# Maximum Likelihood Independent Component Analysis using GA and PSO

H. Azad, M. Hatam

*Abstract*— **In this paper, the problem of independent component analysis based on maximum likelihood criteria has been considered. This approach involves inversion of matrix in each iteration (which is computationally complex). Therefore, genetic algorithm and particle swarm optimization have been proposed to be used for solving maximum likelihood ICA problem. Results, given in MSE of estimating signals with respect to source signals, show good performance of the proposed algorithms. Also a comparison to traditional FastICA method has been presented, which shows acceptable performance of the algorithms.**

***Keywords: Independent Component Analysis, Maximum Likelihood, Genetic Algorithm (GA), Particle Swarm Optimization (PSO)***

## I. Introduction

INDEPENDENT Component Analysis (ICA) is one of main statistical signal processing methods for separating desired signals from their mixtures. The basic assumption in ICA problem is that the source signals are assumed to be statistically independent. Without any knowledge of the nature of the main signals (and even how they mixed) except their independency, ICA algorithms try to use different criteria to estimate the independent components. ICA has got many different applications new years, such as blind separation of voice and image signals, feature extraction, data communication and array signal processing [1].

Different ICA algorithms include an optimization problem in their structure. Genetic Algorithm (GA) has been previously used with different ICA algorithms to solve these optimization problems. In [2], the sum of the absolute values of kurtosis as a criterion for separating independent components is used as the fitness function that should be maximized by GA. In [3], a new ICA algorithm using GA for the case of nonlinear mixture of source signals is proposed. Also in [4], different criteria including mutual information and negentropy, are considered and GA is used to optimize the desired objective functions in these criteria. Finally in [5], combined kurtosis and mutual information criteria have been considered and solved by GA.

In this paper, we consider the maximum likelihood approach of separating the source signals [1]. Conventional methods of solving ICA problem in this approach are based on stochastic gradient ascent of some functions. These iterative methods involve a matrix inversion in each iteration which makes them time-consuming. Also they are not efficient algorithms because of the computational complexity of inverting a matrix in each step [1]. In this work, we propose a new algorithm based on GA, the main advantage of which is that there is no need to invert any matrix. The proposed algorithm shows good performance in estimating the independent source signals.

Also, Particle Swarm Optimization (PSO) is used to solve this problem as another evolutionary approach. Note that the problem formulation and the approach of solving it, is almost similar for both evolutionary algorithms (GA and PSO). Similar to GA, PSO has been also used in some ICA algorithms. In [6], the mutual information criteria for ICA has been considered and solved by PSO. But the ML criterion for ICA problem has never been solved by PSO.

This paper is organized as follows. At first we present the ICA problem and ML approach for solving it. The proposed algorithm based on GA is considered next. Also, as a different approach, the proposed algorithm based on PSO is described. Then the computer simulations are given. Finally the conclusions are presented.

## II. Maximum likelihood ICA

The ICA problem in its common form is formulated as follows. Assume that there exist N independent sources, $s_i, i = 1,2,..., N,$ we have N observed signals $x_i, i = 1,2,..., N$ which are assumed to be linearly mixtures of the source signals with an unknown matrix $\mathbf{A}_{N\times N}$ ; i.e. we have:

$$\mathbf{x} = \mathbf{As} \tag{1}$$

where $\mathbf{x} = [x_1,..., x_N]^T$ and $\mathbf{s} = [s_1,..., s_N]^T$ are observed and source vectors respectively.

The ICA problem in its basic form is to find a matrix $\mathbf{B}_{N\times N}$ in order to estimate the source vector $\hat{\mathbf{s}}$ using the observed signals:

$$\hat{\mathbf{s}} = \mathbf{Bx} \tag{2}$$

There exist different criteria for solving this problem, some of which are nongaussianity, mutual information, negentropy and maximum likelihood estimation [1]. In this

H. Azad is with the Electrical Engineering Department, Azad University, Shiraz Branch, Shiraz, Iran (e-mail: azad@shirazu.ac.ir).

