International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 6 (2014) pp. 603 -614


# BBO Algorithms with Graded Emigration for Yagi-Uda Antenna Design Optimization for Maximal Gain

**Shelja Tayal[1], Satvir Singh[2] and Gagan Sachdeva[3]**

[1] SBS State Technical Campus,
Ferozepur, Punjab [INDIA]
*sheljatayal18@gmail.com*

[2] SBS State Technical Campus,
Ferozepur, Punjab [INDIA]
*satvir15@gmail.com*

[3] Rayat Bahra Group of Institutes,
Mohali, Punjab [INDIA]
*gagan.sachdeva04@gmail.com*

*Abstract*: **Biogeography Based Optimization (BBO) is a swarm based optimization algorithm that has shown impressive performance over other Evolutionary Algorithms (EAs). Immigration Refusal Biogeography Based Optimization (IRBBO), Enhanced Biogeography Based Optimization (EBBO), Blended Migration are the most improved version of BBO and are known as migration variants of BBO. In this paper, a new concept of graded emigration for EBBO is proposed for furthur improved convergence performance. This graded emigration is also experimented on other BBO variants and found to be a competitive option. To validate the performance of Graded Emigration (GE-EBBO), experiments have been conducted on a testbed of unimodal, multimodal and deceptive benchmark test functions. Besides validation, GE-EBBO is also subjected to evolve solution to a real world problem of designing a Yagi-Uda antenna for maximal gain. Designing a Yagi-Uda antenna involves determination of wire-element lengths and their spacings in between them those bear highly complex and non-linear relationships with antenna gain, impedance, and Side Lobe Level (SLL), etc. at a particular frequency of operation. In this paper, a comparative study among BBO, EBBO, IRBBO, PSO (Particle Swarm Optimization) and GE-EBBO is conducted to analyze convergence performance while evolving the antenna designs for maximum gain, multiple times. The average of 10 monte-carlo simulations are plotted for fair quantitative comparative study of convergence performance of these stochastic algorithms.**

***Keywords***: BBO and its variants, GE-EBBO, PSO, Benchmark Functions, Yagi-Uda Antenna, Antenna Gain, NEC2.

## I. Introduction

Antenna is an electrical device which forms an interface between free-space radiations and transmitter or receiver. The choice of an antenna depends on various factors such as gain, impedance, bandwidth, frequency of operation, SLL, etc. A Yagi-Uda antenna is a widely used antenna design due to high forward gain capability, low cost and ease of construction. It is a parasitic linear array of parallel dipoles, one of which is energized directly by transmission-line while the others act as a parasitic radiators whose currents are induced by mutual coupling. The characteristics of Yagi-Uda antenna are affected by all of the geometric parameters of array.

A Yagi-Uda antenna was invented in 1926 by H. Yagi and S. Uda at Tohoku University [1] in Japan, however, published in English [2]. Since its invention, continuous efforts have been put in optimizing the antenna for gain, impedance, SLL and bandwidth using different optimization techniques based on manual, traditional mathematical approaches [3, 4, 5, 6, 7, 8, 9] and Artificial Intelligence (AI) based techniques [10, 11, 12, 13, 14, 15, 16].

Yagi aerials approximate design was proposed for maximum gain in 1949, [17]. Ehrenspeck and Poehler have given a manual approach to maximize the gain of the antenna by varying various lengths and spacings of its elements [18].

Later on, with the availability of improved computational facilities at affordable prices made it possible to optimize antennas numerically. A numerical optimization technique was proposed to calculate the maximum gain of Yagi-Uda antenna arrays with equal and unequal spacings between adjoining elements. Optimum design of Yagi-Uda antenna where antenna gain function is proved to bear a highly nonlinear relationship with its geometric parameters.

In 1975, John Holland introduced Genetic Algorithms (GAs) as a stochastic, swarm based AI technique, inspired from natural evolution of species, to optimize arbitrary systems for certain cost function. Then many researchers investigated GAs to evolve solutions to engineering problems including Yagi-Uda antenna for gain, impedance and bandwidth, separately [19, 10, 20] and collectively [11, 21, 22]. Baskar *et al.*, have optimized Yagi-Uda antenna using Comprehensive Learning Particle Swarm Optimization (CLPSO) and presented better results than other optimization techniques [13]. Li has used Differential Evolution (DE) to optimize geometrical parameters of a Yagi-Uda antenna and illustrated the capabilities of the proposed method with several Yagi-Uda antenna designs in [14]. Singh *et al*. have explored another useful stochastic global search and optimization technique named as Simulated Annealing (SA) for the optimal design of Yagi-Uda antenna [15].

In 2008, Dan Simon introduced yet another swarm based stochastic optimization technique based on science of biogeography where features sharing among various habitats (potential solutions) is accomplished with migration operator and exploration of new features is done with mutation operator

[23]. Singh *et al*. have presented BBO as a better optimization technique for Yagi-Uda antenna designs [16]. In [10, 21, 13, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34], BBO have shown comparatively better performance as compared to EAs.

In 2009, Du *et al*. have proposed the immigration refusal in BBO, where immigration from poor habitat to better habitat tend to occur [35].

In 2010, Pattnaik *et al*. have proposed EBBO in which duplicate habitats created due to migration is modified with random mutation to increase the exploitation ability of BBO. Here, experiments have been conducted on unimodal and multimodal benchmark functions. EBBO gives excellent performance when compared with BBO and other versions of BBO [36].

Ma and Simon introduced new migration operator, i.e., Blended migration, to solve the constrained optimization problem and to make BBO convergence faster [37, 38]. Firstly, Blended Crossover operator of GA outperform standard BBO on a set of benchmark optimization problems. Then, Blended BBO algorithm is compared with solutions based on a Stud Genetic Algorithm (SGA) and PSO.

In this paper, a new variant for EBBO migration, i.e., GE-EBBO is proposed to improve the performance of EBBO and other BBO variants. The proposed algorithm of GE-EBBO is applied on a testbed of benchmark functions. The results of GE-EBBO when compared with Standard BBO, IRBBO, EBBO and Blended Migration, where GE-EBBO outperformed all of them. GE-EBBO is based upon grading of habitats for migration and EBBO itself prevents similar solutions and to increase the diversity in the population. Then the proposed algorithm GE-EBBO along with other BBO variant and PSO are subjected to evolve solution to real world problem of designing Yagi-Uda to study final results and convergence performance. A method of moments based freeware programme, NEC2 (Numerical Electromagnetics Code), is used to evaluate the antenna designs for gain, input impedance, bandwidth, beamwidth and SLL, etc.

After this brief historical background, the paper is outlined as follows: In the Section II, various stochastic algorithms like Standard BBO, IRBBO, EBBO, Blended migration and PSO algorithms are discussed. It is followed by, in Section III our newly proposed algorithm, i.e., GE-EBBO. Section IV discusses about the benchmark function. In the Section V, Yagi-Uda antenna design problems are discussed. Section VI contains the comparison of performance among GE-EBBO, BBO and its variants and PSO while tested on benchmark functions and evolved solution to Yagi-Uda antenna design problem. Finally, conclusions, work in progress and research agenda have been discussed in Section VII.

