

# Cauchy Movements for Artificial Bees for Finding Better Food Sources

Anguluri Rajasekhar *Student Member IEEE*  
Department of Electrical and Electronics Engineering  
National Institute of Technology-Warangal  
Andhra Pradesh-506021, INDIA  
Email: rajasekhar.anguluri@ieee.org

Millie Pant  
Department of Paper and Pulp Technology  
Indian Institute of Technology-Roorkee  
Roorkee 247667, INDIA  
Email: millifpt@iitr.ernet.in

Ajith Abraham *Senior Member IEEE*  
Faculty of Computer Science and Electrical Engineering  
VSB – Technical University of Ostrava, Czech Republic  
Machine Intelligence Research Labs (MIR Labs), USA  
Email: - ajith.abraham@ieee.org

**Abstract** — Artificial Bee Colony (ABC) algorithm is one of the most recent swarm intelligence based algorithms which simulates the foraging behavior of honey bee colonies. In the present study we propose a simple modification in the basic structure of ABC to enhance the capabilities of virtual bees in locating the potential food sources (ones having sufficient amount of nectar in them). If the virtual bees are unable to locate a good food source even after a particular number of cycles then they are forced towards another food source with the help of a small perturbation. The corresponding algorithm is named as C-ABC, where C stands for Cauchy distribution which helps the bees in getting away from food sources which are not so good. Analysis of the proposed C-ABC on 5 standard benchmark problems and 7 non-traditional problems, suggested in the special session of CEC'2008, shows that the incorporation of the proposed scheme considerably improves the performance of basic ABC algorithm.

**Keywords:**- honey-bees; Cauchy Distribution; dynamic optimization problems;

## I. INTRODUCTION

Artificial Bee Colony (ABC) is one of the most recent, swarm intelligence based algorithms employing the foraging behavior of honey bees for solving dynamic numerical optimization problems. ABC was suggested by Karaboga and Basturk for function optimization [1, 2] in 2005. Due to its simplicity and robustness it has been successfully applied to various practical optimization problems like Clustering [3]; IIR filter design [4]; extraction of MESFET [5] and so on;

However like most of the population based algorithms, ABC also has some inherent drawbacks. As pointed out by Zhu and Kwong [6], the structure of ABC is such that it supports global exploration more in comparison to the local exploitation.

Various modifications have been suggested in ABC to improve its performance. Some of which are; virtual bee algorithm (or VBA) in which the bees moves randomly in the phase space and interact by finding odd sources corresponding to the encoded values of the function [7]. Sundareswaran et al. proposed a different approach based on natural behavior of

honey bees in nectar collection in which the randomly generated worker bees are forced to move in the direction of elite bee [8]. Abraham et al. proposed hybrid ABC-DE which combines ABC with Differential Evolution [9].

In the present study we propose modifications in the basic model of ABC. If after a certain number of cycles, the virtual bees fail to find any potent food sources then they are forced to move to some other location with the hope of finding a better food source. Mathematically, the food sources represent the potential solutions and if after a certain number of generations there is no improvement in the fitness function value then there is a possibility that the algorithm is trapped in some local attractor. In order to deal with such a situation an additional mechanism is needed.

In the present study, a 'bee\_scanner' is used to identify the bees which fail to find good food sources even after a successive number of generations. Once such bees are located they are forced to move to some nearby area to locate other food sources by applying a small perturbation. This perturbation is applied using Cauchy distribution. This modification is expected to help ABC bees in escaping the local attractor and also help in faster convergence.

The remaining of the paper is organized as follows: in Section II a brief introduction to ABC is given. Section III describes the proposed C-ABC variant. In Section IV parameter settings and numerical results are given. Finally, the conclusions drawn from the present study are provided in section V

## II. BASIC ARTIFICIAL BEE COLONY

Artificial bee colony algorithm classifies the foraging artificial bees into three groups; the *employed bees*, the *onlooker bees* and the *scouts*. The first half colony consists of the employed bees and second half consist of the onlooker bees. A bee that is currently searching for food or exploiting a food source is called an employed bee and a bee waiting in the hive for making decision to choose a food source is called an

onlooker. For every food source, there is only one employed bee and employed bee of abandoned food source becomes scout. In ABC algorithm each solution to the problem is considered as food source and represented by a  $D$ -dimensional real-valued vector, whereas the fitness of the solution corresponds to the nectar amount of associated food source. Like other swarm based algorithms, ABC is also an iterative process.