M. Hatam is with the Electrical Engineering Department, University of Shiraz, Shiraz, Iran.

paper we consider the maximum likelihood approach. Using (2), the density function of the observed signal can be formulated as:

$$p_x(\mathbf{x}) = \left|\det \mathbf{B}\right| p_s(\mathbf{s}) = \left|\det \mathbf{B}\right| \prod_i p_i(s_i) \tag{3}$$

After some mathematical manipulations, the log-likelihood function can be written as [1]:

$$\begin{aligned} \log L(\mathbf{B}) = \sum_{t=1}^{T} \sum_{i=1}^{N} \log p_i(\mathbf{b}_i^T \mathbf{x}(t)) \\ + T \log \left|\det \mathbf{B}\right| \end{aligned} \tag{4}$$

In this equation, it is assumed that we have T observation of $\mathbf{x}$, denoted by $\mathbf{x}(1)$, $\mathbf{x}(2)$, …, $\mathbf{x}(T)$. Also $\mathbf{b}_i$s are different rows of matrix $\mathbf{B}$; i.e. $\mathbf{B} = (\mathbf{b}_1,...,\mathbf{b}_N)^T$.

In ML approach, (4) should be maximized with respect to matrix $\mathbf{B}$. But as it can be seen, it is a function of densities of the independent components which are assumed to be unknown primarily. To overcome this problem, these densities are assumed to be one of two basic densities called superguassian and subguassian, the logarithmic forms of which are:

$$\begin{aligned} \log \tilde{p}_i^+(s) &= 1 - 2\log\cosh(s) \\ \log \tilde{p}_i^-(s) &= 1 - \left[\frac{s^2}{2} - \log\cosh(s)\right] \end{aligned} \tag{5}$$

where $\tilde{p}_i^+(s)$ and $\tilde{p}_i^-(s)$ are super- and subguassian densities respectively. There is a test that can be used to choose one of these two choices for a variable $s$. For this purpose, one can compute the following equation:

$$\begin{aligned} E\left\{-\tanh(s_i)s_i + (1-\tanh(s_i)^2)\right\} > 0 \Rightarrow \tilde{p}_i^+ \\ E\left\{-\tanh(s_i)s_i + (1-\tanh(s_i)^2)\right\} < 0 \Rightarrow \tilde{p}_i^- \end{aligned} \tag{6}$$

Note that expectation is an average computed from T samples of source signals (eq. 4). But to compute this test, we should use samples of source signals which are unknown and the main goal in ICA problem is to estimate these values. So, some iterative methods are presented in literature, which estimate the unknown values (matrix $\mathbf{B}$ and also source signals) iteratively [1]. Starting with a random generated matrix $\mathbf{B}$, these algorithms compute a rough estimate of source signals using (2) and then use this estimate to find the form of densities by (6). The estimated forms of densities (i.e. one of two forms in (5)) are then used in (4) to compute the value of log-likelihood function. These steps are repeated until the condition of convergence is satisfied. Finally, values of source signals can be estimated by (2) using final value of $\mathbf{B}$. Also FastICA approach is used to solve this problem [1].

In this paper, new approaches using GA and PSO are proposed for solving this problem.

## III. Maximum likelihood ICA based on GA

The GA works on the Darwinian principle of natural selection called "survival of the fittest". The main attraction of GAs is that they are the global optimization procedure and require only the numerical values of the objective function and constraints to direct the search [7].

The algorithm starts with a randomly generated set of solutions called population. Each individual or chromosome of the population can be a potential solution to the optimization problem. The chromosomes evolve through different iterations of GA. In each iteration, the value of the desired objective function called fitness is evaluated for each chromosome. Fitter individuals with better values of the desired fitness function, have higher probability to survive to next iteration. The two main operator of GA, crossover and mutation, are applied to these individuals resulting new chromosomes which can be better solutions for the optimization problem. Different termination criteria such as number of iterations can be used to stop the genetic algorithm. Finally the fittest chromosome is considered as the solution of the problem.