## II. Stochastic Algorithms

Most of AI based EAs are stochastic in nature that uses multiple solutions at a time to evolve better solutions iteratively by imitating one or another natural phenomenon. BBO and PSO are similar EAs those have been developed by imitation of biogeography study and flocking behaviour of birds and fish, etc. These algorithms are discussed in detail as follows:

### *A. Biogeography Based Optimization*

As the name suggest, BBO is a population based global optimization technique developed on the basis of the science of biogeography, i.e., study of the distribution of animals and plants among different habitats over time and space. Originally, biogeography was studied by Alfred Wallace and Charles Darwin mainly as descriptive study [39, 40, 41]. However, in 1967, the work was carried out by MacAurthur and Wilson changed this view point and proposed a mathematical model for biogeography and made it feasible to predict the number of species in a habitat [42] under arbitrary conditions. Mathematical models of biogeography describe migration, speciation, and extinction of species in various islands. BBO has certain features common with other swarm based algorithms. Like GAs and PSO, BBO has a way of sharing information (exploitation) among solutions. GA solutions die at the end of each generation, while in PSO and BBO, solutions survive forever although their characteristics change as the optimization process progresses. PSO solutions are more likely to clump together in similar groups, while GA and BBO solutions do not necessarily have any built-in tendency to cluster. The term *island* is used for any habitat that is geographically isolated from other habitats. Habitats that are well suited residences for biological species are referred to have high Habitat Suitability Index (HSI) value. HSI is analogues to fitness in other EAs whose value depends upon many factors such as rainfall, diversity of vegetation, diversity of topographic features, land area, and temperature, etc. The factors/variables that characterize habitability are termed as Suitability Index Variables (SIVs). *Immigration* is the arrival of new species into a habitat or population, while *emigration* is the act of leaving one’s native region.The habitats with high HSI tend to have a large population of its resident species, that is responsible for more probability of emigration (emigration rate, $\mu$) and less probability of immigration (immigration rate, $\lambda$) due to natural random behavior of species. On the other hand, habitats with low HSI tend to have low emigration rate, $\mu$, due to sparse population, however, they will have high immigration rate, $\lambda$. Suitability of habitats with low HSI is likely to increase with influx of species from other habitats having high HSI. However, if HSI does not increase and remains low, species in that habitat go extinct that leads to additional immigration. For sake of simplicity, linear relationship between HSI (or population) and immigration (and emigration) rates are assumed, and maximum values of emigration and immigration rates are made equal, i.e., $E = I$, as depicted graphically in Figure 1.

For $k$-th habitat values of emigration rate, $\mu_k$, and immigration rate, $\lambda_k$, are given by (1) and (2).

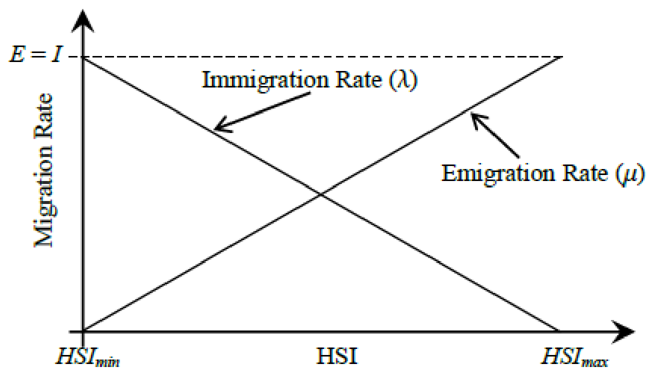


**Figure 1.** Migration Curves

$$\mu_k = E \cdot \frac{HSI_k}{HSI_{\max} - HSI_{\min}} \tag{1}$$

$$\lambda_k = I \cdot \left(1 - \frac{HSI_k}{HSI_{\max} - HSI_{\min}}\right) \tag{2}$$

The immigration of species from high HSI to low HSI habitats
may raise the HSI of poor habitats. Good solutions are more resistant to change than poor solutions whereas poor solutions are more dynamic and accept a lot of features from good solutions.

Each habitat, in a population of size *NP*, is represented by *M*-dimensional vector as $H = [SIV_1, SIV_2, \ldots SIV_M]$ where *M* is the number of SIVs (features) to be evolved for optimal HSI. HSI is the degree of acceptability that is determined by evaluating the cost/objective function, i.e., $HSI = f(H)$. Algorithmic flow of BBO involves two mechanisms, i.e., migration and mutation, these are discussed in the following subsections.

*1) Migration*

Migration is a probabilistic operator that improves HSI of poor habitats by sharing features from good habitats. During migration, *i*-th habitat, $H_i$ (where $i = 1, 2, \ldots, NP$) use its immigration rate, $\lambda_i$ given by (2), to decide probabilistically whether to immigrate or not. In case immigration is selected, then the emigrating habitat, $H_j$ , is found probabilistically based on emigration rate, $\mu_j$ given by (1). The process of migration takes place by copying values of SIVs from $H_j$ to $H_i$ at random chosen sites, i.e., $H_i$(SIV) ← $H_j$(SIV). The pseudo code of migration operator is depicted in Algorithm 1.

*2) Mutation*

Mutation is another probabilistic operator that modifies the values of some randomly selected SIVs of some habitats that are intended for exploration of search space for better solutions by increasing the biological diversity in the population. Here, higher mutation rates are investigated on habitats those are probabilistically participating less in migration process. The mutation rate, *mRate*, for *k*-th habitat is given as (3)

$$mRate_k = C \times \min(\mu_k, \lambda_k) \tag{3}$$

**Algorithm 1** Standard Pseudo Code for Migration

```
for i=1 to NPdo
  Select Hi with probability based on λi
  if Hi is selected then
   for j=1 toNPdo
     Select Hj with probability based on μj
     if Hj is selected
       Randomly select a SIV(s) from Hj
       Copy SIV(s) to Hi
      end if
   end for
  end if
end for
```

where $\mu_k$ and $\lambda_k$ are emigration and immigration rates, respectively, given by (1) and (2) corresponding to $HSI_k$. Here *C* is a constant and kept equal to 1, in this paper, i.e., mutation rate is much higher as compared to other EAs to maintain high diversity in the population. The pseudo code of mutation operator is depicted in Algorithm 2.

**Algorithm 2** Standard Pseudo Code for Mutation

```
mRate = C × min(μk, λk) where C = 1
for i = 1 to NP do
  for j = 1 to length(H) do
      Select Hj(SIV) with mRate
       If Hj(SIV) is selected then
         Replace Hj(SIV) with randomly generated SIV
      end if
  end for
end for
```

*3) BBO Algorithm*

Algorithmic flow for BBO is depicted in Figure 2, and explained stepwise as follows:

1. In first step, identify SIVs and their universe of discourse (UODs).
2. In next step, create a habitat (string).
3. Then generate a random population.
4. Check for maximum iteration number arrived or not. If yes, select the best habitat and stop the BBO algorithm. If no, then evaluate fitness.
5. Check for fitness if achieved then select the best habitat and stop the BBO algorithm. If no, then apply migration.
6. After Migration, apply mutation.
7. If fitness is achieved then select the best habitat and stop the BBO algorithm. If no, then repeat the processes as shown in Figure 2.

*B. BBO Variants*

Simple migration may leads to same values of SIVs in all habitats. To increase the diversity in the population with the objective of improvement in evolutionary performance, in BBO, different variants are investigated here. These most popular variants are explained in the following subsections:

