International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 3, Issue 8, August 2014

Comparison of Metaheuristic Algorithms through Application in Noisy Speech Signal using Adaptive Filtering Approach

Ritika Thakur[1], Papiya Dutta[2]

***Abstract*— An improved method for adaptive noise canceller (ANC) is proposed for noisy speech signal in case of random noise. In this approach, ANC is implemented through four different metaheuristic techniques. A comparative study of the performance of various metaheuristic techniques such as Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Cuckoo Search (CS) and Modified Cuckoo Search (MCS) has been done. Fidelity parameters like signal to noise ratio (SNR) in dB, mean square error (MSE) and maximum error (ME) are observed with varying range of input SNR and it was found that the Modified Cuckoo Search based speech denoising approach give better performance in terms of SNR as compared to other Metaheuristic techniques. The quantitative (SNR, MSE and ME) and visual (Denoised speech signals) results show superiority of the proposed technique over the conventional and state -of-art speech signal denoising techniques.**

***Keywords*— Adaptive filters, Adaptive Algorithms, Artificial Bee Colony, Cuckoo search, Modified Cuckoo Search, Particle Swarm Optimization Algorithm, Speech Enhancement.**

I. INTRODUCTION

The problem of noise cancellation has received considerable attention in recent years in many speech applications, such as speech bandwidth compression, speech recognition, speaker verification, and [1]-[3]. For example, in automatic speech recognition from noise-corrupted speech, the noise cancellation schemes provide an improved quality of speech signal that helps in achieving a better recognition performance. The most common problem in speech processing is the effect of interference noise in the signals. This noise masks the speech signal and reduces its intelligibility. Various sources which produce interferences may include ventilation equipment, traffic, crowds and commonly, echoes and reverberation. It can also arise electronically from thermal noise, distortion products or tape hiss. If the sound system has large peaks in its frequency response, the speech signal can end up masking itself [2]. Quantitative measurements and computer-aided analysis become difficult and unreliable due to poor speech signal quality. Thus, denoising and enhancement of the speech become necessary for many practical applications. Noise reduction or speech enhancement algorithms are used to suppress background noise and improves the quality and intelligibility of speech. Conventional linear filtering is the most common way to do single trial analysis of speech signal, by which contaminations due to on-going background noise can be attenuated from the speech signal. A major difficulty in conventional linear filtering is very low SNR, since then, the concept of adaptive filtering was introduced.

Adaptive Filters: An adaptive filter adapts itself to changes in its input signals automatically according to a given algorithm. The algorithm will change the filter coefficients according to a given criteria, typically an error signal is given to improve its performance. In essence an adaptive filter is a digital filter combined with an adaptive algorithm, which is used to modify the filter coefficients and work as an adaptive noise canceller (ANC). Adaptive filters are used in many diverse applications like radar signal processing, telephone echo cancelling, equalization of communication channels and biomedical signal enhancement [4] [5], [6].

Adaptive Algorithms: Adaptive noise cancellation (ANC) uses various minimization techniques or adaptive algorithms like LMS, NLMS and RLS. These adaptive algorithms are Gradient based algorithm which are most commonly used due to simplicity in computation and easy implementation. These gradient based algorithms suffers from some problem like these are not suitable for multimodal error surface and it gives only one possible solution for each iteration according to generated error. The aim of this paper is to solve the problem of ANC, but these problems are not solved by conventional algorithm so here Gradient free optimization algorithms (PSO, ABC, CS and MCS) are used. Optimization algorithm increased the probability of encountering the global optimum.

Performance of the MCS based denoising technique performs better than other optimizing techniques. The visual and quantitative results are given in results and discussions section. The paper is organized as follows. Section-II gives an overview of ANC for speech denoising. Section-III introduces PSO, ABC, CS and MCS Algorithm for speech denoising case. In section-IV, qualitative and quantitative results of the proposed method with PSO, ABC and CS supported by SNR, MSE and ME have been discussed. Conclusions are given in the final section.