As it can be seen, we encounter with the maximization of (4) to find the ML estimation of matrix $\mathbf{B}$. So, we decide to use genetic algorithm as a procedure to solve this optimization problem. The main advantage of the proposed algorithm is that it is not required to compute inversion of matrices as for conventional methods.

A pseudo code for the proposed algorithm is as follows.

1. *Encoding:* The parameters to be optimized should be encoded into genes and chromosomes. Here the elements of matrix $\mathbf{B}$ are the desired values to be optimized. We use a real-coded GA to encode the chromosomes. A simple example of such an encoding is as below:

$$\mathbf{B} = \begin{bmatrix} 4 & 5 \\ 6 & 3 \end{bmatrix} \Rightarrow chromosome = \begin{bmatrix} 4 & 5 & 6 & 3 \end{bmatrix}$$

Fig. 1: Example of encoding a chromosome

2. *Initial population generation:* Initial population is generated randomly. The population consists of $N_{pop}$ chromosomes or individuals.

3. *Density form estimation:* Each chromosome can be decoded as a matrix $\mathbf{B}$ that can be used to compute a rough estimate of source signals using (2). Related to each estimated signal, the forms of

densities of source signals can be estimated using (6).

4. *Fitness evaluation:* Using the estimated forms of densities, the fitness value corresponds to each chromosome can be computed by (4).

5. *Selection:* Some individuals should be selected to be used for mating. Here, we use the traditional roulette wheel method.

6. *Crossover:* Simple one point and two point crossover is used in this paper. The crossover rate is set to have different values and the one which has the best results is used to compare with other algorithms.

7. *Mutation:* Gaussian mutation is used in this paper. The mutation function adds a random number taken from a Gaussian distribution with mean 0 to each entry of the parent vector. The variance of this distribution is determined by the parameters scale and shrink. The scale parameter determines the variance of Gaussian distribution at the first generation. The shrink parameter controls how the variance shrinks as generations go by. The variance at the kth generation, is given by the following recursive formula:

$$\mathrm{var}_k = \mathrm{var}_{k-1}(1- \quad \times \frac{k}{N_{Gen}}) \qquad (7)$$

where and $N_{Gen}$ are shrink parameter and total number of generations of genetic algorithm [8].

8. *Termination:* If the number of generation equals to $N_{Gen}$, stop the algorithm, else go to 3.

Finally, the best chromosome is used as the best estimate of matrix $\mathbf{B}$ and as a result, the estimate of source signals can be computed using (2).

## IV. MAXIMUM LIKELIHOOD ICA BASED ON PSO

Particle Swarm Optimization is developed out of attempts to model bird flocks and fish schools [7]. Initializing the particles with random positions and random velocities, the PSO algorithm tries to move the whole population through the best position which is the solution of the problem. In each iteration the algorithm updates the value of velocities and positions of the particles using the values of previous iteration, the best position of all particles in previous iterations and also the best position of the particle itself in previous iterations. The position of particles can be considered as chromosomes of GA algorithm; i.e. they contain the values of probable solutions to the problem. Each particle is evaluated by the fitness function to be optimized, the value of which considers the best particle.

Using an almost similar procedure to previous proposed algorithm base on GA, we can solve our problem using PSO. A pseudo code for the PSO based ML-ICA is as follows:

1. *Encoding:* The parameters to be optimized should be encoded as positions of particles. Similar to previous section, the elements of matrix $\mathbf{B}$ are the desired values to be optimized. The encoding procedure is as Fig. 1, except that we should call the encoded vector as position of the particle, which is shown by $\mathbf{x}_i(k)$ (the position of *i*th particle in *k*th iteration.

2. *Initial population generation:* Initial positions and velocities of the particles are generated randomly. The population consists of $N_{pop}$ particles.

3. *Density form estimation:* The position of each particle can be decoded as a matrix $\mathbf{B}$ that can be used to compute a rough estimate of source signals using (2). Related to each estimated signal, the forms of densities of source signals can be estimated using (6).