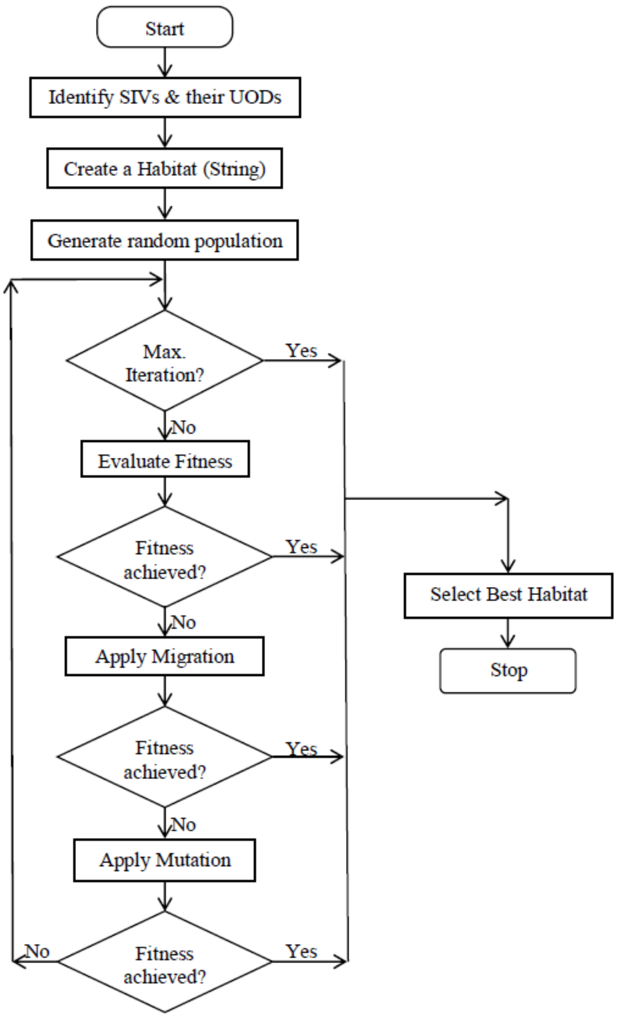


**Figure 2.** Flow Chart of BBO Algorithm

*1) Immigration Refusal*

In standard BBO, migration locations are decided on the basis of the emigration and immigration rates. If the habitat has a high emigration rate, then the probability of emigrating to other islands is high, whereas, the probability of immigration from other habitats is low. However, the low probability does not mean that immigration from low fit solution will never happen. Once in a while a high fit solution can tend to receive solution features from a low fitness solutions that may ruin the high HSI of the better habitat. So, when the SIVs from habitat which has low fitness try to emigrate to other habitats, the receiving habitats should carefully consider whether to accept these SIVs or not. If the emigration rate of the habitat is less than some threshold, and its fitness is also less than that of the immigrating habitat, then the immigrating island will refuse this migration. This idea of conditional migration is known as immigration refusal [35]. Immigration Refusal BBO variant is investigated, in this paper, for evolutionary performance here whose pseudo code is depicted in Algorithm 3.

**Algorithm 3** Pseudo Code for Immigration Refusal

```
for i = 1 to NP do
   Select H_i with probability based on λ_i
   if H_i is selected then
     for j = 1 to NP do
        Select H_j with probability based on μ_j
       if  H_j is selected
        if ((fitness(H_j)>fitness(H_i))
          apply migration
        end if
       end if
     end for
   end if
end for
```

*2) Enhanced BBO*

The exploitation ability of BBO is good as migration operator can efficiently share the SIVs between habitats. However, this creates similar habitats which decreases the diversity of the population. To increase diversity in the population so as to increase the exploration ability, clear duplicate operator is used. This variant is named as Enhanced BBO (EBBO) presented in [36], the same concept of standard migration and mutation is used. however, modified clear duplicate operator is incorporated to get better results and to make convergence faster. EBBO is investigated, in this paper, for convergence performance whose pseudo code is depicted in Algorithm 4.

**Algorithm 4** Pseudo Code for Enhanced BBO

```
for i = 1 to NP do
   Select H_i with probability based on λ_i
   if H_i is selected then
     for j = 1 to NP do
        Select H_j with probability based on μ_j
       if  H_j is selected
        if ((fitness(H_j) == fitness(H_i))
          eliminate duplicates
        end if
       end if
     end for
   end if
end for
```

*3) Blended Migration*

A new migration operator called blended migration [38], which is the modification of the standard BBO migration operator, and which is motivated from blended crossover operator of GAs. In blended crossover operator, new genes values are generated by combination of both parental gene values, instead of simple exchange of gene values. In blended migration, SIV of habitat $H_i$ is not simply replaced by SIV of habitat $H_j$ . However, a new SIV value in Blended Migration comprised of SIVs of both participating habitats, as given by 4. Blended Migration is also investigated here whose pseudo code is depicted in Algorithm 5.

$$H_i(SIV) \leftarrow \alpha . H_i(SIV) + (1-\alpha) . H_j(SIV) \tag{4}$$

**Algorithm 5** Pseudo Code for Blended Migration

```
for i = 1 to NP do
    Select H_i with probability based on λ_i
    if H_i is selected then
      for j = 1 to NP do
         Select H_j with probability based on μ_j
        if  H_j is selected
             H_i (SIV) ← α.H_i + (1-α).H_j
        end if
      end for
    end if
end for
```

Here is a real number between 0 and 1. It could be random or deterministic. In Blended BBO, exploration of search space for better solution is in built, therefore, may require less mutation rates.

*C. Particle Swarm Optimization*

PSO also belongs to the category of swarm based EAs [43] useful in solving global optimization problems. It was originally proposed by James Kennedy, as improvement in flocking behavior of birds and was introduced later as an optimization method in [44,45]. PSO implementation is easy and computationally inexpensive, since its memory and CPU speed requirements are low [46]. Moreover, it does not require gradient information of the fitness function but only its values. PSO has been proved to be an efficient method for many global optimization problems and in some cases, it does not suffer from the difficulties experienced by other EAs.
Particle swarm algorithm originated from flocking behavior of birds for getting maximum protection from predators [47]. A simulation program was developed to generate a bird flock for a hollywood film [48]. In this simulation, a point on the screen was defined as food, called the cornfield vector, the idea was to allow birds to find food through social learning by observing the behavior of nearby birds, who seemed nearer to the food source. The optimization potential was realized in the initial experiments and the algorithm was modified to incorporate topological rather than Euclidean neighborhoods and multi-dimensional search was attempted successfully.
PSO usually initializes the population by assigning each particle an arbitrary random starting position in the solution space with a randomized velocity. GAs use selection, crossover and mutation to replace less fit individuals by combining the traits of high performing chromosomes/solutions. However, in PSO, members of the particle swarm persist over time, retaining their identities and improving through imitation and interactions of best performing particles/solutions in the swarm.

*1) Flocks*

There is something about the way they move, synchronize, fly-without colliding, and resulting in amazing choreography. In 1987, a very influential simulation of bird-flock was published by Craig Reynolds [48]. Reynolds assumed that flocking birds were driven by three concerns:

1. Avoid colliding with their neighbors.
2. Match with velocities of their neighbors.
3. Try to move towards the center of the flock.

These simple rules resulted in a very realistic flocking behavior that showed coherent clusters of boids (name of simulated birds) whirling through space, splitting to flock around obstacles and rejoining again. His simple non-centralized algorithm was used in many animated cinematic sequences of flocks and herds.

*2) Schools and Social Behaviour*

In their book [43], perfectly described the rationale behind the idea that originated PSO was perfectly described as "Whenever people interact, they become more similar, as they influence and imitate one another. Norms and cultures are the result. Human physical behavior is not flock-like or school-like; the trajectories of human thoughts through high-dimensional cognitive space just might be''. The particle swarm algorithm is really a simulation of the way minds think and of human social behavior.
Regarding concordance they state, "The social phenomenon behind thinking is more complex than the choreographed behaviors of fish and birds. First, thinking takes place in belief space, whose dimensionality is far greater than three. Second, when two minds converge on the same point in cognitive space, we call it agreement, not collision''. Each time it agrees, when travels to the same position in belief space (atleast in some of the coordinates). When it disagrees, the distance in belief space increases. Imitative behavior is characterized by a velocity vector whose direction aims at another person's place in belief space.

*3) Global-Best PSO Model*

The PSO algorithm is one of stochastic swarm intelligence based global search algorithms. The motivation behind the PSO algorithm is the social behavior of birds and fish [49]. In PSO, the particles have (1) adaptable *velocities* that determines their movement in the search space, (2) *memory* which enable them for remembering the best position in the search space ever visited. The position corresponding to the previous best fitness is known as past best, *pbest* and the overall best out of all *NP* the particles in the population is called global best, *gbest*. Consider that the search space is *M*-dimensional and *i*-th particle in the swarm can be represented by $X_i = [x_{i1}, x_{i2}, \ldots.x_{id}\ldots, x_{iM}]$ and its velocity can be represented by another *M*-dimensional vector $V_i = [v_{i1}, v_{i2}, \ldots.v_{id}.., v_{iM}]$. Let the best previously visited position of *i*-th particle be denoted by $P_i = [p_{i1}, p_{i2}, \ldots.p_{id}.., p_{iM}]$, whereas, *g*-th particle, i.e., $P_g = [p_{g1}, p_{g2}, \ldots.p_{gd}.., p_{gM}]$, is globally best particle. Figure 3 depicts the vector movement of particle element from location $x_{id}^n$ to $x_{id}^{n+1}$ in $(n + 1)$-th iteration that is being governed by past best, $p_{id}^n$ , global best, $p_{gd}^n$ , locations and current velocity $v_{id}^n$ . Figure 4 depicts the flowchart of PSO Algorithm and discussed as follows:

1. Initialize the population of particles at random positions and velocities. Assign present location and fitness as $p_{id}$ and *pbest* to every particle as starting position and fitness, respectively.

