HYBRID BIOGEOGRAPHY BASED OPTIMIZATION ALGORITHM FOR OPTIMIZATION PROBLEMS

Khug Alam, Wali Khan & Muhammad Asim

Department of Mathematics, Kohat University of Science & Technology, Pakistan.

ABSTRACT

In the last few years and so, Evolutionary Computation (EC) has become a focusing area for research due to the wide application of various developed evolutionary algorithms (EAs) for dealing with different types of optimization and search problem. Biogeography Based Optimization (BBO) is one of the recently newly and efficiently population based techniques. It uses a set of uniformly and randomly generated solutions and optimizes them in order to get a set of optimized solutions in a single simulation run unlike traditional optimization methods. BBO is mainly shares information between species of migration from one island to another island based mathematical model to perform their search process. In this article, Differential Evolution (DE) has been employed in combination with BBO algorithm and as a result we have developed a new hybrid version of BBO called HBBO. Performance of HBBO is examined by 2005 IEEE Conference on Evolutionary Computation (CEC'05) test suite. The suggested algorithm has efficiently tackled most of the test problems as compared to BBO algorithm.

Keywords: *Optimization, Evolutionary Computation, Evolutionary Algorithms & BBO*

INTRODUCTION

Optimization is the mathematical process to find the optimal value of the optimization and search problem with continuous, discrete, integer or mixed of these variables. In general, optimization problem involves an objective function subject to some constraint functions described as follows:

$$\begin{aligned} &Minimize \quad f(x) \quad Subject\ to \\ &g_j(x) \leq 0, \qquad j = 1,2,\dots p, \\ &h_j(x) = 0\ ,\ j = 1,2,..q \end{aligned} \qquad (1)$$

Where $x = [x_1, x_2, \dots, x_n]^T \in R^n$ is an n-dimensional vector of optimization/decision variable? p is the inequality while q is the number of equality constraints respectively. Moreover, $L_i \leq x_i \leq U_i$, $i = 1,2,..n$, L_i and U_i are the lower and upper bounds of parametric space S respectively, and the function $f(x)$ is called an objective/fitness function. Optimal solutions are solutions in which objective function completely optimized like problem (Wolpert & Macready, 2005). Evolution is the basic strategy of the two-step process of random variation & selection. Mathematically it could be represented as follow:

$$x[t+1] = s\big(v(x[t])\big) \qquad (2)$$

Where $x[t]$ and $x[t+1]$ are the previous and current populations obtained through the execution of selection (s) and variation (v) operators. Evolutionary algorithms (EAs) use a set of solutions, usually generated uniformly and randomly called a population. EAs employ diverse evolutionary operator to perturb their population for finding approximated set of optimal solutions in a single simulation run unlike traditional optimization techniques. They are inspired by natural evolution and successful operations are attributed to different intrinsic search operators and well configured settings of parameters.

Since the development of first evolutionary algorithms (EAs) (Goldberg, 1989) vary, many types of EAs have been proposed in existing literature of EC. EAs do not demand for any derivative information regarding problems at hand. They have strong ability of handling different types of optimization and real world problems. In this paper, we have injected different mutation strategies of DE in BBO (Simon, 2008) framework and developed its hybrid version denoted by HBBO with aim to further improve the search abilities of the baseline BBO to cope with benchmark functions developed for the special session of the 2005 conference of the evolutionary computation (CEC05) (Suganthan, Hansen, Liang, Deb, Chen, Auger & Tiwari, 2005). The paper is organized in different sections. Section 2 presents basic information about the biogeography based optimization algorithm. Section 3 provides hybrid biogeography based evolutionary algorithm and four diverse differential evolution mutation strategies. Section 4 devoted to the numerical results of the CEC05 benchmark functions. Section 5 contains conclusion with some future work plan.