The algorithm starts by initializing all the employed bees with randomly generated food sources (solutions). In each generation/iteration every employed bee finds a food source in the neighborhood of its current food source and evaluates its nectar amount i.e., (fitness). In general the position of  $i_{th}$  food source is represented as  $X_i = \{x_{i1}, x_{i2}, \dots, x_{iD}\}$ . After the information is shared by the employed bees, onlooker bees go to the region of food source at  $X_i$  based on the probability  $P_i$  defined as

$$P_i = \frac{fit_i}{\sum_{k=1}^{FS} fit_k} \quad (1)$$

$FS$  is total number of food sources. Fitness value  $fit_i$  is calculated by using following equation.

$$fit_i = \frac{1}{1 + f(X_i)} \quad (2)$$

Here  $f(X_i)$  is the objective function to be minimized. The onlooker finds its food source in the region  $X_i$ , by making use of following equation.

$$x_{new} = x_{ij} + r * (x_{ij} - x_{kj}) \quad (3)$$

Where  $k \in (1, 2, 3, \dots, FS)$  such that  $k \neq i$  and  $j \in (1, 2, 3, \dots, D)$  are randomly chosen indexes,  $r$  is a uniformly distributed random number in the range  $[-1, 1]$ .

If the obtained new fitness value is better than the fitness value achieved so far, than the bee moves to this new food source leaving this old one otherwise it retains its old food source. When all employed bees have completed this process, the information is shared with onlookers. Each of the onlookers selects food source according to probability given above. By this scheme good sources are well accommodated with onlookers than the bad ones. Each bee will search for a better food source for a certain number of cycles (*limit*), and if the fitness value doesn't improve then that particular bee becomes scout bee. The food source is initialized to that scout bee randomly.

### III. MODIFIED ARTIFICIAL BEE COLONY ALGORITHM

Artificial Bee Colony discussed above may get stuck up at local optimum which reduces the probability of obtaining global optimum because of the behavior of the virtual bees,

although scout may be employed. The performance of ABC can easily be viewed by observing the fitness function value. If there is an improvement in fitness in successive generations/iterations, it then signifies that search for the objective function is converging by improving the quality of solution. Also, it may be noted that the region in which the bees are clustered around may or may not be a local attractor basin. In such a case it is necessary to introduce a perturbation in the population which will help the bees to move to a new location.

The focus of this paper is to introduce a mechanism which will not only keep a track of the progress of bees but will also help the individuals in escaping the basin by allowing them to scatter to a new region.

In order to keep a record of the success of bees, in the present study, we introduce a concept of *bee-scanner*. The work of this scanner is to identify the bees which are unable to locate potential food sources for successive generations/iterations. If there is no improvement in fitness, then *bee-scanner* is increased by unity in each generation/iteration. This process is repeated until we achieve user-defined value of highest scan (HS). Once HS is attained, it indicates that the position of the bees should be changed. To do this Cauchy movements are applied [10] for which the probability density function (PDF) is given by the following equation:

$$f(x; x_0; \gamma) = \frac{1}{\pi \gamma \left[ 1 + \left( \frac{x - x_0}{\gamma} \right)^2 \right]} = \frac{1}{\pi} \left[ \frac{\gamma}{(x - x_0)^2 + \gamma^2} \right] \quad (4)$$

where  $x_0$  is the location parameter, specifying the location of the peak of the distribution, and  $\gamma$  is the scale parameter which specifies the half-width at half-maximum. For large values of  $\gamma$  we get a fat tail curve, whereas for smaller values of  $\gamma$ , the shape of the curve changes towards a sharper peak.

In this study we have taken  $\gamma$  as 0.1, which will produce a very sharp peak, resulting in a small area around the search space which will help the employed bee and onlooker bees in exploiting the region near the available food sources.