2. For each particle, evaluate its fitness at the present position, $x_i$.
3. Compare the particle's fitness with *pbest*. If the current fitness value is better, copy it to *pbest* and set $p_{id}$ equal to the current position, $x_{id}$.
4. Identify the most successful particle in the swarm and store it as $p_{gd}$.
5. Update the velocity and position of the particle using equations 5 and 6 [44]:

$$v_{id}^{m+1} = \chi(wv_{id}^{m} + \psi_1 r_1 (p_{id}^{m} - x_{id}^{m}) + \psi_2 r_2 (p_{gd}^{m} - x_{id}^{m})) \quad (5)$$

$$x_{id}^{m+1} = x_{id}^{m} + v_{id}^{m+1} \quad (6)$$

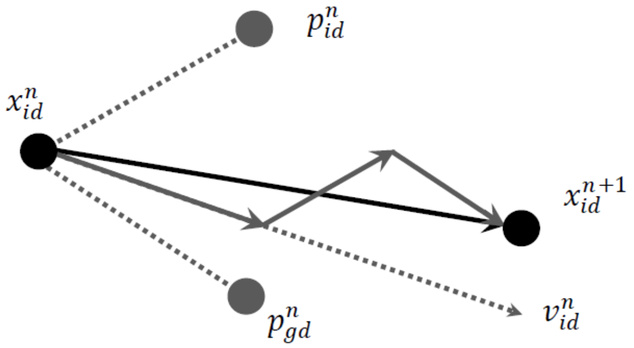


**Figure 3.** Movement of *i*-th particle in 2-dimensional search space

Here, $w$ is inertia weight, $\psi_1$ is cognitive parameter, $\psi_2$ is social parameter and constriction factor $\chi$ are strategy parameters of PSO algorithm, while $r_1$, $r_2$ are random numbers uniformly distributed in the range [0,1]. Generally the inertia weight, $w$, is not kept fixed and is varied as the algorithm progresses. The particle movements is restricted with maximum velocity, $\pm V_{max}$, to avoid jump over the optimal location in the search space.

*4) PSO Characterization*

There are several parameters that need to be defined in order to successfully utilize PSO to solve a given problem [50]

**Solution Encoding**: It is a *M*-dimensional vector representation of collection problem feature to be evolved for desired fitness function. This usually involves a minimum and a maximum value allowed in each dimension, thus defining a hyperspace.

**Fitness Function**: This function is degree of suitability/accepatibility also problem dependent and represents a measurement of a given solution. The function should somehow create a total ranking in the solution.

**Population Size**: This parameter infuences the behavior of the algorithm. A very small population does not create enough interaction for the emergent behavior pertaining to PSO to occur. However, large population size may lead to more computational burden and consequently, take more evolutionary time. So the population size is to be decided as per the problem size and complexity.

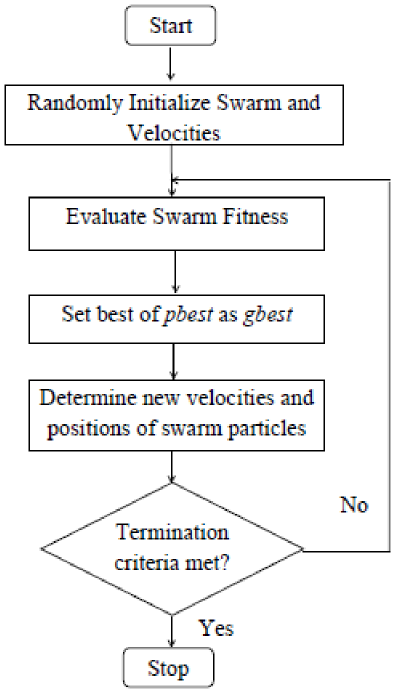


**Figure 4.** Flow Chart of PSO Algorithm

**Acceleration Coefficients**: The acceleration coefficients $\psi_1$ and $\psi_2$ are usually set to the same value. Infact, people usually talk about which sets the other two values $\psi_1 = \psi_2 = \psi/2$. If is too small, the maximum step size becomes quite small and so the algorithm will explore very slowly and degrade its performance. There is a consensus among the researchers that step size is generally optimal if $\psi = 4.1$, however, not for every problem and every time.

**Constriction or Inertia coefficient**: It is not necessary to guess its value as given by equation 7. If the value of $\psi$ is set to 4.1, then $\chi \approx 0.729$.

$$\chi = \frac{2k}{2 - \psi - \sqrt{\psi^2 - 4\psi}} \quad (7)$$

$$where\ k = [0,1],\ \psi = \psi_1 + \psi_2,\ \psi > 4$$

**Maximum Velocity**: With the advent of the constriction coefficient, most researchers do not bother using this parameter. However, to avoid jump overs maximum velocity is fixed to some value less than unity.

**Neighborhood Topology**: If every particle is made to interact with every other in the swarm, then it becomes prone to fall into local optima. However, this may be avoided if swarm is divided into subgroups and every particle is made to interact with all members of its subgroup.

## III. Graded Emigration

In standard migration and its other variants do decide emigrating and immigrating habitats and their SIVs probabilistically. Graded Emigration (GE) is a new migration variant introduced in this paper, where number of SIVs of each emigrating habitat and their SIV number are predecided where to migrate in accordance to with their fitness ranking. In GE the poorest habitat is completely replaced and the best habitat is preserved as it is, whereas the mediocre habitats are partially modified by sharing fixed number of SIVs from better habitats. The number of migrating SIVs are fixed, however their location is decided randomly.

**Example III.1** (Graded Emigration among 10 habitats having 10 SIVs in each habitat). *For Graded Emigration in a population of 10 habitats having 10 SIVs in each habitat following steps are required to be followed:*

*1. Sort habitats in ascending order to their fitness values.*

*2. The last poor habitat constitute a new habitat in the ratio of 4:3:2:1 to replace the poorest in the population.*

*3. Next to the poorest is contributed by 90% by first, second, third and fourth best habitats in the 4:3:2:0.*

*4. Subsequently, the other poorer habitats partially modified by the better habitats as per the matrix given in the Algorithm 6.*

For 20 or 30 habitats the algorithm may be extended by doubling or triplicating the rows of the matrix $X$. Furthur modified clear duplicate opertor is integrated here to increase the exploration ability and thereafter named as GE-EBBO.

## IV. Benchmark Functions

There are many benchmark functions [50, 44] which are commonly used to critically test the performance of numeric optimization algorithms. These functions are chosen because of their particularities, which render their optimization difficulties. These comprise (a) multi-modality (b) deceptive gradient information (c) the curse of dimensionality. There are many benchmark test functions like a few of them listed in Table 1 and used in this paper to validate and compare the concept of GE with other variants.

| Function Name | Function ($f(x)$) |
|---|---|
| Dejong/Sphere | $\sum_{i=1}^{n} x_i^2$ |
| Ackley | $20+e-20e^{-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}}-e^{\frac{\sum_{i=1}^{n}\cos 2\pi x_i}{n}}$ |
| Griewank | $1+\frac{\sum_{i=1}^{n}(x_i-100)^2}{4000}-\prod_{i=1}^{n}\frac{\cos(x_i-100)}{\sqrt{i}}$ |
| Rastrigin | $\sum_{i=1}^{n} x_i^2-10\cos 2\pi x_i+10$ |
| Rosenbrock | $\sum_{i=1}^{n-1} 100(x_{i+1}-x_i^2)^2+(x_i-1)^2$ |

*Table 1.* Benchmark Testbed for various Stochastic Algorithms.