II. OVERVIEW OF ADAPTIVE NOISE CANCELLATION

Noise cancellation technology is a growing field that capitalizes on the combination of disparate technological advancements. This aims to cancel or at least minimize unwanted signal and so to remedy the excess noise that one may experience. There are already several solutions available [7], [8]. Adaptive noise cancellation is widely used to improve the Signal to Noise Ratio (SNR) of a signal by removing noise from the received signal. The basic structure and working of ANC is shown in Fig.1, In this configuration the input x(n) is primary input signal fed into transversal filter and generates filtered output y(n) which is shown as $y(n) = w(n)^H x(n)$ .The filter output y(n) is compared with the reference signal d(n) and produces error e(n), mathematical representation is e(n)=d(n)– y(n). This error is used to update new filter coefficients w (n) for transversal filter.

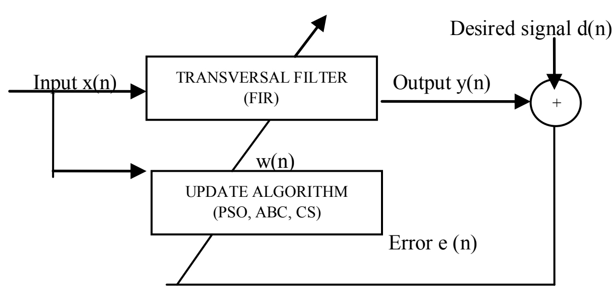


Fig. 1 Adaptive Noise Cancellation Configuration

The error signal should converge to the desired signal d(n), but it will not converge to the exact desired signal. In other words, the difference between the desired signal d(n) and the error signal e(n) will always be greater than zero. The only option to this problem is to minimize the difference between those two signals using certain error minimization techniques [5].

III. DIFFERENT METAHEURISTIC TECHNIQUES FOR DENOISING OF SPEECH SIGNAL

Swarm intelligence (SI) is an innovative technique for problem optimization which is inspired by the social behaviours of animal swarms in nature, examples of systems like fish schooling, bird flocking, ant colonies, bacteria molding and animal herding. SI is an artificial intelligence technique based on animal behavior in which no centralized control (which shows how individual agents should behave) and self-organized systems. The Global behavior is achieved by local interactions between agents. Recent years, several swarm intelligence algorithms have been proposed [9]. Swarm based algorithm are used in various applications to solve different kind of problems which has many possible solutions.

A. PARTICLE SWARM OPTIMIZATION TECHNIQUE

Particle Swarm Optimization was proposed in 1995 by James Kennedy and R. C. Eberhart. PSO is an Evolutionary Swarm technique based on stochastic optimization and inspired by social behavior of bird flocking. Here birds find the food by the social cooperation with other birds. They move in random direction to find a food and all bird follow the positions and who is closest to food will inform to other one of the member, but there is no leader. All of them hold their and all other birds will update their position. This process will continue until they reach to food location [10-12].

A.1) PSO Algorithm with Adaptive Filter:

The main motive of adaptive filter with PSO algorithm is to optimize (minimize) the cost function which is calculated by Mean squared difference i.e. MSE, between desired signal and filtered output. This value is used to calculate fitness value of each particle. The cost function or MSE C (n) (the MSE for the i^{th} particle at the nth iteration) is defined as [10, 13-15]:

$$C_i(n) = \frac{1}{M}\sum_{l=1}^{M}[e_{li}(n)^2] \qquad (1)$$

Where, e_{li} (n) is l^{th} error for i^{th} particle and M is number of input sample.

A.2) Procedure of PSO algorithm with adaptive filters:

The PSO algorithm is start with initialization of particles and their velocity with random swarm within a problem space and remaining procedure of PSO algorithm with ANC can be described in following steps [15]:

Step1. Start: Decide the problem space and define the maximum and minimum limit for particle ($\pm X_i$) i.e. weight vector and their velocity vector ($\pm V_i$).

Step2. Initialize particles X_i with random positions and their corresponding velocities V_i according the problem space.

Step3. Update Position: If present position is within the problem space than follow the next step otherwise adjust the positions.

