each sub-swarm according to GCPSO, specifically as the equation follows [6]: (p_{best}^k is the best position of this sub-swarm)

$$v_i^{k+1} = \omega v_{gbest}^k + c_1 * r_1 * (p_{id}^k - z_{id}^k) + c_2 * r_2 * (p_{gbest}^k - z_{id}^k)$$

$$z_i^{k+1} = z_{pbest,i}^k + v_i^{k+1}$$

7) Update the members of each sub-swarm: If the Euclidian distance between any particles in two sub-swarms is smaller than the threshold, combine these two sub-swarms.

8) Repeat 3) to 7) until converge or the maximum iteration times is reached or all the particles have converged.

9) Use the particle w with the best fitness value as the separation matrix and compute the separated signals

5. Simulation Results

ICA based on DNPSO was tested with three speech signals shows in Figure.2, where totally 100000 sampling pixels are set as abscissa and amplitude are set as ordinates.

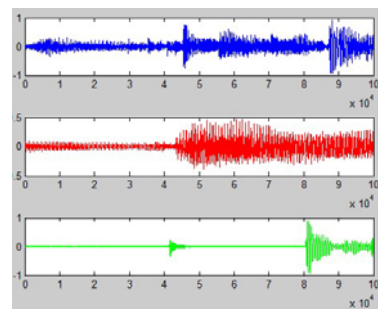


Figure 1. Source signals.

The mixture signals are generated by mixing the source signals with a random 3*3 matrix, which is as the following:

$$\mathbf{H} = \begin{pmatrix} 0.3912 & 0.4423 & 0.4265 \\ 0.2524 & 1.1225 & 0.3527 \\ 0.5223 & 0.4265 & 0.3127 \end{pmatrix}$$

The mixture signals are showed in Figure. 2:

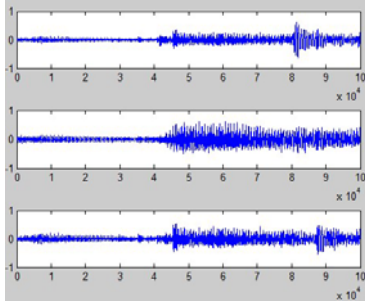


Figure 2. Mixture signals.

In this paper we use the widely-used negative entropy as the cost functions of ICA, its common approximation algorithm is as follows:

$$J(X) = \sum_{i=1}^n (E[f_i(X)] - E[f_i(G)])^2$$

where G is the random Gaussian vectors with zero mean and unit covariance. And f is a none-quadratic function which can be chosen according to the requirement, $f(x) = \frac{1}{a} \ln(\cosh(ax))$ is used in our simulation where $a = 1.5$.

The parameters of DNPSO are set as follows. The number of particles $n = 150$. The maximum number of sub-swarms $M = 30$, learning rate $c = 1.5$, The maximum number of particles in a sub-swarm $MinPts = 5$, $w_{max} = 0.9$, $w_{min} = 0.1$, maximum iterate times $G_{max} = 700$.

To estimate the results of ICA based on DNPSO, we performed the blind source separation with the proposed algorithm along with two other algorithms, including the well-known FastICA and ICA based on standard PSO. Their separation results are showed in Figure. 3 to Figure. 5.

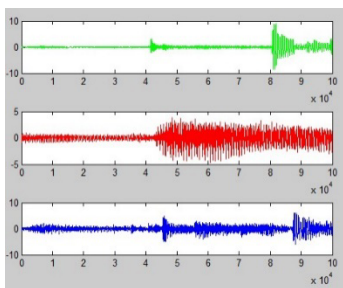


Figure 3. Separated signals using FastICA.

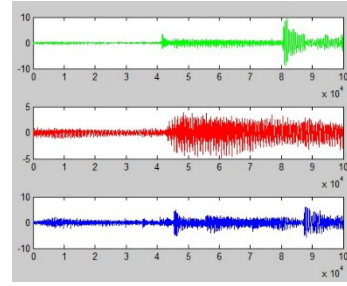


Figure 4. Separated signals using PSO

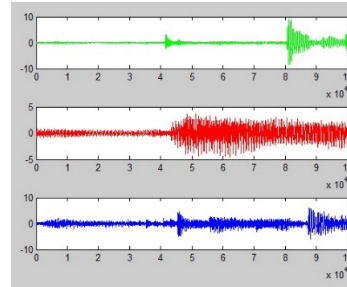


Figure 5. Separated signals using the proposed method.