4. *Fitness evaluation:* Using the estimated forms of densities, the fitness value corresponds to each particle can be computed by (4).

5. *Updating:* The positions and velocities of the particles are updated using following equations [7]:

$$\mathbf{v}_i(k+1) = \text{¤}(k)\mathbf{v}_i(k) + \text{\textasciitilde}_1\left[\text{¥}_{1i}\left(\mathbf{p}_i - \mathbf{x}_i(k)\right)\right] + \text{\textasciitilde}_2\left[\text{¥}_{2i}\left(\mathbf{G} - \mathbf{x}_i(k)\right)\right] \qquad (7)$$

$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + \mathbf{v}_i(k+1)$$

where:
*i*: paticle index
*k*: discrete time index
$\mathbf{v}_i$: velocity of *i*th particle
$\mathbf{x}_i$: position of *i*th particle
$\mathbf{p}_i$: best position found by best particle
$\mathbf{G}$: best position found by swarm (global best)
$\text{¥}_{1i}, \text{¥}_{2i}$: random numbers on the interval [0,1] applied to *i*the particle
¤ : a decreasing linear inertia function
$\text{\textasciitilde}_1, \text{\textasciitilde}_2$ : acceleration constants

6. *Termination:* If the number of generation equals

to $N_{Gen}$, stop the algorithm, else go to 3.

Finally, the best position of particles (i.e. **G**) is used as the best estimate of matrix **B** and as a result, the estimate of source signals can be computed using (2).

## V. Simulation

In this part the results of computer simulations are presented. Similar to [2-3], we use sinusoidal wave and a uniform random signal as sources.

$$s(t)=\begin{bmatrix}\sin(50t+5)\\ rand(t)\end{bmatrix} \tag{8}$$

These signals are mixed using a 2x2 matrix **A** :

$$\mathbf{A}=\begin{bmatrix}0.5 & 0.46\\ 0.8 & -0.6\end{bmatrix} \tag{8}$$

For the comparison of different algorithms, we use mean square error (MSE) between source signals and the estimated signals resulted from algorithms. MSE of error is calculated using 500 independent trials for all following simulations.

First of all, consider the proposed algorithm based on GA. The settings for genetic algorithm are given in table 1. Note that we try different parameters for genetic algorithm and finally the best setting (i.e. the setting with minimum MSE) is used to compare to other approaches.

TABLE 1
SETTINGS OF GENETIC ALGORITHM

| GA Parameter | Value |
|---|---|
| Selection operator | roulette wheel |
| Crossover operator | one & two point crossover |
| Crossover rate | 0.7, 0.75, 0.8, 0.85, 0.9 |
| Mutation operator | Gaussian mutation |
| Mutation scale | 0.5 |
| Mutation shrink | 0.75 |
| Population size ( $N_{pop}$ ) | 20, 10 |
| Number of Generations ( $N_{Gen}$ ) | 50 |

The results in the form of MSE are plotted in Fig. 2 and Fig.3 for sinusoidal and rand sources, respectively. As it can be seen in both figures, the greater the cross over rate is, the larger the MSE becomes. It may because of when the cross over rate increases, although it is better for global search but the convergence property of the algorithm degrades. Also as is expected, the MSE decreases as the population size ( $N_{pop}$ ) increases. It is because by increasing the number of chromosomes ( $N_{pop}$ ), the algorithm will have the opportunity to search through more probable solutions. As it can be seen, the crossover operator doesn't play an important role in the performance of the algorithm. Although the performance improves for two point crossover with respect to one point one. We can justify the phenomena by the fact that using two point crossover, the property of global searching improves, which results in better solutions.

For another comparison, we force $N_{Gen}$ to be 30 and compare the results to that of $N_{Gen}=50$ for sinusoidal source. The results are plotted in Fig. 4. As it is expected, increasing the number of generations, improves the performance of the algorithm.