Step4. Fitness function: Calculate fitness function using Eq.(1) For each particle:
Calculate P_{best}: Compare fitness value of particle with its $P_{best,}$, if current value is better then set P_{best} equal to the current value.

Calculate G_{best}: Find present global minimum from best position of particles, and assign value to the variable g (G_{best}).

Step5. Update V_i and X_i: Update the velocity and position of the particle:

$$V_{pd}(i+1) = W(i)\, V_{pd}(i) + c_1 r_1 (X_{ppd}(i) - X_{pd}(i)) + c_2 r_2 (X_{gpd}(i) - X_{pd}(i)) \quad (2)$$

$$X_{pd}(i+1) = X_{pd}(i) + V_{pd}(i+1) \quad (3)$$

X_i represents the i_{th} particle; P_i represents the best previous position; V_i represents the velocity; c_1 and c_2 are positive constant commonly set to 2. The r_1 and r_2 are two random numbers within range 0 to 1.

$F_{pp}(i)$ and $F_{gp}(i)$ are the personal best position and global best position. $F_{pp}(i)$ is initialized with F_{pd}, which is calculated in step 4 and the best of $F_{pp}(i)$ is taken $F_{gp}(i)$ at initial step.

X_{ppd} (i) and X_{gpd} (i) are the personal best position and the global best position respectively. The position of $F_{pp}(i)$ is stored in $X_{ppd}(i)$ and position of $F_{gp}(i)$ is stored in $X_{gpd}(i)$.

Step6. The updating of position and velocity is restricted by the boundary value which is 80% of maximum and minimum value of the particle search space.

Step7. Calculate the fitness $F_{pd}(i)$ at new search position. Update $F_{pp}(i)$ if current value of $F_{pd}(i)$ is less than the current value of $F_{pp}(i)$, otherwise retain old $F_{pp}(i)$. Update $F_{gp}(i)$ if best of $F_{pp}(i)$ is less than previous $F_{gp}(i)$ otherwise retain $F_{gp}(i)$, similarly $X_{ppd}(i)$ and $X_{gpd}(i)$ is updated accordingly.

Step8. Repeat step 5 to step 7 till stopping criteria or maximum number of iteration is achieved.

B. ABC ALGORITHM

Artificial Bee Colony (ABC) is one of the recently defined algorithms by D. Karaboga in 2005 [16]. The algorithm is specifically based on the model proposed by Tereshko and Loengarov (2005) for foraging the behaviour of honey bee colonies. It is motivated by the intelligent behaviour of bees. The ABC algorithm is as simple as PSO and differential evolutionary algorithms, and is also easy to implement. ABC algorithm uses the common control parameters like colony size and maximum number of cycle. ABC as an optimization tool provides a population based search in which individuals known as foods positions are modified by the artificial bees with time. The bee's aim is to search or to discover the places of food sources with high nectar amount and finally the one with the highest nectar. Initially, all food source positions are searched by scout bees and then the nectar of food sources are exploited by the employed bees and onlooker bees, and this continual exploitation will ultimately cause them to become exhausted.

In ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. The Onlooker bee waits on the dance area for making decision to choose a food source. A bee going to the food source visited by it previously is named an employed bee. A bee carrying out random search is called as scout.

The detailed algorithm steps for the optimization of the (thresholding) function are as follows:

Step1: Initialization of control parameters. The main control parameters of the ABC algorithm is colony size and number of iterations. So, initialize the colony (CS) size by 6.Limit of the scout bees is given by L= (CS*D)/2. The dimension D of the problem is three.

Step2: Initialize the position of CS/2 food source of employed bees, randomly using the variables over a defined range. The range is taken as λ=1 to 150, k=0.1 to 2, m=1 to 4 and find CS/2 solutions.