A new food source  $X_i = \{x_{i1}, x_{i2}, \dots, x_{iD}\}$  by C-ABC is generated as

$$x_{new} = \begin{cases} x_{ij} + r * (x_{ij} - x_{kj}) & \text{if } bee\_scanner < HS \\ x_{i,best} + C(\gamma, 0) & \text{else} \end{cases} \quad (5)$$

Where  $C(\gamma, 0)$  stands for random number generated by Cauchy probability distribution with scale parameter  $\gamma$  and centered at origin. After generation of new food source, optimization process, similar to that of basic ABC is used. This modification allows the algorithm to maintain a better balance between the exploration and exploitation factors

**Step1.** Initialize the population of solutions  $x_{ij}, i = 1, 2, \dots, FS, j = 1, 2, \dots, D, trail_i = 0$  ;

$bee\_scanner = 0$ ;  $bee\_scanner$ ; ,  $trail_i$  is the non-improvement number of the solution  $x_{ij}$  used for perturbation of Cauchy mutation via  $bee\_scanner$  and abandonment of food source via trial (limit).

**Step2.** Evaluate the population  
**Step3.** Cycle=1  
**Step4. REPEAT**  
 {----Produce new food source population for employed bee-----}  
**Step5. For  $i=1$  to FS do**  
 i. Produce a new food source  $v_i$  for the employed bee of the food source  $x_i$  by using (3) (in case of C-ABC if  $bee\_scanner_i > HS$  is reached evaluate the quality by using (5)  
 ii. Apply a greedy selection process between  $v_i$  and  $x_i$  and select the better one  
 iii. If solution  $x_i$  doesn't improve  $trail_i = trail_i + 1$  , otherwise  $trail_i = 0$  ;  $HS = HS + 1$ , otherwise  $HS_i = 0$   
**End for**  
**Step6.** Calculate the probability values  $P_i$  by (4) for the solutions using fitness values.  
 {----Produce new food source population for onlooker bee-----}  
 i.  $t=0$ ;  $i=1$ ;  
**Step7. REPEAT**  
**If  $rand < P_i$  then**  
 i. Produce a new food source  $v_i$  for the employed bee of the food source  $x_i$  by using (3) (in case of C-ABC if  $bee\_scanner_i > HS$  is reached evaluate the quality by using (5)  
 ii. Apply a greedy selection process between  $v_i$  and  $x_i$  then select the better one  
 iii. If solution  $x_i$  doesn't improve  $trail_i = trail_i + 1$  , otherwise  $trail_i = 0$  ;  $HS = HS + 1$ , otherwise  $HS_i = 0$   
 iv.  $t=t+1$   
**End if**  
**UNTIL (t=FS)**  
 {-----Determine Scout-----}  
**Step8. If  $max(trail) > limit$  then**  
 i. Replace  $x_i$  with a new randomly produce solution by using following equation  

$$x_{ij} = x_j^{min} + rand(0,1) * (x_j^{max} - x_j^{min})$$

**End If**

ii. Memorize the best solution achieved so far  
 $Cycle = Cycle + 1$ ;  
**UNTIL (Cycle=Maximum Cycle Number)**

#### IV. PARAMETER SETTINGS AND NUMERICAL RESULTS

##### A. Parameter Settings

The main parameters of basic ABC are no of Bees (NB), no of employed (onlooker) bees; *limit* for checking whether the scout should be introduced or not. For the present study the parameters considered are summarized in Table 1. for standard benchmark problems the dimension is kept 30; while for CEC2008 shifted functions the dimension is taken as 100.

All the algorithms are executed on a *Pentium-IV PC*, using MATLAB 7.6. For each benchmark 25 runs are performed. In order to have a fair comparison, the parameter settings are kept the same for all algorithms over all benchmark functions during the simulations.

In case of C-ABC the additional variable '*bee-scanner*' is assigned a limiting value of 10 i.e., if the bee is unable to find a better food source even after 10 generations then it is given a Cauchy jerk (movement) with the hope of obtaining a better value.

The scale parameter  $\gamma$  in Cauchy distribution is taken as 0.1. If the food source is further not improved then it becomes abandoned and the bee corresponding becomes scout and finds its new solution randomly.