Dejong/Sphere function is very simple and any algorithm capable of numeric optimization should solve it without any problem. It is unimodal function, with global minima located at $x = (0, \ldots, 0)$, with $f(x) = 0$.

Ackley is a multimodal function with many local optima, however global minimum is $f(x) = 0$, is located at $x = (0, \ldots, 0)$. This function is difficult because optimization algorithms can easily be trapped in a local minima on it's way to the global minimization.

Griewank function is strongly multimodal function with significant interaction among its variables, caused by the product term. This function has the interesting property that the number of local minima increases with dimensionality. The
global minimum, $x = (100, 100, \ldots, 100)$, yields a function value $f(x) = 0$.

Rastrigin is a multimodal version of the spherical function, characterized by deep local minima arranged as sinusoidal bumps. The global minimum $f(x) = 0$, is located at $x = (0, \ldots, 0)$.

Rosenbrock function variables are strongly dependent and gradient information often misleads algorithms. It's global minimum of $f(x) = 0$ is located at $x = (1, \ldots, 1)$.

**Algorithm 6** Standard Pseudo Code for GE-EBBO

$$X[i][j] = \begin{array}{cl} \left[\begin{array}{cccc} 4 & 3 & 2 & 1 \\ 4 & 3 & 2 & 0 \\ 4 & 3 & 1 & 0 \\ 4 & 3 & 0 & 0 \\ 4 & 2 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{array}\right] & \begin{array}{l} \text{Poor Habitat} \\ \\ \\ \\ \\ \\ \\ \\ \\ \text{Best Habitat} \end{array} \end{array}$$

**for** *i to NP* **do**
**for** $j$=1 to $X[i][1]$ **do**
Randomly Select SIV(s) from *NP-th* Habitat and copy to random SIV($s$) in *i-th* Habitat
**end for**
**for** $j$=5 to $X[i][1]+X[i][2]$ **do**
Randomly Select SIV(s) from *(NP-1)-th* Habitat and copy to random SIV($s$) in *i-th* Habitat
**end for**
**for** $j$=8 to $X[i][1]+X[i][2]+X[i][3]$ **do**
Randomly Select SIV(s) from *(NP-2)-th* Habitat and copy to random SIV($s$) in *i-th* Habitat
**end for**
**for** $j$=10 to $X[i][1]+X[i][2]+X[i][3]+X[i][4]$ **do**
Randomly Select SIV(s) from *(NP-3)-th* Habitat and copy to random SIV($s$) in *i-th* Habitat
**end for**
**end for**

## V. Antenna Design Parameters

Yagi-Uda antenna consists of three types of wire elements: (a) *Reflector*–biggest among all and is responsible for blocking radiations in one direction. The reflector element is about 5 percent is longer than the feeder element. There is typically only one reflector, adding more reflectors improves performance very slightly. This element is important in determining the front-to-back ratio of the antenna. (b) *Feeder*–which is fed with the signal from transmission line to be transmitted. There is usually just one driven element. A dipole driven element will be resonant when its electrical length is half of the wavelength of the frequency applied to its feed point. and (c) *Directors*–these are usually more then one in number and responsible for unidirectional radiations. The lengths of directors reduces in the direction of radiations and depends upon the director spacing, the number of directors used in the antenna, the desired pattern, pattern bandwidth and element diameter. Figure 5 depicts a typical six-wire Yagi-Uda antenna where all wires are placed parallel to *x*-axis and along

*y*-axis. Middle segment of the reflector element is placed at origin, $x = y = z = 0$, and excitation is applied to the middle segment of the feeder element. An incoming field sets up resonant currents on all the antenna elements which reradiate signals. These re-radiated fields are then picked up by the feeder element, that leads to total current induced in the feeder equivalent to combination of the direct field input and the re-radiated contributions from the director and reflector elements.

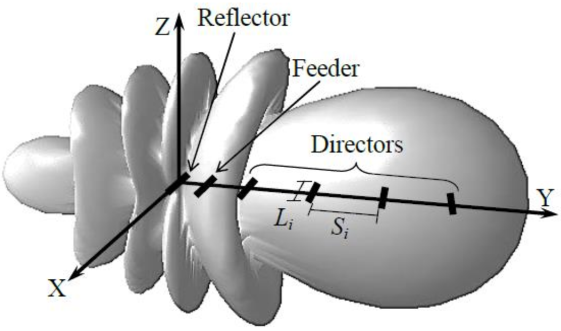


**Figure 5.** Radiation Pattern of atypical 6-wire Yagi-Uda Antenna

The radiation or antenna pattern describes the relative strength of radiated field in various directions from the antenna at a constant distance. The radiation pattern is also called reception pattern as well, since it also describes the receiving properties of the antenna. The radiation pattern is three-dimensional, however, usually the measured radiation patterns are a two dimensional slice of the three-dimensional pattern in the horizontal and vertical planes. These pattern measurements are presented in either a rectangular or a polar format. A polar format of the gain versus orientation (radiation pattern) is useful when characterizing antennas. Some other important features of antenna that appears on plot are:

**1. Forward Gain**: Forward gain is the ability of an antenna to focus energy in a particular direction while transmitting/receiving energy better to/from a particular direction. To determine the gain or directivity of an antenna, a reference antenna is used to compare antenna performance. Forward gain is expressed in decibel (dB) relative to an isotropic source or a standard dipole in direction of maximum gain. Typically, higher the gain, more the efficient antenna performance and longer the range of the antenna will operate. Radiation pattern of a typical six-elements Yagi-Uda antenna is depicted in Figure 5.
**2. Front to Back ratio**: The Front to Back ratio is used in describing directional radiation patterns for antennas. If an antenna radiates maximum in one direction, the F/B ratio is the ratio of the gain in the maximum direction to that in the opposite direction (180 degrees from the specified maximum direction) ans is also expressed in dB.
**3. Beamwidth**: Beamwidth is the angle between directions where the power is half the value at the direction of maximum gain which is -3dB. It gives the measure of directivity of antenna
**4. Sidelobes**: Antenna is not able to radiate all the energy in one preferred direction because some part of energy is inevitably radiated in other directions. Sidelobes are unwanted peaks in the gain at angles other than in forward direction, they reduce the amount of useful energy contained in the forward direction. The peaks are referred to as side lobes, as shown in Figure 5, and commonly specified in dB down from the main lobe.

Other characteristics that do not appear on the polar plot but which are equally important are:
**1. Bandwidth**: Bandwidth is the range of frequency over which the antenna exhibits acceptable characteristics.
**2. Radiative impedance**: For an efficient transfer of energy, the radiative impedance of the antenna and transmission cable connecting them must be the same. Transceivers and their transmission lines are typically designed for 50Ω resistive impedance. If the antenna has an impedance different from 50Ω then there is a mismatch and an impedance matching circuit is required to avoid signal loss.
Designing a Yagi-Uda antenna involves determination of wire-lengths and wire-spacings in between to get maximum gain, desired impedance and minimum SLL at an arbitrary frequency of operation. An antenna with N elements requires $2N - 1$ parameters, i.e., $N$ wire lengths and $N - 1$ spacings, that are to be determined. These $2N - 1$ parameters, collectively, are represented as a string vector referred as a habitat in BBO or particle in PSO given as (8).

$$H = [L_1, L_2, \ldots, L_N, S_1, S_2, \ldots, S_{N-1}] \tag{8}$$

where $L_s$ are the lengths and $S_s$ are the spacing of antenna elements.

## VI. Simulation Results and Discussions

Simulation results by testing with benchmark functions and its convergence performance on Yagi-Uda antenna is explained in the following subsections:

### *A. Testing with Benchmark Functions*

A testbed of benchmark functions is used to test the proposed algorithm of GE-EBBO and compared with the results of other migration variants of BBO are tabulated in Table 2.

