Study of Differential Evolutionary Algorithm in Blind Source Separation

Monorama Swain
Dept. of ETC
Silicon Institute of Technology
Bhubaneswar, India

Rutuparna Panda
Dept. of ETC
VSSUT
Burla, India

Sneha Tibrewal
Dept. of ETC
Silicon Institute of Technology
Bhubaneswar, India

ABSTRACT
Blind source separation is a well known problem that arises in a large number of signal processing applications. In this paper we proposed a novel Evolutionary algorithm for Blind source separation of Instantaneous mixtures for optimization of continuous time domain signals. Among various evolutionary optimization principles, a population-based real-parameter optimization technique based on differences among population members is getting popular in various real-life optimization problems. This paper addresses this so-called Differential Evolution strategy and shows some sample cases where it can be utilized to separate a number of source signals using a particular channel.

Keywords
Blind source separation, Instantaneous mixture, Differential evolution

1. INTRODUCTION
One of the most challenging problem in signal processing is the separation of independent sources from observed outputs. In many practical situations, one or more desired signals need to be recovered from the mixtures. It is an approach to estimate the source signals by using only the information of mixed signals observed at each input channel [5, 8]. The estimation is performed blindly i.e. without possessing information such as its location and active time. In several situations it is desirable to recover all sources from the recorded mixtures, or at least to segregate a particular source. This method can be further utilized to extract information about the physical mixing system.

Generally two types of mixing models are defined instantaneous and convolutive models. The instantaneous mixing model as shown in Figure1.1 exhibits each mixture as a linear combination of source signals. [6, 14].

In this case, the observed mixtures can be written as

$$x(n) = As(n) + n(n) \dots\dots\dots\dots\dots\dots\dots (1)$$

Where A is the mixing matrix and n(n) is additive noise.

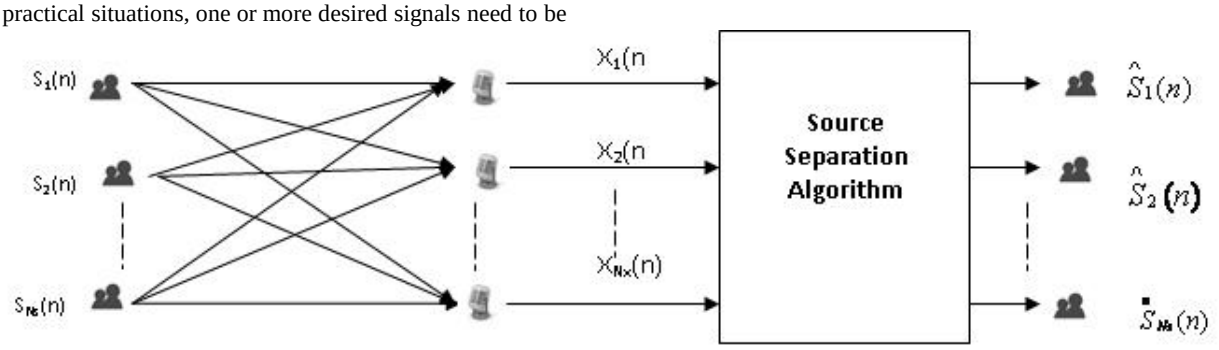

Fig 1.1: Block Diagram of Instantaneous BSS model

The additive noise is still a very difficult problem to solve. Generally the number of microphones never exceeds the number of sources so it will be assumed throughout this paper that the additive noise is negligible, i.e. the SNR is high. It will also be assumed that the number of sources equals the number of microphones, i.e., Ns = Nx [4, 15]. In this instantaneous mixture case, the goal of blind source separation is to estimate an unmixing matrix B such that:

$$B \approx A^{-1} \dots\dots\dots\dots\dots\dots\dots\dots (2)$$

In other words, blind source separation attempts to find a matrix B such that

$$\hat{s} = Bx(n) \dots\dots\dots\dots\dots\dots\dots (3)$$

Where $\hat{s}(n)$ is an estimate of the original source signals $s(n)$.