**Table 1** Parameter Settings for ABC and C-ABC

Parameter	Value
No of bees (NB)	20
Food Sources (FS)	NB/2
Employed bees ( $n_e$ )	50 % of total bees
Onlooker bees	50 % of total bees
Scout	1
Limit	( $n_e * D$ )
Bee_scanner	10
Scale parameter $\gamma$	0.1

For non-traditional shifted problems NB is taken to 50

##### B. Comparisons and convergence graphs

The traditional test suite is given in Table II for which the number of iterations are set to 3000 and the comparisons are done based upon the termination criterion i.e., for maximum number of generations. The test suite for nontraditional problems is provided in Table IV. In case of CEC'2008 problems the comparisons are based on the number of function evaluations. A maximum of 5,00,000 Function Evaluations (NFES) are used for 100-dimensions. Following figures (1-8) show the convergence of algorithm towards minimum and the optimal values are tabulated in Tabular forms as well.

The function error value, used for performance measures for obtained solution  $x$  is defined as  $|f(x) - f(x^*)| = \epsilon$  where  $x^*$  is the global optimum of the function. The performance of C-ABC is also illustrated graphically with the help of convergence graphs where fitness function values are plotted against iterations/NFE corresponding to minimum error in all runs.

## V. CONCLUSIONS

In the present study a new variant of ABC algorithm is proposed, which aims at improving the performance of artificial bees with the help of Cauchy movements (perturbations). The performance of bees is analyzed with the help of *bee-scanner* which identifies the bees that are unable to locate a potential food source. Once such bees are identified they are forced to search their neighborhood for better food sources with the help of Cauchy distribution. This modification provides the ABC algorithm with better exploitation capabilities. Analysis of the proposed C-ABC algorithm on traditional and nontraditional test problems shows the competence of the proposed scheme.

## REFERENCES

[1] D. Karaboga, B. Basturk, A powerful and efficient Algorithm for Numerical Optimization: Artificial bee Colony (ABC) Algorithm, Journal of Global Optimization, Volume: 39, Issue:3, pp: 459-172, 2007.

[2] D.Karaboga, B.Basturk, On The Peformance of Artificial bee Colony (ABC) Algorithm, Applied Soft Computing, Volume 8, Issue 1, pp: 687-697, 2008.

[3] D. Karaboga, C. Ozturk, A Novel Clustering Approach: Artificial Bee Colony (ABC) Algorithm, Applied Soft Computing, Volume 11, Issue 1, pp: 652-657, 2011

[4] N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, Journal of the Franklin Institute, Volume 346, Issue 4, pp: 328-348

[5] Samrat Sabat, Siba K Udgata and Ajith Abraham, Artificial Bee Colony Algorithm for Small Signal Model Parameter Extraction of MESFET, Engineering Applications of Artificial Intelligence, Elsevier Science, Netherlands, 2010.

[6] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Applied Mathematics and Computation. Volume 217, Issue 7, pp: 3166-3173, 2010

[7] XS. Yang, Engineering Optimizations via Nature-Inspired Vitrual Bee Algorithms, In: Lecture notes in computer science, Springer (GHMB), pp: 317-323, 2005

[8] K. Sundareswaran, VT. Sreedevi, Development of novel optimization procedure based on honey bee foraging behaviour. In: IEEE International Conference on systems, man and cybernetics, 2008

[9] Ajith Abraham, Ravi Kumar Jatoth and A. Rajasekhar, Hybrid Differential Artificial Bee Colony Algorithm, Journal of Computational and Theoretical Nanoscience, USA, 2011.

[10] Stacey A, Jancie M, Grundy L, Particle Swarm Optimization with Mutation In: Proceeding of IEEE congress of evolutionary computation, pp: 1245-1430