Step 3: Evaluate fitness for each of the obtained solution.

$$C_i(n) = \frac{1}{M}\sum_{l=1}^{M}[e_{li}(n)^2] \quad (1)$$

Fitness function:

$$fitness_i = \begin{cases} \dfrac{1}{1+C_i}, & if \;\; C_i \geq 0 \\ 1 + abs(C_i) & if \;\; C_i < 0 \end{cases} \quad (4)$$

Step 4: Select the maximum value of the fitness, which is the best quality of food source.

Cycle 1: Employed bees phase

$$v_{i,j} = x_{i,j} + \varphi_{i,j}(x_{i,j} - x_{i,j}) \quad (5)$$

Where, k and j are random selection index, ϕ is randomly produced number in the range [0, 1].

Calculate $C(v_i)$ and the fitness of v_i. By comparing with the fitness of previously obtained solution with the $C(v_i)$.

Replace C_i with $C(v_i)$ if $C_i<C(v_i)$ replace it with $C(v_i)$ otherwise increase the trial counter.

Step 5: calculate the probability values p for the solutions x by means of their fitness using the equation:

$$p_i = \frac{fitness_i}{\sum_{i=1}^{CS/2} fitness_i} \quad (6)$$

Step 6: Onlooker bee's phase:

Produce the new solution v_i for onlooker bees from the solutions xi selected, and depending upon the probability pi evaluate them. Then apply greedy selection between $C(v_i)$ and C_i.

Step 7: memories the best solution and corresponding vector.

Step 8: scout bee phase, for replacing the abandoned solution (the solution of which trial is more than L), a new random solution is generated.

Step 9: update cycle and repeat the process until stopping criteria is achieved.

C. CUCKOO SEARCH ALGORITHM

In 2009, Yang and Deb have proposed the Cuckoo Search (CS) optimization algorithm [17]. The CS algorithm is a new meta-heuristic algorithm for solving the optimization problems. It is inspired by the obligate brood parasitism of some Cuckoo species by laying their eggs in the nests of other host birds (of other species). Some host birds can engage direct conflict with the intruding Cuckoos. The algorithm is based on the obligate brood parasitic behavior of some cuckoo species in combination with the L'evy flight behavior of fruit flies and some of the birds. Each egg in cuckoo search algorithm represents a solution and Cuckoo egg represents a new solution. Overall there is aim to employ new and potentially better solutions to replace weak solutions in the nests. In the simplest form, each nest has one egg. The Cuckoo algorithm can be extended to more complicated cases, which are having more than one egg representing a set of solutions. The CS is based on three idealized rules:

(i) Each Cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;

(ii) The best nests with high quality of eggs (solutions) will carry over to the next generations;

(iii) The hosts nests are fixed in count, and a host can discover an alien egg with probability $p_a \in [0\ 1]$. In this case, the host bird can either throw the egg away or abandon the nest to build a completely new nest in a new location.

Based on above mentioned rules, the basic steps of Cuckoo algorithm are:

Step 1: Set the number of nest. Nest is nothing but different solutions. In this problem, it is taken as 20. Set the probability with a discovery rate (probability). Set the stopping criteria, which is either fixed number of iteration or tolerance value. Set dimension of the problem. The number of dimension is 3 here. Also set the boundaries of the parameters.

Step 2: Randomly initialize the solution, by generating n different nest for obtaining n different solutions.

Step 3: Evaluate fitness for each of the obtained solution. Find the best nest corresponding to minimum value of fitness.

Step 4: Start iteration, generate new nest by Levy flight but keep the current best. A Levy flight can be formed as Cuckoo i, a Levy flight is performed by the equation:

$$x_i(t+1) = x(t)_i + \alpha \oplus L'evy(\lambda) \qquad (7)$$

Where, α is step size. It essentially provides a random walk while the random step length is drawn from L´evy distribution, which has an infinite variance with an infinite mean. L'evy distribution is given by:

$$L'evy\ u = t^{-\lambda}, \quad (1 < \lambda \leq 3) \qquad (8)$$

L'evy function can be changed according to application. Mantegna's algorithm is one of the L'evy function.

Step 5: Evaluate this set of solutions and obtain the new fitness. Compare old fitness with this new fitness, and replace old fitness if new fitness is better than old one. Update the best nest corresponding to fitness.