EVOLUTIONARY ALGORITHMs (EAs), inspired by the natural evolution of species, have been successfully applied to solve numerous optimization problems in diverse fields. The basic definition of biological evolution says: "Evolution is a process that results in heritable changes in a population spread over many generations". In this paper we have presented Differential Evolutionary algorithm for the Blind source separation problem. The differential evolution (DE) algorithm, proposed by Storm and Price, is a simple yet powerful population-based stochastic search technique, which is an efficient and effective global optimizer in the continuous search domain[9,13]. The diversified application of DE includes many fields of engineering such as communication and mechanical engineering. In DE, there exists much trial vector generation strategies out of which a few may be suitable for solving a particular problem. Moreover, three crucial control parameters involved in DE, i.e., population size, scaling factor, and crossover rate, may significantly influence the optimization performance of the DE. Hence, solving a specific optimization problem requires time consuming trial and error search so as to find out the most appropriate strategy in order to adjust the parameter values [1, 12]. However, such a trial-and-error searching process requires high computational costs. Moreover, as evolution proceeds, the population of DE may move through different regions in the search space, wherein the associated parameter settings for certain strategies may be more efficient than the others. Therefore, it is desirable to adaptively determine an appropriate strategy and its associated parameter values at different stages of evolution/search process.

The rest of the paper is organized as follows. Section 2 presents the problem statement, Section 3 provides a brief literature overview of the DE algorithm, section 4 describes the experiments and simulation strategies, results have been presented. Finally conclusions and future research directions are provided in section 5.

2. PROBLEM FORMULATION

Due to the powerful ability of the DE algorithm to optimize data values it can be used to solve the BSS problem. The original source signal which gets distorted in a channel is denoted by X. The channel mixing is denoted by matrix A and the resultant signal is denoted by Y [3, 11]. In this method we provide X and Y to estimate the mixing matrix called as A_estimated which will be used to design the unmixing system to estimate the original source signal as shown in Figure.1.2.

While designing the system we passed a large number of sample patterns through the mixing system to find the distorted patterns in order to estimate the mixing matrix [7]. For the sake of simulation we have taken the mixing matrix as random. After estimating the matrix we found out its inverse which gave us the unmixing system as shown in Figure.1.3.

Now any number of signals which are distorted by the particular channel can be processed to find the original source signal [10].

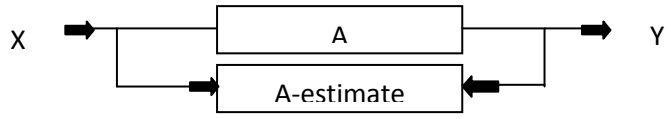

Fig.1.2.Block Diagram of mixing matrix optimization

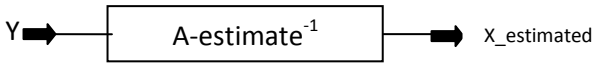

Fig.1.3.Block Diagram of unmixing system

3. DIFFERENTIAL EVOLUTION

Differential evolution is an efficient optimization technique developed by Storn and Price [9] which is oriented to optimize real values. It is a new heuristic and efficient stochastic direct search approach like evolutionary algorithm. The beauty of this algorithm lies in its simple and compact structure. Like any other evolutionary algorithm, DE also starts with a population of NP D dimensional parameter vectors. DE is a novel parallel direct search method which uses M parameter vectors i=0, 1, 2….M-as a population for each generation G. M doesn't change during the minimization process. DE is also similar to genetic algorithm using the similar operators like crossover, mutation and selection. The most important point in DE strategy is that it relies on mutation operation [1, 2]. The main operation is based on the differences of randomly sampled pairs of solutions in the population. It is desirable to adaptively determine an appropriate strategy and its associated parameter values at different stages of evolution/search process so as to avoid spreading of search space.
The main steps in DE are:

1. Initialization
2. Evaluation
3. Repeat
   i. Mutation
   ii. Recombination
   iii. Evaluation
   iv. Selection

Until (termination criteria are met) as shown in Figure.1.4.