BIOGEOGRAPHY BASED ALGORITHM

Biogeography Based Optimization (BBO) was first developed by Simon (2008). Biogeography shows us how the objects change their places due to the environmental changes with time and how the objects get extinct. Those areas which are more suitable for objects called high habitat suitability index (HSI). The features which are affected by species such as land area, rainfall, temperature and many other issues are called suitability index variables (SIVs). HSI are the dependent variables and SIVs are independent variables. The areas whose HSI is high have a quality to attract more and more objects and areas whose HSI low can attract only number of objects. High HSI immigrate many objects to their neighboring area because the population of that area is so populated. Emigration occurs because it is affected a number of objects within population. When objects emigrate from an island, only a few number of representatives objects emigrate.

BBO simulates the immigration of objects in multidimensional area; each island describes a candidate solution of the optimization problem (Wolpert & Macready, 2005). Areas whose HSI high their emigration rate would be high, but immigration rate is low. Those objects which migrate to a high HSI island, they die because they compete for resources with other objects. Areas whose HSI is low their immigration rate would be high because the population of that island is low. Objects want to emigrate from that island because that island is an unpleasant place for living. The rate of immigration is high to these islands because there is more additional place available for new objects. However, when the objects arrive to the area who's HIS is low, the HSI of the area is increased.

Algorithm 1 The framework of the BBO algorithm

```
generate the popsize
for i ← 1 : popsize do
    μ = (popsize − i)/popzize
    λ=1-μ
    if rand < 0.4 then
        for k ← 1 : popsize do
            Select μ
            Randomly select an SIV from X_i
            Replace a random SIV in X_k
        end for
    end if
end for
```

The pseudocode of the original BBO as outlined above in the Algorithm1 is given as.

- First, define the island modification probability and mutation. These steps are the same as in Genetic Algorithm.
- The population is initialized.
- The symbols λ and μ represents immigration and emigration rates respectively. λ and μ are calculated for each population. The best solution has high emigration and low immigration rate.
- The selection criteria are based on the immigration rates.
- Randomly the migrated SIVs based on the selected population.
- Apply mutation probability on each population.
- The best values are calculated for each individual population.
- If the criteria of termination does not meet, then go to step 2.

Hybrid Biogeography Based Evolutionary Algorithm

The proposed algorithm has been used BBO in combination with different mutation strategies of differential evolution and as resultant hybrid BBO algorithm developed. A mutation strategy DE/rand/1 has employed in BBO algorithm as additional algorithms probabilistically aiming to improve the search abilities of original BBO over the test problems designed for the special session of Conference on Evolutionary Computation (CEC'05) (Suganthan et al., 2005). Framework of HBBO Algorithm is hereby outlined in the Algorithm 2.

Algorithm 2. Framework of the HBBO

```
Initialize the popsize
for i ← 1 : popsize do
    μ = (popsize − i)/popsize
    λ=1-μ
    Select uniformly randomly r1 ≠ r2 ≠ r3 ≠ i
    j_rand = rand(1 : n)
    for j ← 1 : n do
        if rand(0, 1) < λ_i then
            CR = 0.5
            F = 0.5
            if rand_j(0, 1) < CR then
                U_i(j) = X_r1(j) + F.(X_r2(j) − X_r3(j))
            else
                Select X_p with probability ∝ μ_p
                U_i(j) = X_p(j)
            end if
        else
            U_i(j) = X_i(j)
        end if
    end for
end for
```

Differential Evolution (DE)

DE is another powerful EA introduced by Storn & Price (1996). It came from the idea of using vector differences for perturbing the vector population (Storn & Price, (1995), (Storn & Price, 1997). Adaptation scheme introduced in the DE framework (Zhang & Sanderson, 2008; Montes et al., 2006; Shah et al., 2016; Wazir et at., 2016 and Khanum, Nasser, Jan, Mashwani & Salhi, 2016) to further improve their convergence properties over both complicated constrained and nonlinear optimization problems. The interested readers are referred to (Mashwani (2014) for more detail. Mutation, crossover and selection are important operators of DE framework for generation and selection of solutions for the upcoming generation. Population sizeNP, mutation factor$F_m$and crossover ratio Cr are their important intrinsic parameters. To maintain genetic diversity from generation to generation is called mutation. In every generation g of DE, a mutant vector, $V_{i,g}$ of the current population, $x_{i,g\ ,i=1,2,....,pop}$ is designed by using one by one from the following strategies which are listed in literature (Mezura, Reyes & Coello, 2006).