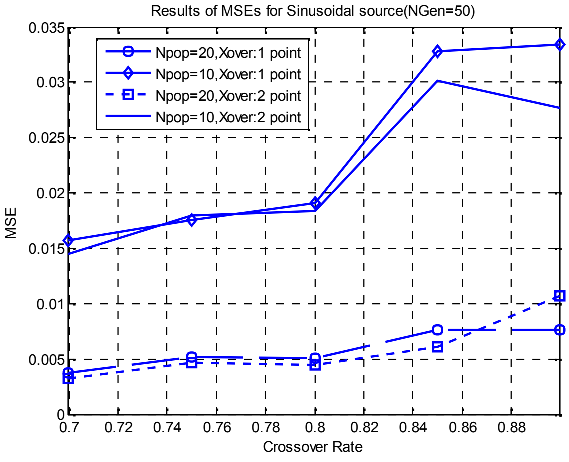

Fig2. MSE for sinusoidal source

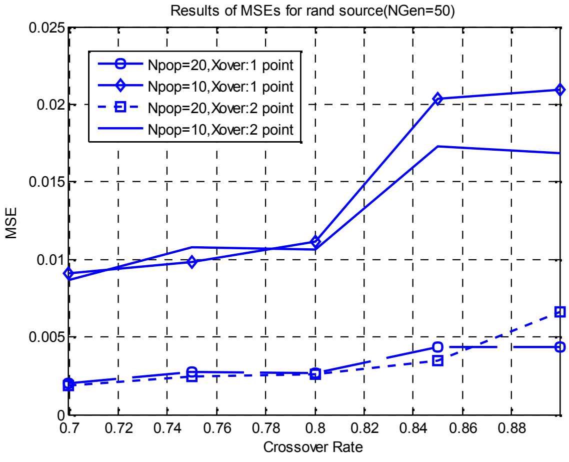

Fig3. MSE for rand source

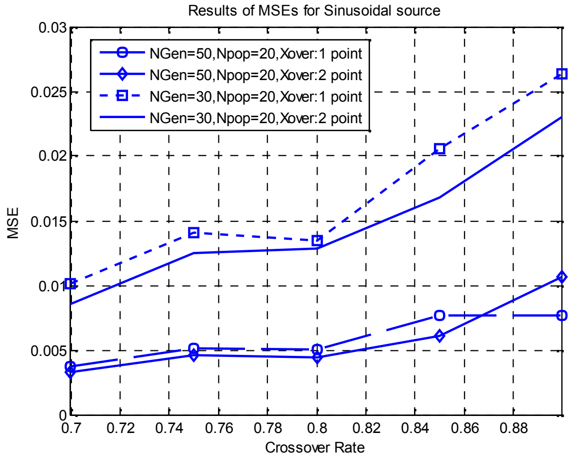

Fig4. Performance comparison for sinusoidal source for two values of $N_{Gen}$

So, using the results concluded from the plotted figures, for upcoming comparisons, we select crossover rate, $N_{pop}$ and $N_{Gen}$ to be 0.7, 20 and 50, respectively. Also two point crossover is selected.

As an example, we plot the results for the configuration of GA algorithm. The source signals are shown in Fig. 5. Also the observed and separated signals are illustrated in Fig. 6 and 7, respectively. As it can be seen, the proposed algorithm based on GA shows good performance in separating the signals.