Step 6: Repeat the above process until some stopping criteria is achieved giving the best fitness and corresponding best nest.

D. MODIFIED CUCKOO SEARCH ALGORITHM

A modification of the standard Cuckoo Search was made by Walton et al. [18] with the aim to speed up convergence. The modification involves the additional step of information exchange between the top eggs. It was shown that Modified Cuckoo Search (MCS) outperforms the standard cuckoo search and other algorithms, in terms of convergence rates, when applied to a series of standard optimization benchmark objective functions.

The first modification was to the L'evy flight coefficient, α. In the CS, the α is constant and typically the value α = 1 is employed [17]. In MCS, the value of α was made decrease as the number of generations increased. An initial value of the L'evy flight coefficient α=A=1 was selected, and at each generation, a new value is calculated using,

$$\alpha = A / \sqrt{G} \qquad (9)$$

where A is Maximum L'evy step size and G is the generation number . This exploratory search is only performed on those nests that are to be abandoned.

Crossover between cuckoos, which survive to the next generation, is modeled as follows:

1. Each of the top elite eggs randomly picks a second elite egg.

2. A new egg is generated along the line which connects these two eggs and the fitness evaluated. The distance

along this line at which the new egg is located is calculated, using the inverse of the golden ratio

$$\varphi = \left(1 + \sqrt{5}\right) / 2 \qquad (10)$$

such that it is located closer to the egg with the better fitness. In the case that both eggs have the same fitness, the new egg is generated at the midpoint. There is a possibility that, in the previous step, the same egg was picked twice.
In this case, a local L'evy fight search is performed, from the randomly picked nest, with L'evy flight coefficient

$$\alpha = A / G^2 \qquad (11)$$

3. A nest is picked at random, from all nests. The egg generated in the process above is compared with the egg in the random nest. If the newly generated egg is of better fitness than the egg already in that nest, then it replaces it, otherwise it is discarded. Once all eggs which are not in nests are discarded, the group of top eggs is updated and the process iterates.

There is a possibility, particularly during crossover, that a new egg may be generated at the same location as an existing egg. To evaluate the objective function again at this location would be wasteful, so a check is made to see if the newly generated egg exists. If it is found that the newly generated egg does already exist in the population, a local L'evy flight search is performed. Ideally, it would be good to check if the location of a new egg has been visited previously by the algorithm, but this would require a potentially prohibitively large amount of storage, so it is considered unfeasible.

IV. RESULTS AND DISCUSSION

In this section, MCS was compared to PSO, ABC and CS. Performance of the proposed scheme is computed by determining different fidelity parameters such as SNR, MSE and ME given by Eqs (12), (13) and (14) respectively In this paper, denoising performance of PSO algorithm, ABC algorithm, Cuckoo Search (CS) and Modified Cuckoo Search (MSC) algorithm is compared on the basis of SNR, MSE and ME. For making this comparison, PSO, ABC, CS and MCS algorithm is executed with MATLAB R2012a. The performance result of various metaheuristic algorithms is shown in Fig.2, Fig.3, Fig.4 and Table I, Table II and Table III shows the results obtained by running the algorithms for an input SNR of 5 dB, 10 dB and 15 dB respectively at 400 iterations. It was observed that the MCS algorithm gives better value of SNR, MSE and ME as compared to PSO, ABC and CS algorithm.

