Comparative Study of Teaching-Learning-Based Optimization, Artificial Bee Colony and Differential Evolution on Numerical Benchmark Problems

Himansu Bhushan Mohapatra[1] and Mrutyunjay Rout[2]

Department of Electronics & Communication Engineering

National Institute of Technology Jamshedpur, Jamshedpur, India

***E-mail:-** hbmece@gmail.com[1], mrutyunjay.ece@nitjsr.ac.in[2]*

DOI: - https://doi.org/10.47531/MANTECH/ECC.2021.16

Abstract

Metaheuristic modern optimization techniques are Superior to the traditional optimization techniques to find the global optimization solution. These techniques are verified as efficient for optimization in almost all fields, but still, they have some restrictions in one or another feature. Due to this, more and more research is required to check the algorithms for different problems to verify their suitability for the numerical problem. In this paper, performances of Teaching-Learning-based optimization, Artificial Bee Colony and Differential Evolution algorithms were compared on general benchmark problems. The effect of problem dimensional and control parameters on the performance of the algorithms was verified, and the algorithms were measured in terms of mean, the standard deviation of the fitness function values.

***Keywords—** Meta-heuristic optimization, teaching-Learning- Based Optimization, Artificial Bee Colony Optimization, Differential Evolution and Benchmark functions*

INTRODUCTION

Optimization is a technique to compute the best value of a problem under certain boundaries. Sometimes, we obtained a satisfactory value of the problem under certain conditions in place of the best value or the optimum value. Optimization does not give a new solution to a problem, but it works on the set of solutions available and gives the best among them. It will check all the solutions available with some rules, verify the conditions given and suggest the best one under that condition [1]. It may happen that if the condition changes, then the best suggestion no longer are the best one.

It was verified that one method is not sufficient to solve all types of problems. Due to this, a number of different optimization techniques are developed to satisfy the requirement and solve real-world problems. [2], [3] All these techniques are centre around global optimization of mathematical, real value black-box problems for which exact and analytical techniques don't apply. These optimization algorithms are broadly classified as evolutionary algorithms and swarm intelligence-based algorithms. The genetic algorithm works on the principle of the Darwinian hypothesis of the survival of the fittest and the theory of evolution of the living creatures. Genetic Programming [4], Evolution programming [5], Evolution strategies [7], Differential evolution (DE) [9] which is similar to a genetic algorithm with specialized, unique crossover and selection method are the example of some evolutionary optimization technique. Swarm intelligence-based optimization techniques are Particle swarm optimization (PSO) which is working on the foraging behaviour of birds, Artificial Bee Colony, Ant colony optimization, which is working on the foraging behaviour of the ant for the food etc.

In this paper, we compared the Teaching-Learning based optimization (TLBO), Artificial Bee Colony (ABC) and Differential Evolution (DE) algorithms on a diverse set of problems. We investigated the performance of TLBO, ABC and DE on a selected number of numerical benchmarks. The main idea was to study and test whether each of the tested algorithms would do better than others on most functions.

The paper is prearranged as follows: In part II, we discuss the methods used for this purpose. The simulation details, the parameter used and benchmark function considered for this purpose are outlined in section III. The simulation results and their interpretation are given in section IV. Finally in section V gives a discussion on the simulation solution.

METHODS

A. Teaching learning-based optimization

TLBO was proposed by Rao in 2011. It is a population-based method that works on the effect of the influence of a teacher in the class and on the output of learners in a class [1],[7]. This idea claims that it can obtain the global solutions for continuous non-linear function with less computational effort and high consistency. It works on the philosophy of the Teaching and Learning process adopted in a class. Here the output is considered in terms of results or grades [8].

The optimization algorithm comprises two stages, a teacher phase and a learner phase. The teacher phase implies gaining from the teacher, and the learner phase means learning through the connection between learners. In both phases, a new solution is generated and updated the population, and at the end of the termination criteria, the algorithm gives its best value. The following equation is for the generation of a new solution.

$$DifferenceMean_i = r_i(M_{new} - T_F M_i) \quad (1)$$

Here TF is the teaching factor that decides the value of the mean to be changed, and ri is a random number in the range [0,1]. Mnew and Mi mean are the new and existing means, respectively. This difference modifies the existing solution and generates the new solution according to the following expression.

$$X_{newi} = X_{oldi} + DifferenceMean_i \quad (2)$$

In the Learner Phase, the learner increases the knowledge through input from the teacher and interaction between themselves. Here the learner learns something new if the other learner has more knowledge than him. To update the solution here, the algorithm chooses a different learner randomly and uses the following expression.

$$X_{newi} = X_{oldi} + r_i(X_i - X_j) \quad (3)$$

Suppose the new solution is better than the existing one. The main strength of this algorithm is that it does not require any parameter setting for the working of the algorithm.

B. Artificial bee colony

ABC is an optimization technique developed by Karaboga. In ABC, the solutions are the food sources of the bees. Employed Bees, Onlooker Bees and Scout Bees are the main components of ABC. Employed bees search and memories the food source in the surroundings of their hive. Onlooker bees gather the data from employed bees and choose food sources for further extraction. If the nectar measure in food sources is low or worn out, the scout bee arbitrarily finds another food source in search space. The set of the mathematical equation used for these methods are listed below:

$$v_{id} = pha_{id}(x_{id} - x_{kd}) + x_{id} \quad (4)$$

Here fitness si is the fitness of the ith solution. If a solution cannot revise itself for a particular running cycle, then it is measured as a discarded solution, and the equivalent bee becomes the scout. In ABC, the control parameter used for the algorithm after a particular solution is considered exhausted and known as a limit.

$$prob_i(G) = \frac{0.9 * fitness_i}{\max fitt} + 0.1 \quad (5)$$

Here fitness si is the fitness of the ith solution. If for a particular running cycle, a solution is not capable to revise itself then it is measured as discarded solution and the equivalent bee becomes the scout. In ABC, the control parameter used for the algorithm after which a particular solution is considered exhausted is known as limit.