1. **DE/rand/1**
   $V_{i,g}=x_{r1,g} + F_m(x_{r2,g} - x_{r3,g})$
2. **DE/best/1:**
   $V_{i,g}=x_{best,g} + F_m(x_{r2,g} - x_{r3,g})$
3. **DE/rand-to-best/1:**
   $V_{i,g}=x_{r1,g} + x_{best,g} + F_m(x_{r2,g} - x_{r3,g})+ F_m(x_{best,g} - x_{r3,g})+$
4. **DE/current-to-best/1:**

$$V_{i,g}=x_{r1,g} + x_{best,g} + F_m(x_{r2,g} - x_{r3,g})+ F_m(x_{best,g} - x_{i,g})+$$

Where $x_{r2,g} - x_{r3,g}$ is a difference variation vector with respect to the current best and i[th] individual $x_{i,g}$, $x_{best,g}$ of the current generation and the values of the scaling factor F_m $(0, +1)$.

Benchmarks Functions and Discussion

In the study of this paper, we have used 20 benchmark functions in order examine the algorithmic behavior of the suggested algorithm HBBO. The results of these functions are listed in the form of best, worst, mean, Standard deviation and Average CPU time. The details regarding the used CEC'5 benchmark functions can be found in (Suganthan et al., 2005).

Table 1: The Properties of Benchmark Functions of Evolutionary Computation

f1	Shifted Sphere Function	[-100,100]	2,10
f2	Shifted Schwefel's Problem 1.2	[-100,100]	2,10
f3	Shifted Rotated High Conditioned Elliptic Function	[-100,100]	2,10
f4	Shifted Schwefel's Problem 1.2 with Noise in Fitness	[-100,100]	2,10
f5	Schwefel's Problem 2.6 with Global Optimum on Bounds	[-100,100]	2,10
f6	Shifted Rosenbrock's Function	[-100,100]	2,10
f7	Shifted Rotated Griewank's Function without Bounds	[0, 600]	2,10
f8	Shifted Rotated Ackley's Function with Global Optimum on Bounds	[-32,32]	2,10
f9	Shifted Rastrigin's Function	[-5,5]	2,10
f10	Shifted Rotated Rastrigin's Function	[-5,5]	2,10
f11	Shifted Rotated Weierstrass Function	[-0.5,0.5]	2,10
f12	Schwefel's Problem 2.13	[-π,π]	2,10
f13	Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)	[-5,5]	2,10
f14	Shifted Rotated Expanded Schaffer's F6	[-100,100]	2,10
f15	Hybrid Composition Function	[-5,5]	2,10
f16	Rotated Hybrid Composition Function	[-5,5]	2,10
f17	Rotated Hybrid Composition Function with Noise in Fitness	[-5,5]	2,10
f18	Rotated Hybrid Composition Function	[-5,5]	2,10
f19	Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum	[-5,5]	2,10
f20	Rotated Hybrid Composition Function with the Global Optimum on the Bounds	[-5,5]	2,10
f21	Rotated Hybrid Composition Function	[-5,5]	2,10
f22	Rotated Hybrid Composition Function with High Condition Number Matrix	[-5,5]	2,10
f23	Non-Continuous Rotated Hybrid Composition Function	[-5,5]	2,10
f24	Rotated Hybrid Composition Function	[-5,5]	2,10

RESULTS AND DISCUSSIONS

In this paper, we have carried out our experiments using the platform as under:

- Operating system: Windows XP Professional;
- Programming language : Matlab;
- CPU: Core 2 Quad 2.4 GHz;
- RAM: 4 GB DDR2 1066 MHz.