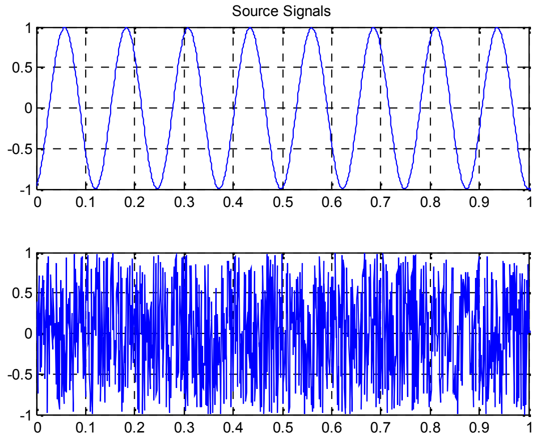

Fig. 5: Source signals for GA based algorithm

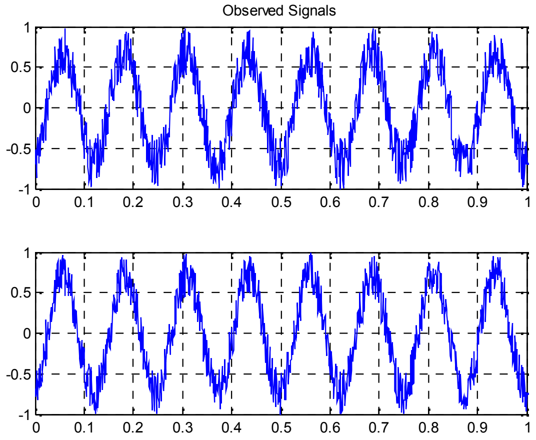

Fig. 6: Observed signals for GA based algorithm

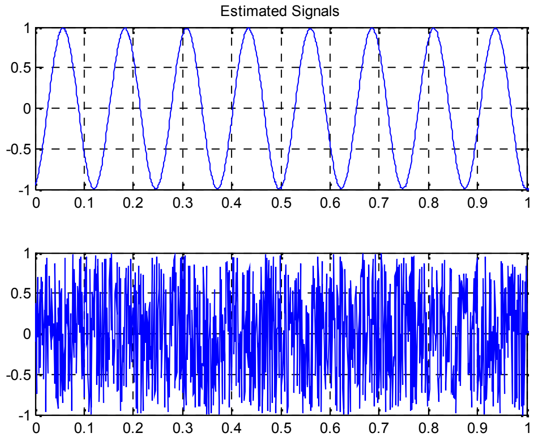

Fig. 7: Separated signals for GA based algorithm

The next proposed algorithm is based on PSO. The settings for PSO algorithm is given in table 2.

TABLE 2

SETTINGS OF PARTICLE SWARM OPTIMIZATION

| PSO Parameter | Value |
|---|---|
| Number of Particles ( $N_{pop}$ ) | 10 |
| acceleration constant 1 ( $\sim_1$ ) | 2.1 |
| acceleration constant 2 ( $\sim_2$ ) | 2.1 |
| Number of Iterations ( $N_{It}$ ) | 200 |

As an example of using this algorithm, see Fig. 8 to 10. The results show good performance for the proposed algorithm.