**Table 2 Description of the Benchmark Functions**

Function	Mathematical Representation	Dim (D)	Range of search (S)	Theoretical optimum
Sphere ( $f_{SP}$ )	$f_1(\vec{x}) = \sum_{i=1}^D x_i^2$	30	$(-100, 100)^D$	$f_1(\vec{0}) = 0$
Rosenbrock ( $f_{RB}$ )	$f_2(\vec{x}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$(-30, 30)^D$	$f_2(\vec{1}) = 0$
Rastrigin ( $f_{RG}$ )	$f_3(\vec{x}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$(-5.12, 5.12)^D$	$f_3(\vec{0}) = 0$
Grienwank ( $f_{GW}$ )	$f_4(\vec{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$(-600, 600)^D$	$f_4(\vec{0}) = 0$
Ackley ( $f_{AK}$ )	$f_5(\vec{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	30	$(-30, 30)^D$	$f_5(\vec{0}) = 0$

**Table 3 Comparison of ABC and C-ABC in terms of fitness value, absolute error, and standard deviation for 30 runs**

Fun.	Fitness		Error		Standard Deviation	
	ABC	C-ABC	ABC	C-ABC	ABC	C-ABC
$f_{SP}$	4.76283E-16	<b>6.60599E-16</b>	6.91226E-16	6.60599E-16	1.21068E-16	1.28854E-16
$f_{RB}$	0.5153	<b>0.2265</b>	4.43774	0.954975	6.95336	2.223183
$f_{RG}$	2.07905E-06	<b>0.00E+00</b>	3.50691E-02	7.7631E-08	1.81589E-01	4.25167E-07
$f_{GW}$	1.0479E-07	<b>0.00E+00</b>	2.229352E-03	1.5211E-10	8.05307E-03	2.4974E-10
$f_{AC}$	4.1149E-12	<b>4.3521E-14</b>	3.13526E-10	4.58892E-14	1.3532E-09	7.65581E-15

**Table 4 Description of the Non-traditional Shifted Problems in CEC'2008**

Function	Name	Properties	Search space
$f_1$	Shifted Sphere	Uni-modal, separable, scalable	[-100, 100]
$f_2$	Shifted Schwefel's	Uni-modal, non-separable	[-100, 100]
$f_3$	Shifted Rosenbrock's	Multimodal, non-separable. A narrow valley from local optimum to global optimum	[-100, 100]
$f_4$	Shifted Rastrigin's	Multimodal, separable huge number of local optima	[-5, 5]
$f_5$	Shifted Griewank's	Multi-modal, non-separable	[-600, 600]
$f_6$	Shifted Ackley's	Multi-modal, separable	[-32, 32]
$f_7$	FastFractal DoubleDip	Multi-modal, non-separable	[-1,1]