To provide similar platform for comparitive study parameters used are:
1. Population size: 20
2. Number of SIV's: 10
3. Search space of f(x) : $-2 \le x \le 2$
4. Number of Iterations: 10000
5. Number of Monte-Carlo simulations per experiment: 10

In Figure 6, 7 GE-EBBO optimizes faster as compared to other stochastic algorithms. EBBO gives poor results followed by IRBBO, BBO and Blended.
In Figure 8, GE-EBBO and EBBO gives almost same results. Initially, EBBO converges faster. But with increase in iterations, GE-EBBO performs almost equal to EBBO.

In Figure 9, Initially, GE-EBBO converges slowly than other algorithms. But with increase in number of iterations, GEEBBO performs better than all other algorithms.

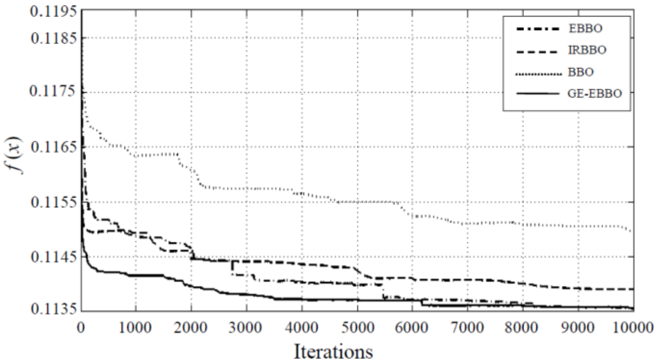


**Figure 6.** Performance of various Stochastic Algorithms using Ackley

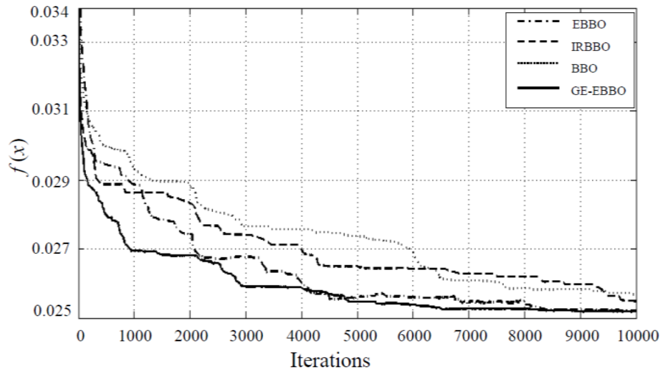


**Figure 7.** Performance of various Stochastic Algorithms using Dejong

In Figure 10, At initial stage GE-EBBO does not perform upto mark but with increase in iterations at 3000-4000 the performance increases and approaches almost to the performance of IRBBO which gives best results.

Figures 6, 7, 8, 9 concluded that GE-EBBO outperforms all other algorithms. Average of 10 Monte Carlo simulations runs is depicted in Table 2. Here the minimization approach has been taken by taking the overall average of all test functions.

*B. Convergence Performance for Yagi-Uda Antenna*

Six-wire Yagi-Uda antenna designs are optimized for gain using BBO, EBBO, PSO and GE-EBBO algorithms are investigated. Average of 10 monte-carlo evolutionary runs for each algorithm are plotted here for investigation in Figure 11. The C++ programming platform is used for coding of optimization algorithms, whereas, a method of moments based software named as Numerical Electromagnetics Code (NEC) [51] is used for evaluation of antenna designs. Each potential solution in BBO is encoded as vector with 11 SIVs as given by equation (8) and shown in Figure 5. The universe of discourse for the search of optimum values of wire lengths and wire spacings are fixed at 0.40λ - 0.50λ and 0.10λ - 0.45λ, respectively, however, cross sectional radius and segment sizes are kept same for all elements, i.e., 0.003397λ and 0.1λ respectively, where λ is the wavelength corresponding to frequency of operation, i.e, 300 MHz. Excitation is applied to the middle segment of driven element and location of middle segment of the reflector element is always kept at $x = 0$.

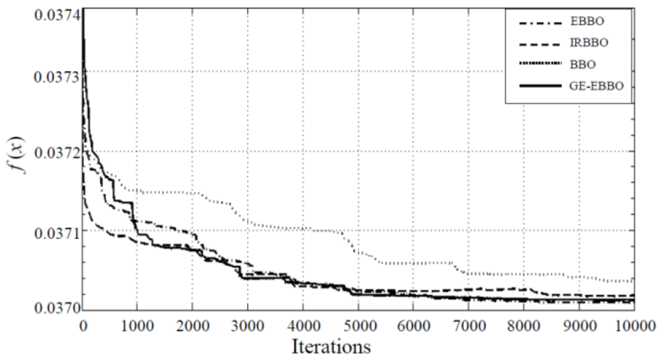


**Figure 8.** Performance of various Stochastic Algorithms using Griewank

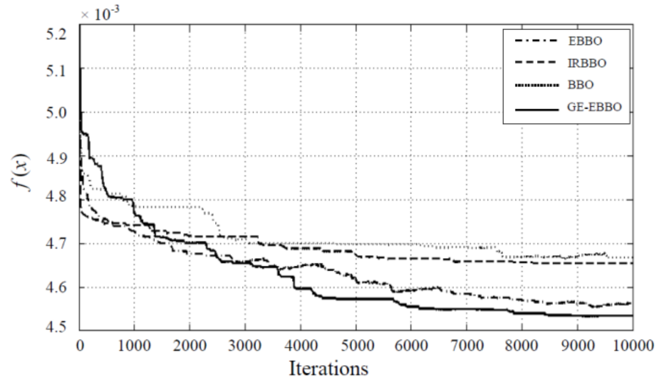


**Figure 9.** Performance of various Stochastic Algorithms using Rastrigin

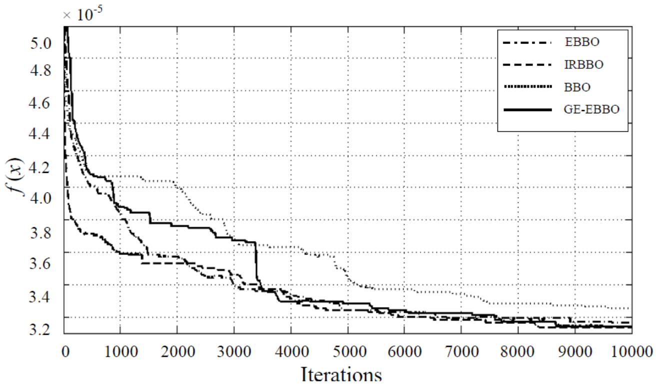


**Figure 10.** Performance of various Stochastic Algorithms using Rosenbrock

| Test Functions | BBO | IRBBO | EBBO | Blended | GE-EBBO |
|---|---|---|---|---|---|
| **Ackley** | 0.1149567 | 0.1139057 | 0.1135618 | 0.1211875 | **0.1135563** |
| **Dejong** | 0.0256786 | 0.0254857 | 0.0252104 | 0.0286977 | **0.0251785** |
| **Griewank** | 0.0370556 | 0.0370368 | **0.0370287** | 0.0371918 | 0.0370322 |
| **Rastrigin** | 0.0046668 | 0.0046538 | 0.0045540 | 0.0052622 | **0.0045334** |
| **Rosenbrock** | 0.0000325 | **0.0000313** | 0.0000316 | 0.0000414 | 0.0000314 |
| **Overall Average** | **0.0364780** | **0.0362226** | **0.0360773** | **0.0384761** | **0.0360663** |

*Table 2.* Result obtained by testing on Benchmark Testbed

The parameters used in experiments are:

1. No. of Habitats or population: $NP = 30$
2. Generations: 50
3. No. of SIVs per habitat: 11
4. Mutation range: ±1%
5. Maximum migration rates $E = 1$ and $I = 1$

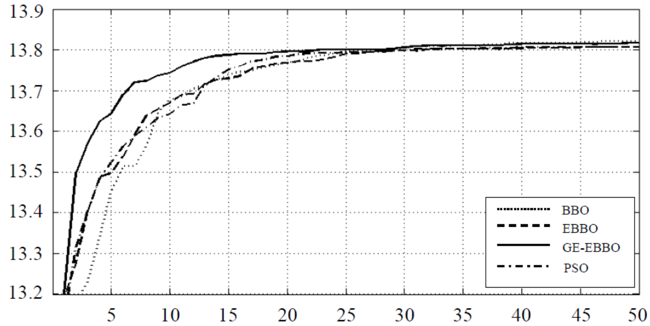


**Figure 11.** Convergence Performance of BBO, EBBO, GE-EBBO and PSO

Here, mutation range is the percentage of SIV's that adds to the randomly selected SIVs to increase the exploration of search space. Typically, the best antenna designs obtained during process of optimization are tabulated in Table 3. Figure 11 concludes that GE-EBBO is performing best among all optimization algorithms. Though EBBO converges faster than BBO, but performs in comparison to PSO.