The Parameters were settled to perform our experiment are given as follow:

- $NP = 100$, the population size;
- $F\,ES = n \times NP$, FES means Function Evaluations;
- $n = 5, 10, 30$, the dimension of the search/decision/parametric space;
- $F = 0.4$, the scaling factor of the DE parameter;
- $Cr = 1/n$, the probability of the use of binomial crossover;

The Experimental results regarding $n = 10$ are given in Table 2 and figure 1 while Table 3 presents comparison of both algorithms HBBO and BBO. Table 2 presents the experimental results of the HBBO algorithm in terms of best, worst, mean, Standard deviation and Average CPU time and Table 3 provides comparison of HBBO and BBO in terms of best and mean during the 25 independent executions with different seeds. Figure 1 shows the variation in the average function values in twenty five independent runs on each CEC'05 (Suganthan et al., 2005) test with search space dimension $n = 10$. The results provided by suggested algorithm both in numerical and graphical form for each CEC '05 test function are more promising in term of proximity as shown in Table 3.

Table 2. Numerical Results for the CEC'05 Problems

Problems	best	mean	St. dev.	Worst	Avg CPU Time (s)
f01	0.000000	0.000000	0.000000	0.000000	262.137611
f02	0.000079	1.394698	0.001059	0.005634	272.826619
f03	23336.798	582735.049	297065.292	1734523.96	570.136425
f04	0.000658	5.124856	0.013340	0.068959	843.323107
f05	0.000000	0.000557	0.000000	0.000000	1144.032475
06	3.038268	5.277536	0.002765	4.742417	242.412855
f07	0.000000	0.000000	0.000000	0.000000	0.000000
f08	20.210415	20.789490	0.014353	20.987086	523.443369
f09	0.000000	0.000000	0.000000	0.000000	0.000000
f10	11.417355	50.738798	5.798386	60.406202	297.211704
f11	5.604508	11.591520	0.080878	12.980332	1098.352873
f12	100.016495	2741.799178	1089.071407	7733.491735	1523.299708
f13	0.000000	0.000000	0.000000	0.000000	0.000000
f14	2.826769	4.246931	0.019029	4.402642	283.118051
f15	0.000000	0.000000	0.015802	58.201975	2069.046251
f16	128.031967	211.850454	19.927218	232.775737	3791.802968
f17	0.000000	0.000000	0.000000	0.000000	0.000000
f18	300.000000	300.000026	0.000000	300.000000	1903.238542
f19	0.000000	0.000000	0.000000	0.000000	0.000000
f20	300.000000	300.000013	0.000000	300.000000	1895.641406

Table 3: The Comparison of HBBO VS BBO over f01-f10

problems	Best	Mean	Algorithm
f01	**0.00000**	**0.00000**	HBBO
	0.110732	1.592177	BBO
f02	**0.000079**	**1.394698**	HBBO
	134.566826	2957.740893	BBO
f03	**263336.798739**	**582735.049465**	HBBO
	493363.693005	19137468.889277	BBO
f04	**0.000658**	**5.124856**	HBBO
	881.722169	7412.329508	BBO
f05	**0.000000**	**0.000557**	HBBO
	194.726285	636.706292	BBO
f06	**3.038268**	**5.277536**	HBBO
	93.769585	2691.244344	BBO
f07	0.00000	0.000000	HBBO
	0.000000	0.000000	BBO
f08	**20.210415**	**20.789490**	HBBO
	20.242844	20.872887	BBO
f09	**0.000000**	**0.000000**	HBBO
	0.037797	0.652922	BBO
f10	**11.417355**	**50.738798**	HBBO
	30.186811	70.082813	BBO