Fidelity parameters:

(i) $$SNR_{dB} = 10\log_{10}\left(\frac{Desired\_Signal}{Error}\right)^2 \qquad (12)$$

(ii) $$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Desired\_Signal - Error\right)^2 \qquad (13)$$

(iii) $$ME = max\ [abs\ (Desired\_\ Signal - Error)]^2 \qquad (14)$$

TABLE I
For 5 dB input SNR

S.No.	Algorithm	SNR(dB)	MSE input	MSE Out	ME input	ME Out
1	PSO	20.4927	0.0218	0.0030	-5.9809e-006	-3.6356e-007
2	ABC	23.7843	0.0218	0.0021	-1.0417e-006	-5.9249e-008
3	CS	26.0071	0.0218	0.0016	-9.7086e-006	-9.0430e-008
4	MCS	26.4682	0.0218	0.0015	-9.7129e-006	-9.032e-008

TABLE II

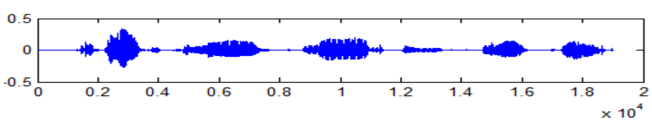
For 10 dB Input SNR

S.No	Algorithm	SNR(dB)	MSE input	MSE Out	ME input	ME Out
1	**PSO**	28.0091	0.0107	0.0012	-4.1868e-007	-2.2596e-007
2	**ABC**	34.8703	0.0107	6.0357e-004	-1.1083e-006	-8.8172e-008
3	**CS**	38.1462	0.0107	3.9771e-004	-3.8780e-007	-9.5410e-008
4	**MCS**	38.7546	0.0107	3.8471e-004	-3.8412e-007	-9.1521e-008

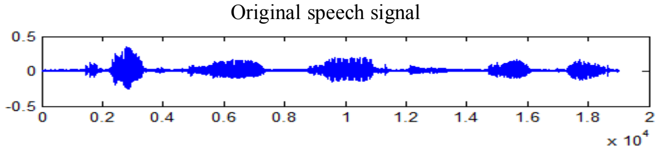
TABLE III

For 15 dB Input SNR

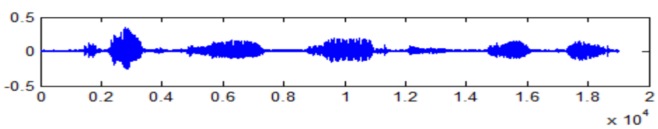
S.No.	Algorithm	SNR(dB)	MSE input	MSE Out	ME input	ME Out
1	**PSO**	30.1475	0.0055	9.9852e-004	-6.4021e-007	-2.4185e-007
2	**ABC**	45.1972	0.0055	1.6974e-004	-4.1940e-007	-8.8045e-009
3	**CS**	58.4689	0.0055	4.0203e-005	-1.2580e-006	-4.5944e-009
4	**MCS**	58.8214	0.0055	3.9451e-005	-1.2103e-006	-4.3001e-009



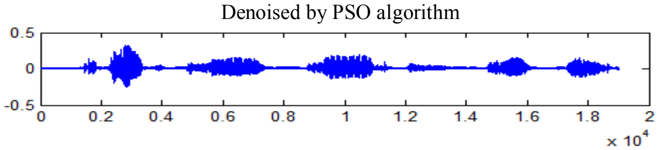
Original speech signal



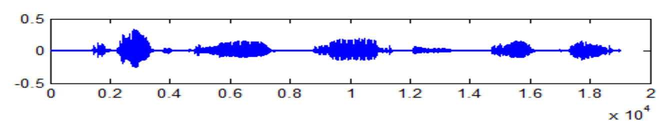
5 dB Noisy signal



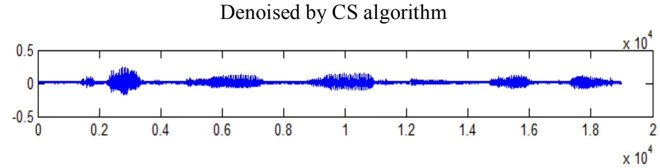
Denoised by PSO algorithm



Denoised by ABC algorithm

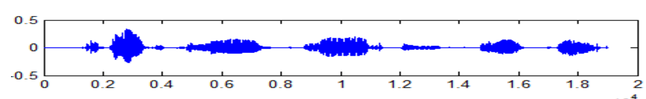


Denoised by CS algorithm

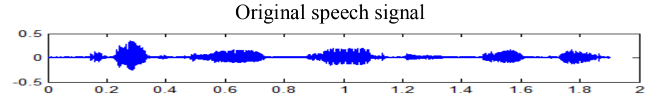


Denoised by MCS algorithm

Fig.2 Result of various Metahueristic algorithms for 400 iterations (at 5 dB Input SNR)