The reason for GE-EBBO's best performance is the exploitation on less fit habitats, whereas less exploitation on high fit habitats and then the modified clear duplicate operator is furthur integrated to increase the diversity.

| Stochastic algorithms | Element | 1(λ) | 2(λ) | 3(λ) | 4(λ) | 5(λ) | 6(λ) | Gain(dBi) | Z(Ω) |
|---|---|---|---|---|---|---|---|---|---|
| **BBO** | Length | 0.4875 | 0.4884 | 0.4400 | 0.4233 | 0.4217 | 0.4233 | **13.84** | **4.09+j54.53** |
| | Spacing | - | 0.1503 | 0.2571 | 0.4087 | 0.3932 | 0.4095 | | |
| **EBBO** | Length | 0.4827 | 0.4735 | 0.4424 | 0.4259 | 0.4201 | 0.4256 | **13.83** | **5.23+j31.59** |
| | Spacing | - | 0.2255 | 0.2160 | 0.3889 | 0.4181 | 0.3911 | | |
| **GE-EBBO** | Length | 0.4842 | 0.4910 | 0.4425 | 0.4253 | 0.4181 | 0.4270 | **13.84** | **4.47+j59.63** |
| | Spacing | - | 0.1778 | 0.2381 | 0.3918 | 0.4212 | 0.3870 | | |
| **PSO** | Length | 0.4872 | 0.4944 | 0.4423 | 0.4272 | 0.4194 | 0.4276 | **13.85** | **3.83+j62.16** |
| | spacing | - | 0.1597 | 0.2420 | 0.3857 | 0.4190 | 0.3841 | | |

*Table 3.* The Best Results obtained during Gain Optimization

## VII. Conclusions and Future Scope

In this paper, Experimental Analysis shows that GE-EBBO gives comparitively better results than various stochastic algorithms discussed here, when applied on testbed of benchmark functions and to optimize six-element Yagi-Uda antenna designs for gain maximization. GE-EBBO solves the global optimization problem with faster convergence rate because of high exploitation and performs better. Reasons for poor performance of PSO include use of global best PSO model where each particle learns from every other and *gbest*. This may lead to be trapped into local optima. Our future agenda is to apply GE-EBBO on various real time applications

and to investigate it's performance by influencing the population size and search space to get better results. It can also be explored for multi-objective optimization of different antenna as well.

## References


[1] S. Uda and Y. Mushiake, *Yagi-Uda Antenna*. Research Institute of Electrical Communication, Tohoku University, 1954.

[2] H. Yagi, "Beam Transmission of Ultra Short Waves," *Proceedings of the Institute of Radio Engineers*, vol. 16, no. 6, pp. 715–740, 1928.

[3] D. G. Reid, "The Gain of an Idealized Yagi Array," *Journal of the Institution of Electrical Engineers-Part IIIA: Radiolocation*,, vol. 93, no. 3, pp. 564–566, 1946.

[4] J. Bojsen, H. Schjaer-Jacobsen, E. Nilsson, and J. Bach Andersen, "Maximum Gain of Yagi–Uda Arrays," *Electronics Letters*, vol. 7, no. 18, pp. 531–532, 1971.

[5] D. K. Cheng, "Optimization Techniques for Antenna Arrays," *Proceedings of the IEEE*, vol. 59, no. 12, pp. 1664–1674, 1971.

[6] L. C. Shen, "Directivity and Bandwidth of Single-band and Double-band Yagi Arrays," *IEEE Transactions on Antennas and Propagation*,, vol. 20, no. 6, pp. 778–780, 1972.

[7] D. Cheng and C. Chen, "Optimum Element Spacings for Yagi-Uda Arrays," *IEEE Transactions on Antennas and Propagation*,, vol. 21, no. 5, pp. 615–623, 1973.

[8] C. Chen and D. Cheng, "Optimum Element Lengths for Yagi-Uda Arrays," *IEEE Transactions on Antennas and Propagation*,, vol. 23, no. 1, pp. 8–15, 1975.

[9] D. K. Cheng, "Gain Optimization for Yagi-Uda Arrays," Antennas and Propagation Magazine, IEEE, vol. 33, no. 3, pp. 42–46, 1991.

[10] E. A. Jones and W. T. Joines, "Design of Yagi-Uda Antennas using Genetic Algorithms," *IEEE Transactions on Antennas and Propagation*,, vol. 45, no. 9, pp. 1386–1392, 1997.

[11] H. J. Wang, K. F. Man, C. H. Chan, and K. M. Luk, "Optimization of Yagi array by Hierarchical Genetic Algorithms," *IEEE*, vol. 1, pp. 91–94, 2003.

[12] N. Venkatarayalu and T. Ray, "Optimum Design of Yagi-Uda Antennas Using Computational Intelligence," *IEEE Transactions on Antennas and Propagation*,, vol. 52, no. 7, pp. 1811–1818, 2004.

[13] S. Baskar, A. Alphones, P. N. Suganthan, and J. J. Liang, "Design of Yagi-Uda Antennas using Comprehensive Learning Particle Swarm Optimisation," *IEEE*, vol. 152, no. 5, pp. 340–346, 2005.

[14] J. Y. Li, "Optimizing Design of Antenna using Differential Evolution," *IEEE*, vol. 1, pp. 1–4, 2007.

[15] U. Singh, M. Rattan, N. Singh, and M. S. Patterh, "Design of a Yagi-Uda Antenna by Simulated Annealing for Gain, Impedance and FBR," *IEEE*, vol. 1, pp. 974–979, 2007.

[16] U. Singh, H. Kumar, and T. S. Kamal, "Design of Yagi-Uda Antenna Using Biogeography Based Optimization," *IEEE Transactions on Antennas and Propagation*,, vol. 58, no. 10, pp. 3375–3379, 2010.

[17] R. M. Fishenden and E. R. Wiblin, "Design of Yagi Aerials," *Proceedings of the IEE-Part III: Radio and Communication Engineering*, vol. 96, no. 39, p. 5, 1949.

[18] H. Ehrenspeck and H. Poehler, "A New Method for Obtaining Maximum Gain from Yagi Antennas," *IRE*

*Transactions on Antennas and Propagation,*, vol. 7, no. 4, pp. 379–386, 1959.

[19] E. Altshuler and D. Linden, “Wire-antenna Designs using Genetic Algorithms,” *Antennas and Propagation Magazine, IEEE*, vol. 39, no. 2, pp. 33–43, 1997.

[20] D. Correia, A. J. M. Soares, and M. A. B. Terada, “Optimization of gain, impedance and bandwidth in Yagi-Uda Antennas using Genetic Algorithm,” *IEEE*, vol. 1, pp. 41–44, 1999.

[21] N. V. Venkatarayalu and T. Ray, “Single and Multi-Objective Design of Yagi-Uda Antennas using Computational Intelligence,” *IEEE*, vol. 2, pp. 1237–1242, 2003.

[22] Y. Kuwahara, “Multiobjective Optimization Design of Yagi-Uda Antenna,” *IEEE Transactions on Antennas and Propagation,*, vol. 53, no. 6, pp. 1984–1992, 2005.