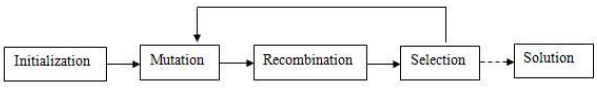

Fig.1.4. General Evolutionary Algorithm scheme

3.1 DE Algorithm Description

DE algorithm aims at evolving a population of NP D-dimensional parameter vectors, so-called individuals, which encode the candidate solutions, i.e. $X_{i,G} = x^1_{i,G}, \ldots\ldots\ldots\ldots, x^D_{i,G}$ $i = 1, \ldots\ldots\ldots, NP$ towards the global optimum [9]. The initial population should better cover the entire search space as much

as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum parameter bounds $X_{\min} = x^1_{\min},.....,x^D_{\min}$ and $X_{\max} = x^1_{\max},.....x^D_{\max}$ .For example, the initial value of the jth parameter in the ith individual at the generation is generated by

$$x^j_{i,0} = x^j_{\min} + rand\ 0,1 \square x^j_{\max} - x^j_{\min}\ , j = 1, 2.....D..... \quad 4$$

Where rand(0,1) represents a uniformly distributed random variable within the range [0,1].

3.1.1 Mutation Operation

After initialization, DE employs the mutation operation to produce a mutant vector $V_{i,G}$ with respect to each individual $X_{i,G}$ so-called target vector, in the current population. For each target vector $X_{i,G}$ at the generation G, its associated mutant vector $V_{i,G} = v^1_{i,G}, v^2_{i,G},.....v^D_{i,G}$ can be generated via certain mutation strategy. For example, the five most frequently used mutation strategies implemented in the DE codes are listed as follows:

"DE/rand/1":

$$\boldsymbol{V_{i,G} = X_{r_1^i,G} + F\square\ X_{r_2^i,G} - X_{r_3^i,G}} \quad \text{.....} \quad 5.1$$

"DE/best/1":

$$\boldsymbol{V_{i,G} = X_{best,G} + F\square\ X_{r_1^i,G} - X_{r_2^i,G}} \quad \text{.....} \quad 5.2$$

"DE/rand-to-best/1":

$$V_{i,G} = X_{i,G} + F\square\ X_{best,G} - X_{i,G}\ + F\square\ X_{r_1^i,G} - X_{r_2^i,G} \quad \text{.....} \quad 5.3$$

"DE/best/2":

$$V_{i,G} = X_{best,G} + F\square\ X_{r_1^i,G} - X_{r_2^i,G}\ + F\square\ X_{r_3^i,G} - X_{r_4^i,G} \quad \text{.....} \quad 5.4$$

"DE/rand/2":

$$V_{i,G} = X_{r_1^i,G} + F\square\ X_{r_2^i,G} - X_{r_3^i,G}\ + F\square\ X_{r_4^i,G} - X_{r_5^i,G} \quad \text{.....} \quad 5.5$$

The indices $r^i_1, r^i_2, r^i_3, r^i_4, r^i_5$ are mutually exclusive integers randomly generated within the range $1, NP$ which are also different from the index i. These indices are randomly generated once for each mutant vector. The scaling factor F is a positive control parameter for scaling the difference vector. $X_{best,G}$ is the best individual vector with the best fitness value in the population at generation G.

3.1.2 Crossover Operation

After the mutation phase, crossover operation is applied to each pair of the target vector $X_{i,G}$ and its corresponding mutant vector $V_{i,G}$ to generate a trial vector:

$U_{i,G} = u^1_{i,G},.....,u^D_{i,G}$ .In the basic version, DE employs the binomial (uniform) crossover defined as follows:

$$u^j_{i,G} = \begin{cases} v^j_{i,G}, & if\ \ rand_j\ 0,1 \le CR\ \ or\ \ j = j_{rand} \\ x^j_{i,G}, & otherwise \qquad\qquad j = 1,2,\ldots\ldots D \end{cases} \quad \ldots\ldots(6)$$

The crossover rate CR is a user-specified constant within the range [0, 1), which controls the fraction of parameter values copied from the mutant vector. j_{rand} is a randomly chosen integer in the range[1,*D*].The binomial crossover operator copies the j^{th} parameter of the mutant vector $V_{i,G}$ to the corresponding element in the trial vector $U_{i,G}$ if $rand_j\ 0,1 \le CR$ or $j = j_{rand}$. Otherwise, it is copied from the corresponding target vector $X_{i,G}$ .There exists another exponential crossover operator, in which the parameters of trial vector $U_{i,G}$ are inherited from the corresponding mutant vector $V_{i,G}$ starting from a randomly chosen parameter index till the first time $rand_j\ 0,1 > CR$. The remaining parameters of the trial vector $U_{i,G}$ are copied from the corresponding target vector $X_{i,G}$ .The condition $j = j_{rand}$ is introduced to ensure that the trial vector $U_{i,G}$ will differ from its corresponding target vector $X_{i,G}$ by at least one parameter. DE's exponential crossover operator is functionally equivalent to the circular two-point crossover operator [9].