Original speech signal



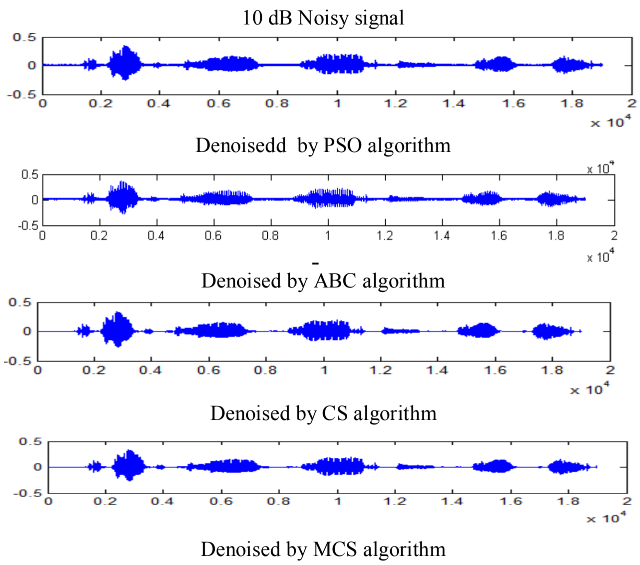


Fig.3 Result of various Metahueristic algorithms for 400 iterations (at 10 dB Input SNR)

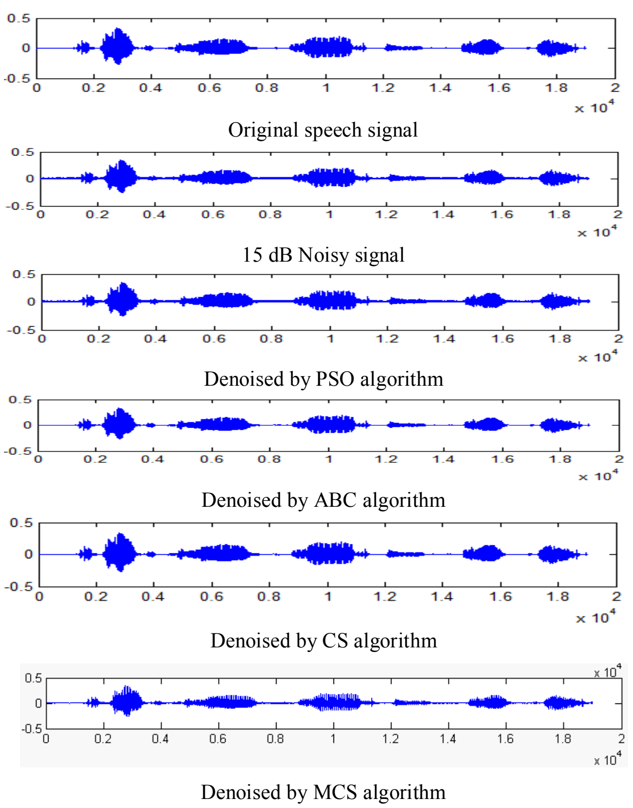


Fig.4 Result of various Metahueristic algorithms for 400 iterations (at 15 dB Input SNR)

V. CONCLUSION

In this work, an improved denoising scheme for speech signal is presented using different evolutionary algorithms such as CS algorithm, ABC algorithm and PSO algorithm. The fidelity parameters obtained clearly show superiority of the proposed technique over other conventional speech denoising techniques. A comparative study of different evolutionary algorithms has also been made, and it has been found that the proposed technique based on CS algorithm and ABC algorithm yields better performance as compared to PSO in terms of SNR, MSE and ME.