**Table 5 Fitness values of 7-shifted functions for 100 dimensions using ABC**

FES ABC		$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$
1/100 FES	<i>1<sup>st</sup> Best</i>	3.1940e+005	1.4132e+002	<b>1.0161e+011</b>	1.5581e+003	2.2394e+003	2.0808e+001	-9.9803e+002
	<i>7<sup>th</sup></i>	3.5946e+005	1.5098e+002	2.4666e+011	1.7707e+003	3.0426e+003	2.0911e+001	-9.7465e+002
	<i>13<sup>th</sup> (Median)</i>	3.8799e+005	1.5428e+002	2.8158e+011	1.8032e+003	3.2660e+003	2.0950e+001	-9.5881e+002
	<i>19<sup>th</sup></i>	4.0736e+005	1.5663e+002	3.1635e+011	1.8790e+003	3.4742e+003	2.0986e+001	-9.4653e+002
	<i>25<sup>th</sup> (Worst)</i>	4.2712e+005	1.6197e+002	3.5981e+011	2.0304e+003	3.6019e+003	2.1055e+001	-9.1652e+002
	<i>Mean</i>	3.8455e+005	1.5352e+002	2.7319e+011	1.8241e+003	3.2224e+003	2.0945e+001	-9.6155e+002
	<i>Std</i>	3.1031e+004	4.7605e+000	6.0451e+010	9.9062e+001	3.0087e+002	6.1644e-002	2.2062e+001
1/10 FES	<i>1<sup>st</sup> Best</i>	2.94080e+01	1.2488e+002	7.4433e+003	2.2342e+002	1.1777e+000	1.4969e+001	-1.3544e+003
	<i>7<sup>th</sup></i>	2.62371e+02	1.3155e+002	5.5751e+004	2.5491e+002	3.1482e+000	1.6700e+001	-1.3281e+003
	<i>13<sup>th</sup> (Median)</i>	4.05785e+02	1.3532e+002	3.2540e+005	2.7141e+002	1.1387e+001	1.7154e+001	-1.3236e+003
	<i>19<sup>th</sup></i>	1.3833e+003	1.3764e+002	1.9280e+006	2.9459e+002	1.4875e+001	1.7569e+001	-1.3102e+003
	<i>25<sup>th</sup> (Worst)</i>	4.8894e+003	1.4191e+002	5.8947e+008	3.2080e+002	3.5786e+001	1.8241e+001	-1.2950e+003
	<i>Mean</i>	9.9393e+002	1.3493e+002	2.5610e+007	2.7223e+002	1.1296e+001	1.7022e+001	-1.3203e+003
	<i>Std</i>	1.1848e+003	4.5001e+000	1.1760e+008	2.5901e+001	1.0081e+001	8.5111e-001	1.5046e+001
FES	<i>1<sup>st</sup> Best</i>	3.1657e-015	6.1027e+001	0.2052e+00	3.3197e-011	2.7756e-015	1.4225e-008	-1.4766e+003
	<i>7<sup>th</sup></i>	4.2120e-015	7.3652e+001	0.5534e+00	5.5090e-009	2.2315e-014	2.8489e-008	-1.4580e+003
	<i>13<sup>th</sup> (Median)</i>	4.6756e-015	7.6246e+001	3.8669e+00	1.8838e-005	5.7843e-014	3.8056e-008	-1.4554e+003
	<i>19<sup>th</sup></i>	5.8405e-015	7.9603e+001	1.12635e+01	5.8198e-002	5.3602e-013	4.7451e-008	-1.4461e+003
	<i>25<sup>th</sup> (Worst)</i>	6.5405e-015	8.6271e+001	3.99425e+01	9.9512e-001	5.6514e-010	7.5805e-008	-1.4436e+003
	<i>Mean</i>	4.93204e-015	7.5836e+001	8.51983	0.207436	2.46765e-011	3.87588e-008	-1.4542e+003
	<i>Std</i>	9.66887e-016	5.4765e+000	10.5658	0.40272	1.12794e-010	1.44255e-008	8.0311e+000