3.1.3 Selection

Finally the offspring produced after the mutation and crossover operations is evaluated. Then the performance of the trial vector and its parent is compared and the better one is selected. If the parent is still better, it is retained in the population. They use either tournament selection or a deterministic selection scheme to determine which newly generated individuals are to enter the next population. Both selection processes require additional cost in sorting of the population at each generation. Also the population size is the most important factor [9]. DE usually employees a fixed population size throughout the search process. In order to achieve high computational speed, the population size should be as small as possible. The above 3 steps are repeated generation after generation until some specific termination criteria are satisfied as shown in Figure.1.5.

3.1.4 Step-Wise Algorithm

Step 1: Set the generation number G=0 and randomly initialize a population of NP individuals

$P_G = X_{1,G,}.....X_{NP,G}$ with

$X_{i,G} = X^1_{i\,G,}.....X^D_{i,G}$, $i = 1.....NP$ uniformly distributed in the range $X_{\min}, X_{\max}$,

Where $X_{\min} = x^1_{\min},.....x^D_{\min}$ and

$X_{\max} = x^1_{\max},.....x^D_{\max}$

Step 2: WHILE stopping criterion is not satisfied.

DO

Step 2.1: Mutation step

/* Generate a mutated vector

$V_{i,G} = v^1_{i,G},.....,v^D_{i,G}$ for each target vector

$X_{i,G} * /$

FOR i = 1 to NP

Generate a mutated vector

$V_{i,G} = v^1_{i,G}, v^2_{i,G},.....v^D_{i,G}$

corresponding to the target vector $X_{i,G}$

via one of the strategies.

END FOR

Step 2.2: Crossover step

/*Generate a trial vector

$U_{i,G} = u^1_{i,G},.....,u^D_{i,G}$ for each target vector $X_{i,G} * /$

Binomial Crossover

FOR i = 1 to NP

$j_{rand} = rand\ 0,1 * D$

FOR j = 1 to D

$$u^j_{i,G} = \begin{cases} v^j_{i,G}, & if\ \ rand_j\ 0,1 \le CR\ \ or\ \ j = j_{rand} \\ x^j_{i,G}, & otherwise \end{cases} \quad , \quad j = 1,2,.....D$$