REFERENCES

[1] J.M. Gorriz, J. Ramrez, S. C. Alvarez, C. G. Puntonet, E. W. Lang, and D. Erdogmus, Jan. 2009 **"A novel LMS algorithm applied to adaptive noise cancellation,"** IEEE Signal Processing Letters, vol. 16, pp. 44–47,

[2] Ben G., Nelson M., 2000 **"Speech and Audio Signal Processing"** John Wiley and Sons, Inc.

[3] M.S.Ahmed, Jan.1989. **"Comparison of noisy speech enhancement algorithms in terms of LPC perturbation,"** IEEE Trans. Acoust., Speech, Signal Processing, vol. 37, pp. 121–125,

[4] Simon Haykin, September 2001 **"Adaptive Filter theory",** 4thed. Prentice Hall.

[5] Allam Mousa, Marwa Qadus, Sherin Bader, April 7-9 2011"**Adaptive Noise Cancellation Algorithms Sensitivity to Parameters",** The 2nd International Conference on Multimedia Computing and Systems (ICMCS'11), IEEE Catalog Number CFP731-3-CDR, ISBN: 978-1-61284-731-3, pp. 1-5, Ouarzazate, Morocco,

[6] Colin H Hansen, 2002. **"Understanding Active Noise Cancellation",** Taylor and Francis Inc. Print ISBN: 0415231922,

[7] V.R.Vijaykumar, P.T.Vanathi, P. Kanagasapabathy, 2007. **"Modified Adaptive Filtering Algorithm for Noise Cancellation in Speech Signals"**, Electronics and Electrical Engineering, 2007 No 2(74) pp. 17-20,

[8] Xiaohui Y., Yunlong Z., Hao Z., Chen H. And Ben N., 2010 **"An Adaptive Bacterial Foraging Optimization Algorithm with Lifecycle and Social Learning"**, Hindawi Publishing Corporation Discrete Dynamics in Nature and Society, 1-20.

[9] Mitul K. A., Anil K., Girish K. S., 2012 **"Analysis and testing of PSO variants through application in EEG/ERP adaptive filtering approach"**, Biomedical Engineering Letters, , Vol. 2, 186-197.

[10] R. Eberhart, Y. Shi and J. Kennedy, 2001 **"Swarm Intelligence".** San Mateo, CA: Morgan Kaufmann.

[11] Dian Palupi Rini, Siti Mariyam Shamsuddin and Siti Sophiyati Yuhaniz, 2011 **"Particle Swarm Optimization: Technique, System and Challenges"**, International Journal of Computer Applications, , Vol. 14(1), 19-27.

[12] D.J. Krusienski and W.K.Jenkins, 2003 **"Adaptive Filtering Via Particle Swarm Optimization"**, Conf Signal Syst Computer. Vol. 1, 571-575.

[13] Upal Mahbub, Celia Shahnaz and Shaikh Anowarul Fattah, 2010 **“An Adaptive Noise Cancellation Scheme Using Particle Swarm Optimization Algorithm”**, IEEE International Conference on Communication Control and Computing Technologies, Vol.71, 683 - 686.

[14] Ali T.Al- Awami, Azzedine Zerguine, Lahouari Cheded, Abdelmalek Zidouri, Waleed Saif, 2010 **“A new modified particle swarm optimization algorithm for adaptive equalization”,** Digital Signal Processing, 195–207.

[15] A. Chatterjeea, P.Siarry, 2006 **“Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization”,** Computers and Operations Research, Vol. 33, 859–871.

[16] Karaboga, D.:2005 **‘An Idea Based On Honey Bee Swarm for Numerical Optimization’,** Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

[17]Yang X.S., and Deb, S.:2009 ‘**Cuckoo search via Lévy flights’**, World Congress on Nature & Biologically Inspired Computing (NaBIC 2009). IEEE Publications. 2009, pp. 210–214.

[18] S. Walton; O. Hassan, K. Morgan and M.R. Brown (30 June 2011). **"Modified cuckoo search: A new gradient free optimization algorithm".** Chaos, Solitons & Fractals Vol. 44 Issue 9, Sept 2011 pp.710-718.