Figure 1. Convergence graphs of CEC'05 (Suganthan, 2005) displayed by HBBO

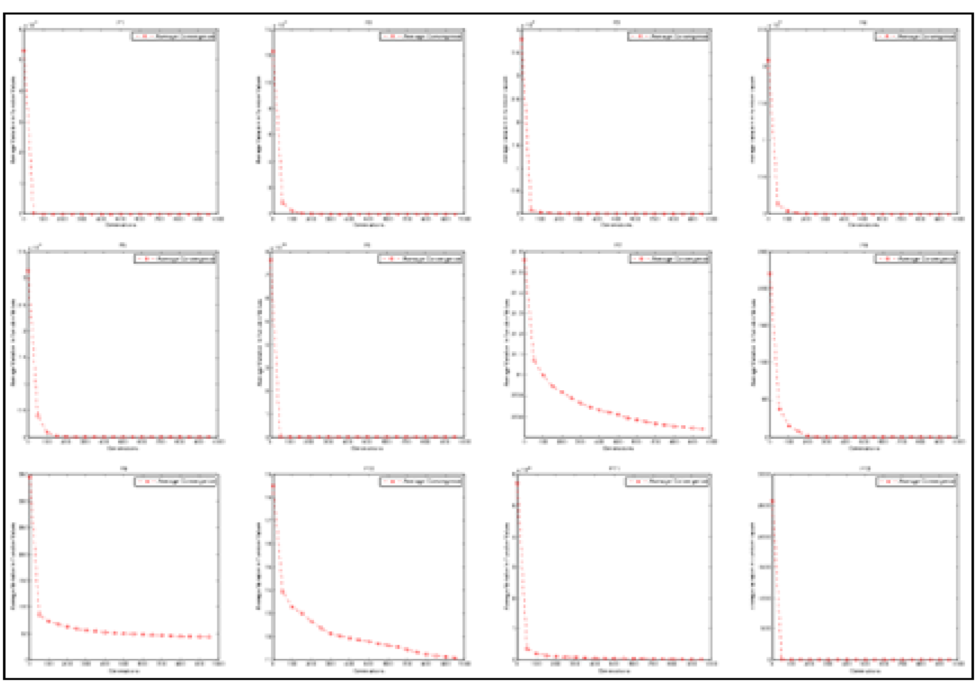
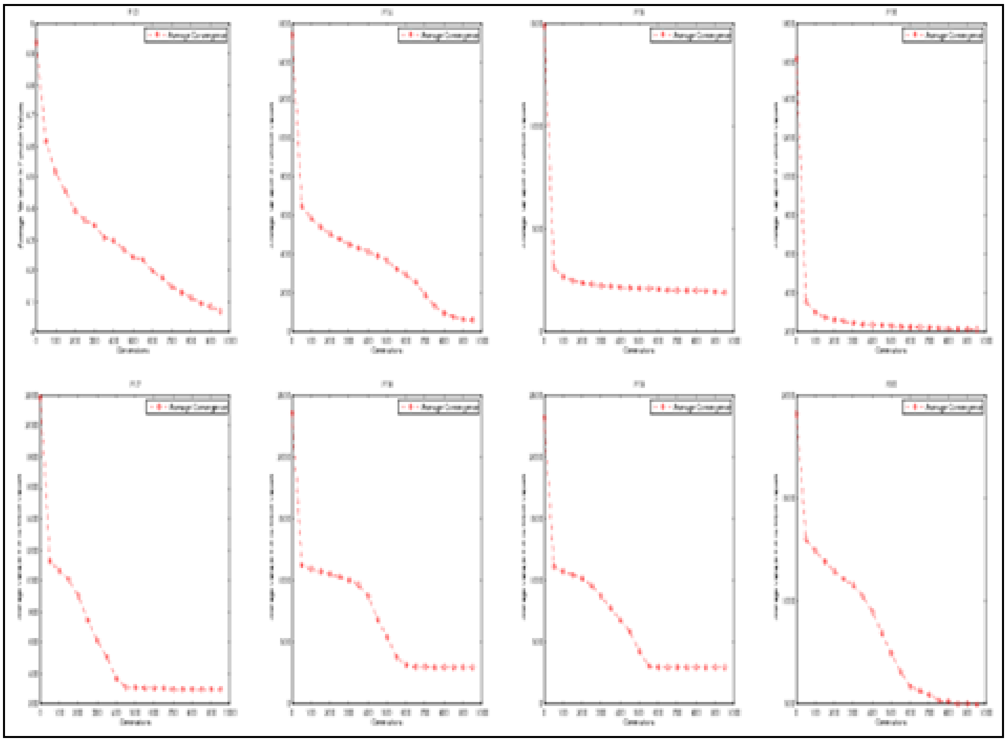