END FOR

END FOR

Exponential Crossover

FOR i = 1 to NP

$j = \lfloor rand\ 0,1 * D \rfloor, L = 0$

$U_{i,G} = X_{i,G}$

DO

$u^j_{i,G} = v^j_{i,G}$

$j = \langle j+1 \rangle_D$

$L = L+1$

WHILE $rand\ 0,1 < CR \& L < D$

END FOR

Step 2.3: Selection Step

/* Selection */

FOR i = 1 to NP

Evaluate the trial vector $U_{i,G}$

IF $f\ U_{i,G} \le f\ X_{i,G}$, THEN

$X_{i,G+1} = U_{i,G}, f\ X_{i,G+1} = f\ U_{i,G}$

IF $f\ U_{i,G} < f\ X_{best,G}$, THEN

$X_{best,G} = U_{i,G}, f\ X_{best,G} = f\ U_{i,G}$

END IF

END IF

END FOR

Step 2.4: Increment the generation count

$G = G+1$

Step 3: END WHILE

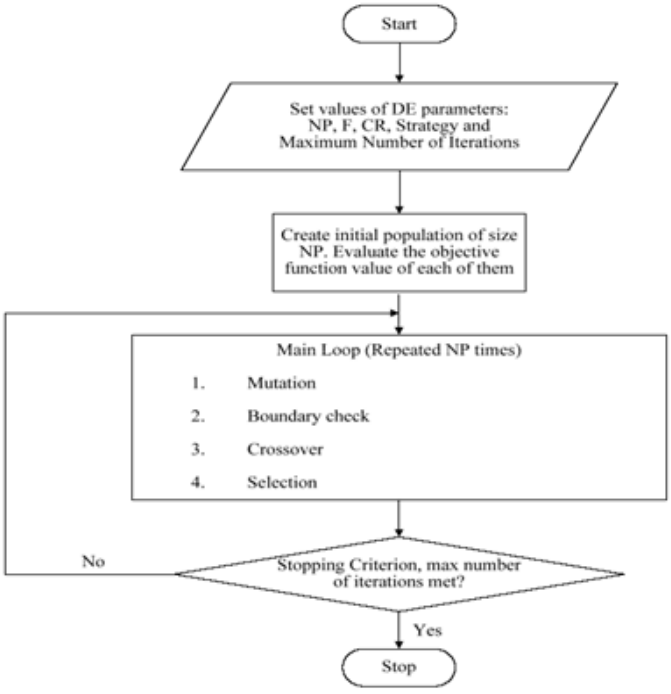

Fig.1.5 .Flow chart of a DE algorithm

The parameters used for optimization in the DE algorithm are

Strategy = 5

$$V_{i,G} = X_{r_1^i,G} + F_1 \left(X_{r_2^i,G} - X_{r_3^i,G} \right) + F_2 \left(X_{r_4^i,G} - X_{r_5^i,G} \right)$$

V = Mutant Vector

X = target Vector

Scaling factor (F_1) = 0.5

Scaling factor (F_1) = 0.3

Crossover Rate (CR) = 0.8

Population (NP) = 20

Generations (G) = 1000

Computational Experiments

4.1 Simulation using 11 Sources

11 Sine waves having frequency ranging from **10 Hz** and above with 10Hz frequency difference and sampling time **0.01 sec** are taken as source signal and passed through a mixing channel denoted by mixing matrix **A** as shown in Figure.1.6.

4. SIMULATION AND RESULTS

Algorithm Parameters

0.814724	0.970593	0.849129	0.046171	0.186873	0.498364	0.547216	0.251084	0.547216	0.251084	0.380446	0.337123	0.450542
0.905792	0.957167	0.933993	0.097132	0.489764	0.959744	0.138624	0.616045	0.138624	0.616045	0.567822	0.162182	0.083821
0.126987	0.485376	0.678735	0.823458	0.445586	0.340386	0.149294	0.473289	0.149294	0.473289	0.075854	0.794285	0.228977
0.913376	0.80028	0.75774	0.694829	0.646313	0.585268	0.257508	0.35166	0.257508	0.35166	0.05395	0.311215	0.913337
0.632359	0.141886	0.743132	0.317099	0.709365	0.223812	0.840717	0.830829	0.840717	0.830829	0.530798	0.528533	0.152378
0.09754	0.421761	0.392227	0.950222	0.754687	0.751267	0.254282	0.585264	0.254282	0.585264	0.779167	0.165649	0.825817
0.278498	0.915736	0.655478	0.034446	0.276025	0.255095	0.814285	0.549724	0.814285	0.549724	0.934011	0.601982	0.538342
0.546882	0.792207	0.171187	0.438744	0.679703	0.505957	0.243525	0.917194	0.243525	0.917194	0.129906	0.262971	0.996135
0.957507	0.959492	0.706046	0.381558	0.655098	0.699077	0.929264	0.285839	0.929264	0.285839	0.568824	0.654079	0.078176
0.964889	0.655741	0.031833	0.765517	0.162612	0.890903	0.349984	0.7572	0.349984	0.7572	0.469391	0.689215	0.442678
0.157613	0.035712	0.276923	0.7952	0.118998	0.959291	0.196595	0.753729	0.196595	0.753729	0.011902	0.748152	0.106653

Fig.1.6. Mixing Matrix A

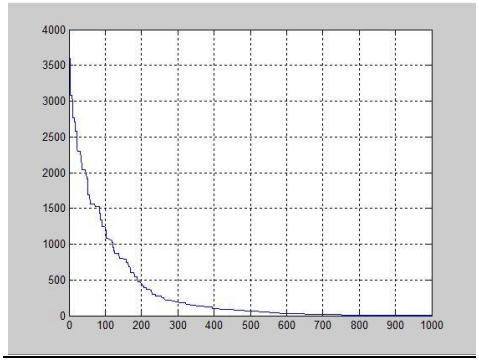

Fig 1.7. Error convergence curve of 1000 generations

Optimization Results

Train Error = 1.9448110450385

Test Error = 0.0277830149291215

Error in training = 0.027783014929 per pattern trained

Error in testing = 0.000926100498 per pattern tested

50^{th} pattern of all input sources signals, distorted sources signals output estimated signals and error convergence curve were displayed for comparison as shown in Figure.1.7-Figure.1.10.