We used measurement index of similarities between signals to estimate the results quantitatively (y indicates the separation signals while s indicates the source signals):

$$\Phi_{ij} = \Phi(y_i, s_j) = \frac{\left| \sum_{t=1}^T y_i(t) s_j(t) \right|}{\sqrt{\sum_{t=1}^T y_i^2(t) * \sum_{t=1}^T s_j^2(t)}}$$

The results of the three used algorithm is showed in the following tables:

Table 1. Performance of three algorithm in simulation

Methods	Proposed	PSO-ICA	Fast-ICA
Similarity of signal 1	0.9989	0.9670	0.9840
The similarity of signal 2	0.9892	0.9368	0.9897
The similarity of signal 3	0.9879	0.9644	0.9743
Time cost (in second)	4.321	9.917	0.553

The performance results show that ICA based on DNPSO outperforms both FastICA and ICA based on standard PSO in separation precision. As to the coverage rate, ICA based on DNPSO also shows remarkable superiority over ICA based on standard PSO.

Owing to its fast converge rate, ICA based on DNPSO requires fewer iteration times and has great superiority in stability. In all of the methods except the proposed algorithm, the exploration will probably focuses on the optimum found earlier and been trapped in the nearby domain. But in ICA based on DNPSO, the free particles are always search for better solutions. In general, ICA based on DNPSO solves the problem of low separation accuracy in blind source separation with an ideal processing rate.

6. Conclusion

In this paper, an ICA method based on Dynamic Niching Particle Swarm Optimization (DNPSO) is proposed for blind source separation in view of the local optimum and slow convergence rate difficulty. The DNPSO is for the first time in this paper to be used in ICA method. The division of main swarm in DNPSO not only help guarantee the global optimum to be found in searching domain, but also improves the separation precision of blind source separation remarkably. According to the simulation results which compare the proposed method with Fast-ICA and PSO-ICA, the proposed method significantly improves BSS's performance and gives better results.

As the parameter setting of ICA base on DNPSO still has much space to be optimized, and the additive noises is not concerned in this paper. Our future work will focus on the optimization of parameter as well as taking the additive noises into consideration, and to further deal with the nonlinear BSS problem.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No.61379001) and the Key Project of Beihang University (No. 201412) ("Exploration and Practice of Innovation Ability Cultivation in Mathematical Education").

References

1. Y. Gao, S. Xie. A blind source separation algorithm using particle swarm optimization. In Proceedings of the Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, **1**, 297-300 (2004)
2. T. Kurihara, K. Jinno. Analysis of convergence property of PSO and its application to nonlinear blind source separation. In Proceedings of IEEE Congress on Evolutionary Computation, 976-981 (2013)
3. C. L. Lin, S. T. Hsieh, T. Y. Sun. PSO-based learning rate adjustment for blind source separation. In Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems, 181-184 (2005)
4. F. Vrins, D. T. Pham, M. Verleysen. Mixing and non-mixing local minima of the entropy contrast for blind source separation. IEEE Transactions on Information Theory, **3(3)**, 1030-1042 (2007)
5. R. Brits, A. P. Engelbrech, F. V. Bergh. A niching particle swarm optimizer. In Proceedings of Asia-Pacific Conference on Simulated Evolution and Learning, **2**, 692-696 (2002)
6. A. Nickabadi, M. M. Ebadzadeh, R. Safabakhsh. DNPSO: A Dynamic Niching Particle Swarm Optimizer for Multi-modal Optimization. In Proceedings of IEEE Congress on Evolutionary Computation, 26-32 (2008)
7. W. Ding. A New Method for Image Noise Removal using Chaos-PSO and Nonlinear ICA. Procedia Engineering, **4**, 111-115 (2011)
8. H. Li, C. Wang, D. Zhao. An Improved EMD and Its Applications to Find the Basis Functions of EMI Signals. Mathematical Problems in Engineering, (2015)
9. H. Li, Z. Song, D. Zhao, P. Wang, J. Chen. A Single Channel EMI Signal Separation Method Based on Directly-mean Empirical Mode Decomposition. Journal of Information & Computation Science, **12(17)**, 6333-6340 (2015)
10. D. Zhao, H. Li, On the computation of inverses and determinants of a kind of special matrices. Applied Mathematics and Computation, **250**, 721-726 (2015)