**Table 5 Fitness values of 7-shifted functions for 100 dimensions using C-ABC**

FES (C-ABC)		$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$
1/100 FES	<i>1<sup>st</sup> Best</i>	<b>1.9677e+005</b>	<b>1.3884e+002</b>	1.0378e+011	<b>1.2582e+003</b>	<b>1.7761e+003</b>	<b>2.0530e+001</b>	<b>-1.1296e+003</b>
	<i>7<sup>th</sup></i>	2.6324e+005	1.4946e+002	1.6366e+011	1.3740e+003	2.2337e+003	2.0648e+001	-1.0870e+003
	<i>13<sup>th</sup> (Median)</i>	2.8323e+005	1.5230e+002	1.8027e+011	1.4125e+003	2.4504e+003	2.0688e+001	-1.0656e+003
	<i>19<sup>th</sup></i>	3.1055e+005	1.5606e+002	2.1485e+011	1.5541e+003	2.6191e+003	2.0731e+001	-1.0485e+003
	<i>25<sup>th</sup> (Worst)</i>	3.4942e+005	1.6310e+002	3.0570e+011	1.6185e+003	2.8461e+003	2.0753e+001	-1.0279e+003
	<i>Mean</i>	2.8355e+005	1.5192e+002	1.9454e+011	1.4482e+003	2.3898e+003	2.0677e+001	-1.0697e+003
	<i>Std</i>	3.4433e+004	5.6987e+000	5.1295e+010	1.0807e+002	2.8413e+002	6.2586e-002	2.8062e+001
1/10 FES	<i>1<sup>st</sup> Best</i>	<b>0.0084</b>	<b>1.0464e+002</b>	<b>5.04668e+02</b>	<b>6.12505e+01</b>	<b>2.2005e-002</b>	<b>4.9920e+000</b>	<b>-1.4103e+003</b>
	<i>7<sup>th</sup></i>	0.0295	1.2068e+002	7.56190e+02	8.15753e+01	1.3488e-001	6.6122e+000	-1.3927e+003
	<i>13<sup>th</sup> (Median)</i>	0.0398	1.2750e+002	1.0265e+003	8.96499e+01	1.9824e-001	7.7236e+000	-1.3866e+003
	<i>19<sup>th</sup></i>	0.0615	1.3107e+002	6.7057e+003	9.43989e+01	2.7220e-001	8.7143e+000	-1.3785e+003
	<i>25<sup>th</sup> (Worst)</i>	3.4205	1.3674e+002	1.8995e+004	1.08604e+02	4.3088e-001	1.0109e+001	-1.3679e+003
	<i>Mean</i>	0.1933	1.2551e+002	4.0678e+003	8.78576e+01	2.1612e-001	7.6278e+000	-1.3860e+003
	<i>Std</i>	0.6751	7.2676e+000	5.6987e+003	1.20771e+01	1.1819e-001	1.2994e+000	1.0731e+001
FES	<i>1<sup>st</sup> Best</i>	<b>1.8259e-015</b>	<b>4.1888e+001</b>	<b>0.15482e+00</b>	<b>0</b>	<b>0</b>	<b>1.3589e-013</b>	<b>-1.5254e+003</b>
	<i>7<sup>th</sup></i>	2.2486e-015	5.0062e+001	3.22081e+00	0	1.1102e-016	1.4655e-013	-1.5182e+003
	<i>13<sup>th</sup> (Median)</i>	2.3088e-015	5.3593e+001	6.27347e+00	0	5.5511e-016	1.5010e-013	-1.5164e+003
	<i>19<sup>th</sup></i>	2.3308e-015	5.5406e+001	8.97395e+00	1.1369e-013	5.5511e-016	1.5721e-013	-1.5143e+003
	<i>25<sup>th</sup> (Worst)</i>	2.5465e-015	6.0421e+001	2.44559e+01	1.1369e-013	9.9920e-016	1.6431e-013	-1.5088e+003
	<i>Mean</i>	2.2842e-015	5.2772e+001	7.66186	3.63798e-014	4.17444e-016	1.50955e-013	-1.5168e+003
	<i>Std</i>	1.84707e-16	4.8233e+000	6.30147	5.41258e-014	2.97697e-016	8.53638e-015	3.9102e+000

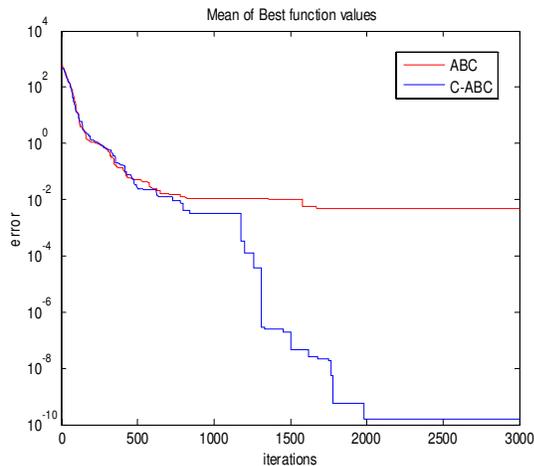


Figure1: Convergence plot of traditional Griewank function

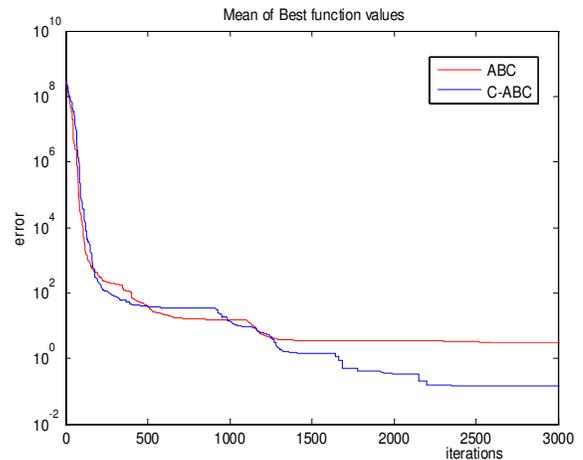


Figure2: Convergence plot of traditional Rosenbrock function