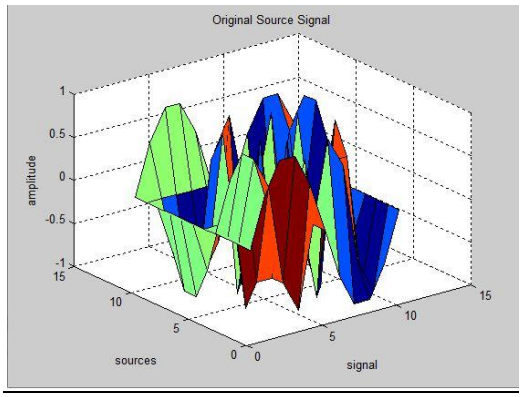

Fig.1.8.Original Source Signal

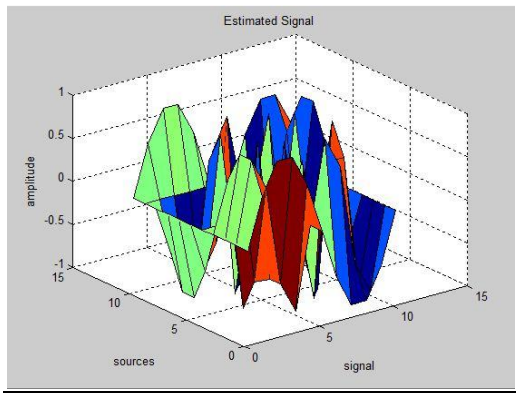

Fig 1.10.Estimated Signal

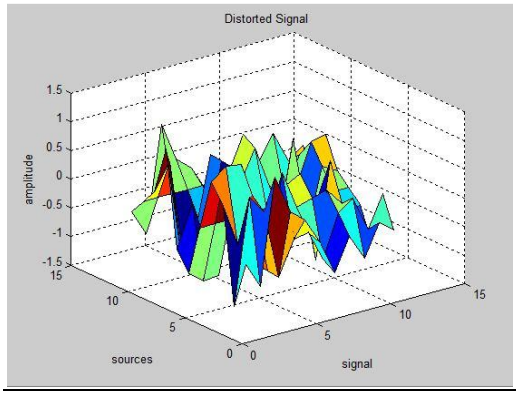

Fig 1.9.Distorted Signal

4.2 Simulation using 20 Sources

20 Sine waves having frequency ranging from **10 Hz** and above with 10Hz frequency difference and sampling time **0.01 sec** are taken as source signal and passed through a mixing channel denoted by mixing matrix **A** as shown in Figure.1.11.

0.26	0.83	0.15	0.70	0.39	0.34	0.31	0.4	0.36	0.80	0.2	0.43	0.70	0.62	0.73	0.92	0.83	0.45	0.82	0.15
0.89	0.65	0.51	0.66	0.92	0.25	0.22	0.77	0.31	0.99	0.87	0.57	0.72	0.92	0.60	0.68	0.93	0.92	0.09	0.91
0.01	0.25	0.91	0.91	0.98	0.22	0.83	0.09	0.32	0.43	0.52	0.22	0.98	0.32	0.38	0.06	0.01	0.49	0.43	0.48
0.96	0.66	0.08	0.25	0.46	0.24	0.26	0.9	0.09	0.34	0.22	0.29	0.47	0.70	0.57	0.61	0.74	0.82	0.82	0.55
0.87	0.87	0.81	0.02	0.61	0.15	0.62	0.52	0.37	0.72	0.62	0.04	0.39	0.30	0.73	0.95	0.99	0.67	0.68	0.64
0.93	0.49	0.55	0.39	0.89	0.56	0.30	0.9	0.31	0.26	0.79	0.12	0.59	0.23	0.50	0.62	0.25	0.58	0.73	0.53
0.86	0.39	0.35	0.29	0.59	0.39	0.72	0.17	0.47	0.50	0.96	0.68	0.29	0.12	0.42	0.76	0.34	0.62	0.45	0.32
0.34	0.87	0.66	0.10	0.50	0.29	0.21	0.82	0.85	0.31	0.12	0.02	0.37	0.46	0.91	0.56	0.26	0.39	0.16	0.14
0.13	0.68	0.35	0.17	0.48	0.11	0.74	0.95	0.61	0.21	0.65	0.24	0.53	0.44	0.49	0.52	0.11	0.03	0.36	0.72
0.59	0.64	0.15	0.03	0.75	0.45	0.58	0.85	0.99	0.91	0.06	0.34	0.74	0.71	0.55	0.72	0.28	0.06	0.16	0.61
0.62	0.50	0.70	0.50	0.99	0.24	0.44	0.82	0.62	0.62	0.78	0.53	0.62	0.97	0.76	0.24	0.44	0.90	0.70	0.53