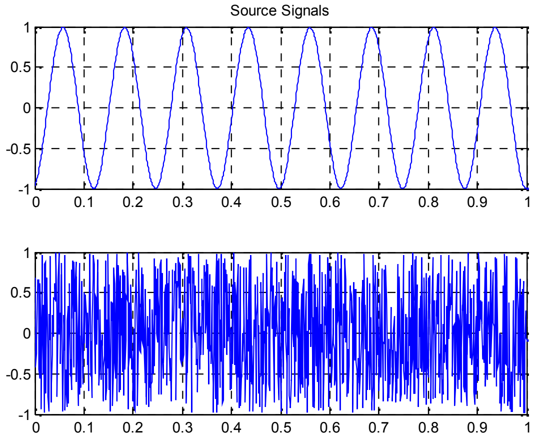

Fig. 8: Source signals for PSO based algorithm

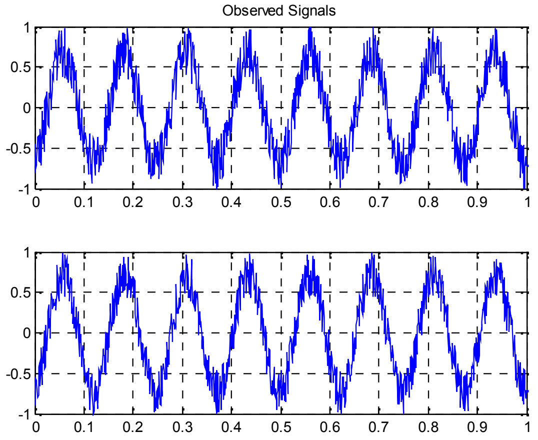

Fig. 9: Observed signals for PSO based algorithm

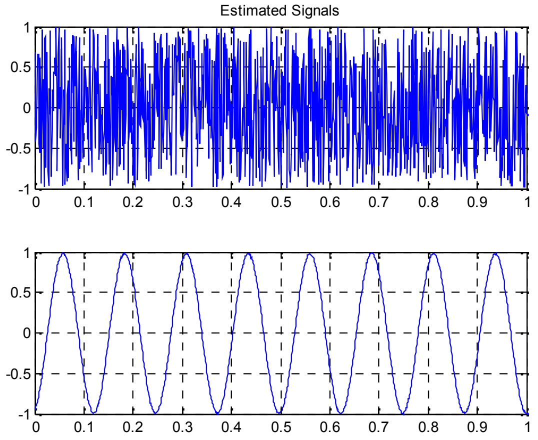

Fig. 10: Separated signals for PSO based algorithm

Finally, the MSE of the proposed algorithms are compared to FastICA [1] in table 3. The results of proposed algorithms are acceptable to that of FastICA.

TABLE 3
COMPARISON OF MSE OF DIFFERENT ALGORTIHMS

| | sinusoidal source | rand source |
|---|---|---|
| GA-ML ICA | 3.2697e-3 | 1.8588e-3 |
| PSO-ML ICA | 1.6346e-3 | 9.6123e-4 |
| FastICA | 8.2819e-4 | 4.2375e-4 |

## VI. CONCLUSION

Two algorithms based on genetic algorithm and particle swarm optimization are proposed to be used for solving maximum likelihood ICA problem. Results, given in MSE of estimating signals with respect to source signals, show good performance of the proposed algorithms. Also a comparison to traditional FastICA method was given, which shows acceptable performance of the algorithms.

## REFERENCES

[1] A. Hyvarinen, J. Karhunen, and E. Oja., *Independent Component Analysis*, John Wiley & Sons, Inc., 2001.

[2] X. Y. Zeng, Y. W. Chen, Z. Nakao, and K. Yamashita, “Signal separation by independent component analysis based on a genetic algorithm,” *in Proc. of 5th International Conference on Signal Processin, WCCC-ICSP 2000*, Aug. 21-25, 2000, pp. 1688-1694.

[3] F. Rojas, C.G. Puntonet, I. Rojas, J. Ortega, and A. Prieto, “Genetic Algorithm Approach to Nonlinear Blind Source Separation”, *in Proc. of the 2002 IEEE World Congress on Evolutionary Computation, CEC'2002*, Vol. 2, May 12-17, 2002, pp. 1098 -1102.

[4] D. P. Acharya, G. Panda, Y. V. S. Lakshmi, “Constrained genetic algorithm based independent componenet analysis,” *in Proc. of the 2007 IEEE World Congress on Evolutionary Computation, CEC'2007*, Sept. 25-28, 2007, pp. 2443-2449.

[5] Cristina P. Dadula and Elmer P. Dadios, “A genetic algorithm for blind source separation based on independent component analysis,” *in Proc. of the 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Nov. 12-16, 2014, pp. 1-6.

[6] FuZhong Nian, Weijuan Li, Xiangfeng Sun, Ming Li, “An Improved Particle Swarm Optimization Application to Independent Component Analysis,” *in Proc. of the 2009 ICIECS 2009. International Conference on Information Engineering and Computer Science*, Dec. 19-20, 2009, pp. 1-4.

[7] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.

[8] *The Genetic Algorithm and Direct Search Toolbox*, MATLAB R2006a, copyright 1984-2006, The Mathworks, Inc.

[9] Particle Swarm Optimization Toolbox, B. Birge. (2006, March, 20). Available: http://www.mathworks.com/matlabcentral/fileexchange/7506-particle-swarm-optimization-toolbox