[23] D. Simon, “Biogeography-based Optimization,” *IEEE Transactions on Evolutionary Computation,*, vol. 12, no. 6, pp. 702–713, 2008.

[24] M. Rattan, M. S. Patterh, and B. S. Sohi, “Optimization of Yagi-Uda Antenna using Simulated Annealing,” *Journal of Electromagnetic Waves and Applications*, 22, vol. 2, no. 3, pp. 291–299, 2008.

[25] S. Singh, S. Tayal, and G. Sachdeva, “Evolutionary performance of bbo and pso algorithms for yagi-uda antenna design optimization,” *IEEE*, pp. 861–865, 2012.

[26] S. Singh and G. Sachdeva, “Mutation effects on bbo evolution in optimizing yagi-uda antenna design*,” in Third International Conference on Emerging Applications of Information Technology (EAIT 2012)*, Kolkata, India, November-December 2012.

[27] ——, “Yagi-uda antenna design optimization for maximum gain using different bbo migration variants,” International Journal of Computer Applications, vol. 58, no. 5, pp. 1–5, 2012.

[28] S. Singh, E. Mittal, and G. Sachdeva, “Nsbbo for gain impedance optimization of yagi-uda antenna design,” *in Information and Communication Technologies (WICT)*, 2012 World Congress on. IEEE, 2012, pp. 856–860.

[29] S. Singh, Shivangna, and S. Tayal, “Analysis of different ranges for wireless sensor node localization using pso and bbo and its variants,” *International Journal of Computer Applications*, vol. 63, no. 22, pp. 31–37, February 2013, published by Foundation of Computer Science, New York, USA.

[30] S. Singh, Shivangna, and E. Mittal, “Range based wireless sensor node localization using pso and bbo and its variants,” 2013.

[31] S. Singh, E. Mittal, and S. Tayal, “Evolutionary performance comparison of bbo and pso variants for yagi-uda antenna gain maximization,” in *National Conference on Contemporary Techniques & Technologies in Electronics Engineering*, 2013.

[32] S. Singh, Shivangna, and E. Mittal, “Performance of pso with different ranges for wireless sensor node localization,” in *National Conference on Contemporary Techniques & Technologies in Electronics Engineering*, Murthal, Sonepat, India, March 2013.

[33] S. Singh, S. Tayal, E. Mittal, and Shivangna, “Evolutionary performance of graded emigration in bbo for yagi-uda antenna design optimization,” *CiiT International Journal of Programmable Device Circuit and Systems*, April 2013, ciiT International Journal

[34] S. Singh, E. Mittal, and G. Sachdeva, “Multi-objective gain-impedance optimization of yagi-uda antenna using nsbbo and nspso,” *International Journal of Computer Applications*, vol. 56, no. 15, pp. 1–6, October 2012, published by Foundation of Computer Science, New York, USA.

[35] D. Du, D. Simon, and M. Ergezer, “Biogeography based Optimization Combined with Evolutionary Strategy and Immigration Refusal,” *IEEE*, vol. 1, pp. 997–1002, 2009.

[36] S. S. Pattnaik, M. R. Lohokare, and S. Devi, “Enhanced Biogeography-Based Optimization using Modified Clear Duplicate Operator,” *IEEE*, vol. 1, pp. 715–720, 2010.

[37] H. Ma and D. Simon, “Biogeography-based optimization with blended migration for constrained optimization problems,” *in Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, 2010, pp. 417–418.

[38] ——, “Blended Biogeography-based Optimization for Constrained Optimization,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 3, pp. 517–525, 2011.

[39] C. Darwin, “On the origins of species by means of natural selection,” *London: Murray*, 1859.

[40] A. R. Wallace, *The geographical distribution of animals: With a study of the relations of living and extinct faunas as elucidating the past changes of the earth's surface*. Cambridge University Press, 1876, vol. 1.

[41] C. Darwin, *On the Origin of Species: A Facsimile.* Harvard University Press, 1964.

[42] R. MacArthur and E.Wilson, *The Theory of Island Biogeography*. Princeton Univ Pr, 1967.

[43] R. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. Morgan Kaufmann Publisher, 2001.

[44] Y. Shi and R. Eberhart, “Empirical Study of Particle Swarm Optimization,” in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 3. IEEE, 1999.

[45] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4. IEEE, 1995, pp. 1942–1948.

[46] Y. Shi et al., “Particle Swarm Optimization: Developments, Applications and Resources,” *in Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 1. IEEE, 2001, pp. 81–86.

[47] F. Heppner and U. Grenander, “A stochastic nonlinear model for coordinated bird flocks.” *AMERICAN ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE, WASHINGTON, DC(USA)*. 1990., 1990.

[48] C. W. Reynolds, “Flocks, herds and schools: *A distributed behavioral model*,” vol. 21, no. 4, pp. 25–34, 1987.

[49] K. Parsopoulos and M. Vrahatis, “Recent Approaches to Global Optimization Problems through Particle Swarm Optimization,” *Natural computing*, vol. 1, no. 2, pp. 235–306, 2002.

[50] R. Mendes, “Population topologies and their influence in particle swarm performance,” 2004.

[51] G. J. Burke and A. J. Poggio, “Numerical Electromagnetics Code (NEC) method of moments,” *NOSC Tech. DocLawrence Livermore National*

*Laboratory, Livermore, Calif*, USA, vol. 116, pp. 1–131, 1981.

## Author Biographies

**Shelja Tayal** was born on Oct 14, 1989. She received her Bachelor's degree (B.Tech.) from Lala Lajpat Rai Institute of Engineering & Technology (Moga) in year 2011 and presently she is a research scholar in Shaheed Bhagat Singh State Technical Campus (formerly, SBS College of Engineering & Technology), Ferozepur, Punjab (India) with specialization in Electronics & Communication Engineering. She has published and communicated many papers in International/National Journals and Conferences. Her research interests include Artificial Intelligence, Evolutionary Algorithms and Antenna design optimization.

**Dr. Satvir Singh** was born on Dec 7, 1975. He received his Bachelors degree (B.Tech.) from Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, Punjab (India) with specialization in Electronics & Communication Engineering in year 1998, Masters degree (M.E.) from Delhi Technological University (Formerly, Delhi College of Engineering), Delhi (India) with distinction in Electronics & Communication Engineering in year 2000 and Doctoral degree (Ph.D.) from Maharshi Dayanand University, Rohtak, Haryana (India) in year 2011. During his 13 years of teaching experience he served as Assistant Professor and Head, Department of Electronics & Communication Engineering at BRCM College of Engineering & Technology, Bahal, (Bhiwani) Haryana, India and as Associate Professor & Head, Department of Electronics & Communication Engineering at Shaheed Bhagat Singh State Technical Campus (Formerly, SBS College of Engineering & Technology), Ferozepur Punjab, India.
His fields of special interest include Evolutionary Algorithms, High Performance Computing, Type-1 & Type-2 Fuzzy Logic Systems, Wireless Sensor Networks and Artificial Neural Networks for solving engineering problems. He is active member of an editorial board of International Journal of Electronics Engineering and published nearly 30 research papers in International Journals and Conferences. He has delivered nearly 20 Invited Talks during National and International Conferences, Seminar, Short Term Courses and Workshops. He completed two AICTE funded projects under MODROB Scheme worth 15 Lacs. He has also conducted four Faculty Development Programmes, of total duration of six weeks, around Soft Computing techniques under various schemes of AICTE and TEQIP.

**Gagan Sachdeva** was born on Oct 4, 1988. He received his Bachelors degree (B.Tech.) and Masters degree (M.Tech.) from Shaheed Bhagat Singh State Technical Campus (formerly, SBS College of Engineering & Technology), Ferozepur, Punjab (India) with specialization in Electronics & Communication Engineering in year 2010 and 2012, respectively. He is presently with Rayat Bahra Group of Institutes, Mohali Campus as Assistant Professor in the department of ECE. His research interests include Evolutionary Algorithms, Antenna design optimization, and Wireless Sensor Networks.