Fig.1.11.Mixing Matrix A

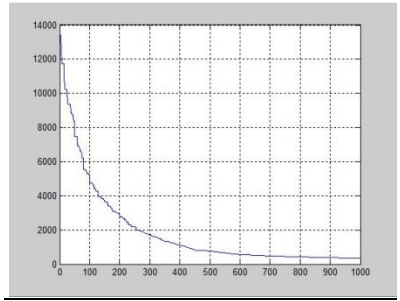

Fig.1.12.Error convergence curve of 1000 generations

Optimization Results

Train Error = 338.436040547461

Test Error = 4.83480057924944

Error in training = 4.834800579249 per pattern trained

Error in testing = 0.161160019308 per pattern tested

Signal System

50th pattern of all input sources signals, distorted sources signals output estimated signals and error convergence curve were displayed for comparison as shown in Figure1.12-Figure.1.15.

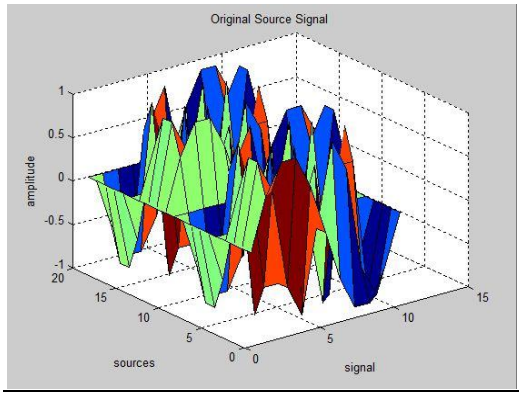

Fig.1.13.Original Source Signal

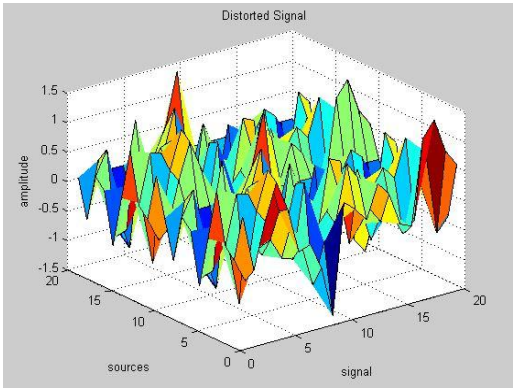

Fig.1.14.Distorted Signal

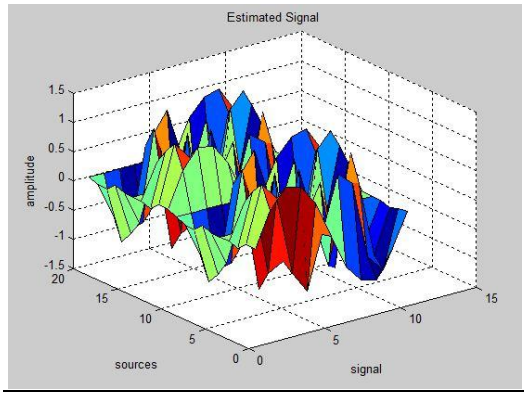

Fig.1.15.Estimated Signal

5. CONCLUSION

In this paper we have studied an effective evolutionary algorithm termed DE used for solving the blind source separation problem. DE is a very simple and straightforward strategy. Three crucial control parameters involved in DE, i.e., **population size, scaling factor, and crossover rate,** may significantly influence the optimization performance of the DE. DE is a stochastic direct search method and thus has the advantage of easy implementation to the experimental miniaturization. The method remains efficient on linear instantaneous mixtures. Initial experiment shows the performance of DE in terms of number of sources to be separated. It also shows an excellent performance in separating a large number of data sets.