Figure 2. Convergence graphs of CEC'05 (Suganthan, 2005) displayed by HBBO



CONCLUSION

In this paper, modified version of BBO has been suggested in which DE has been employed as an extra search operator for the purpose to improve the search ability of the baseline BBO algorithm. We have injected different mutation strategies of DE in the framework of the BBO and as a resultant we developed hybrid version of BBO denoted by HBBO. In this paper, we have summarized the experimental results by employing DE/rand/1 strategy in combination with BBO as a global search technique. The results provided by the suggested algorithm indicated that HBBO have tackled most of the used test problems with fast convergence and better accuracy in single objective optimization parlance. We also intend to examine the performance of the suggested proposal over some latest test suites of the IEEE CEC series in order to judge the ability and credibility of the suggested algorithms in our future plan.

References

Alam, K., Mashwani, W. K., Asad, M., & Asim, M. (2016). Biogeography Based Algorithm for Optimization Problems: Preliminary Experimental Results 3rd Conference on Sustainability in Process Industry 19-20th October, 2016.

Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Pub. Co.

Khanum, R. A., Islam, Z., Jan, M. A., & Mashwani, W. K. (2016). Reproductive Nelder-Mead Algorithm for Unconstrained Optimization Problems. *Science International Journal*, 7(2).

Khanum, R.A., Tairan, N., Jan, M.A., Mashwani, W. K., & Salhi, A. (2016). Reflected Adaptive Differential Evolution with Two External Archives for Large-Scale Global Optimization. *International Journal of Advanced Computer Science and Applications*, 7(2), 675-683.

Mashwani, W. K. (2014). Enhanced versions of Differential Evolution: State-of-the-art Survey. *International Journal Computing Sciences and Mathematics*, 5(2), 107-126.

Montes, E., Reyes, J., & Coello, C. A. (2006). Comparative Study of Differential evolution variants for Global Optimization, in the Proceeding of the Genetic Evolutionary Computation Conference (*GECCO*), *Seattle*, 485-492.

Shah, T., Asif, M. J., Mashwani, M. K., & Wazir, H. (2016). 'Adaptive Differential Evolution for Constrained Optimization Problems. *Science International*, 28 (3), 2313-2320.

Simon, D. (2008). Biogeography-Based Optimization. *IEEE Transactions on Evolutionary Computation*, 12, 702-713.

Storn, R., & Price, K. (1995). Differential Evolution, A Simple and efficient Adaptive Scheme for Global Optimization over Continuous Spaces. *International Computer Science Institute*, BerkeleyTech. Rep. TR-95-012.

Storn, R., & Price, K. V. (1997). Differential evolution, a simple and efficient heuristic for Global optimization over continuous Spaces. *Journal of Global Optimization*, 11, 341-359.

Wazir, H., Asif, J. M., Mashwani, W. K., & Shah, T. (2016). A Penalty Function Based Differential Evolution Algorithm for Constrained Optimization. The Nucleus Journal, 53 (1), 155-161.

Wolpert, D., & Macready, H. (2005). Co evolutionary Free Lunches. *IEEE Transactions on Evolutionary Computation*, 9 (6), 721- 735.

Zhang, J., & Sanderson, A. C. (2008). Adaptive Differential Evolution with Optional External Archive, in Proc, IEEE Congress on. Evolutionary. Computation.