6. FUTURE WORK

An interesting opportunity for future research would be the extension of this method to the real world environment where convolutive mixtures are involved. Noise strongly limits the separation performance, encouraging us to use the denoising processing. We further intend to carry out comparison of DE performance with other evolutionary algorithms. Performance of DE in combination with other optimization algorithms for quick problem solving applications while maintaining its simplicity remains an area of interest for us.

7. ACKNOWELDEGEMENT

We would like to express our sincere thanks to Dr. R. N. Pal, retired Professor IIT Kharagpur, India for his valuable comments on an earlier version of this paper and suggestions concerning the test cases.

8. REFERENCES

[1] A.K. Qin, V. L. Huang, and P. N. Suganthan "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization" IEEE Trans. On Evolutionary Computation 1,2008.

[2] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proc. IEEE Congr. Evolut.Comput.*, Edinburgh, Scotland, Sep. 2005, pp. 1785–1791.

[3] Hyvarinen, et al., "Independent Component Analysis", John Wiley & Sons Co, 2001.

[4] Hyvärinen and E. Oja, "Independent Component Analysis:Algorithms and Applications", Neural Networks, vol 13, pp. 411-430, 2000.

[5] Andrzej Cichocki and Frank Ehlers "Advances in Blind Source Separation" EURASIP Journal on Advances in Signal Processing Volume 2007

[6] Cardoso, J. F., (1999), "High-order Contrasts for IndependentComponent Analysis", Neural Computation, Vol. 11(1), pp. 157-192.

[7] F. Yin, J. Wang, and C. Guo (Eds.): "Frequency-Domain Separation Algorithms for Instantaneous Mixtures" Springer-Verlag Berlin Heidelberg 2004, pp. 666–671, 2004.

[8] M. Kadou, K. Arakawa, "A Method of Blind Source Separation for Mixed Voice Separation in Noisy and Reverberating Environment", IEICE Tech. Rep., vol. 108, no. 461, SIS2008-81, pp. 55- 59, March 2009.

[9] Storn, R. and Price, K., (1997), "Differential Evolution –A Simple and Efficient Heuristic for Global Optimizationover Continuous Spaces", Journal of Global Optimization,Vol. 11, pp. 341-359.

[10] S. C. Douglas, H. Sawada, and S. Makino, "Natural gradient multichannelblind deconvolution and speech separation using causal FIR filters," *IEEE Trans. Speech Audio Process.*, vol. 13, pp. 92–104, Jan.2005.

[11] Shuxue Ding, Member, IEEE, Jie Huang, Member, IEEE, Daming Wei, Member, IEEE, and Andrzej Cichocki, Member, IEEE "A Near Real-Time Approach for Convolutive Blind Source Separation" IEEE Trans. On Circuits and systems—I: Regular Papers, vol.53, no. 1, January 2006

[12] Y. Gao, S. Xie, "A blind source separation algorithm using particle swarm optimization", Proceedings of the IEEE 6th Circuits and Systems Symposium, Vol. 1, Issue , pp. 297 - 300 Vol.1, 31 May-2 June 2004.

[13] Yan Wanga,_, Xiao-Yue Fenga, Yan-Xin Huanga, Dong-Bing Pub, Wen-Gang Zhoua,Yan-Chun Lianga, Chun-Guang Zhoua " A novel quantum swarm evolutionary algorithm and its applications" Neurocomputing 70 (2007) 633–640

[14] Yan Li', Peng Wen' and David Powers' "METHODS FOR THE BLIND SIGNAL SEPARATION PROBLEM" IEEE Int. Conf. Neural Networks & Signal Processing Nanjing, China, December 14-17.2003

[15] Z. Shi, Z. Jiang and F. Zhou, "A fixed-point algorithm for blind source separation with nonlinear autocorrelation," Journal of Computational and Applied Mathematics, 223 908–915, 2009.