C. Differential evolution

Differential evolution is a direct search population-based global optimization method [9], [10]. Mutation, crossover and selection are the basic operations performed by the DE method.

Mutation operation generates a new vector by adding the weighted difference between two population vectors to a third vector. The following expression is used to generate a mutated vector.

$$V = X_{r_1} + F(X_{r_2} - X_{r_3}) \quad (6)$$

Where F a scaling factor, a constant between 0 and 2. r1, r2, r3 are random solutions. The randomly chosen inters are different from the index i. F control the amplification of the differential variation. Crossover operation mainly increases the diversity of the vector solution. The operation is performed using the following equation.

$$u^j = \{v^j if r \leq P_c OR j = \delta x^j if r > P_c AND j \neq \delta \quad (7)$$

Where Pc crossover probability, ∂ randomly selected variable location. R is a random number between o and 1, vj variable of donor vector, xj variable of target vector. The trial vector is created either from the donor vector or from the target vector using the expression. Probability of crossover is generally high which generate more variable from the donor.

The selection operation decides whether the trial vector replaces the target vector or not in the current generation. It decides using the greedy selection mechanism. DE is classified as [9] or [10] depending on the types of mutation and the number of differences involved. Depending on that, 10 variants of DE is proposed in the literature. The basic notation is DE/x/y/z, where x specifies the vector to be mutated (randomly or best), y is the number of difference vectors used, z denotes the crossover scheme (i.e., uniform or exponential).

SIMULATION RESULTS

A. Simulation setup

We implemented all the algorithms (TLBO, ABC, DE) with MATLAB software. Every algorithm was repeated 30 more times with various arbitrary seeds, and the mean fitness of the final solutions was recorded. The number of iterations was set to be 2500 for all the functions for all the algorithms. Specific parameters used for different algorithms are described below. For TLBO: Populace Size was set to be 10. For ABC: swarm size was set to be 20, Limit was set to be 100.For DE: Population size was set to be 100, F=0.5 and Pc=0.9.

Each algorithm was tried with the entirety of the mathematical benchmark function that appeared in Table I. The algorithms are run with various benchmark functions with dimensions 5, 30, and 100. The mean and standard deviations are determined from the results for comparison are shown in the tables from Table- II to IV. Convergence graphs of different benchmark functions with varying algorithms for dimension 100 are compared and shown in Figures 1 to 5.

B. Benchmark functions

For evaluation of the three algorithms, we take into consideration a set of benchmark problems previously used by Karaboga and Akay [13], [14]. The list given here contained a diverse and different set of problems, such as unimodal and multimodal functions. Table-I has the benchmark functions, the range of the search spaces and the global minimum fitness values. The benchmark functions cited in Table I are shown in Table II to IV to compare the performance. The convergence graph for dimension 100 of the benchmark functions is plotted using the iteration number versus the average of best fitness value.

TABLE I: BENCHMARK PROBLEM

Function	Formulae	Range	Minimum Value
Sphere	$f(\vec{x})=\sum_{i=1}^{n} x_i^2$	[-100,100]	$f(\vec{0})=0$
Rosenbrock	$f(\vec{x})=\sum_{i=1}^{n-1}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$	[-30,30]	$f(1)=0$
Griewank	$f(\vec{x})=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[-600,600]	$f(\vec{0})=0$
Rastrigin	$f(\vec{x})=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	[-5.12,5.12]	$f(\vec{0})=0$
Ackley	$f(\vec{x})=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)-\exp\left(\frac{1}{n}\sqrt{\sum_{i=1}^{n}\cos(2\pi x_i}\right)+20+e$	[-32 ,32]	$f(\vec{0})=0$

C. Functions with dimensionality 5

For dimension 5, from Table-II, we saw that both TLBO and DE perform almost the same for Ackley and Griewank functions. The analysis of ABC and DE are similar for Rastrigin Function. DE out the form over TLBO and ABC in Rosenbrock functions is best among other Sphere functions in this dimensionality.

D. Functions with dimensionality 30

Table-III shows the simulation results of the benchmark function with dimension 30. TLBO is best when we consider the unimodal function sphere function followed by DE and then ABC. Similarly, TLBO is better for Ackley function as compared to ABC and DE. Both TLBO and DE were performing best for the Grievance function. ABC outperform in the Rastrigin function and Rosenbrock function among TLBO and DE.

E. Functions with dimensionality 100

The result of dimensionality 100 is shown in Table IV. The table shows that TLBO is the best algorithm for finding the global optima among the other two, i.e., AB and DE, in almost all benchmark functions except the Rosenbrock function. Fig. 1 to 5 plotted with iteration number in X-axis and the logarithm value of the average best fitness values in the Y-axis to verify the speed of convergence.

TLBO is the best algorithm among ABC and DE in terms of speed when considering the sphere function (Fig.-1).ABC and DE are almost the same. For the Ackley function, none of the algorithms performs better, but if we arrange the algorithms in decreasing order, TLBO comes first, then ABC, then DE (Fig.-2). For the Griewank function, TLBO convergence is better than the other two. The convergence performance is almost the same in the case of ABC and DE (Fig. 3). The convergence of TLBO is better followed by ABC and then DE for Rastrigin function (Fig.- 4). For Rosenbrock function ABC convergence quickly than the other two algorithms, as shown in Fig. 5.

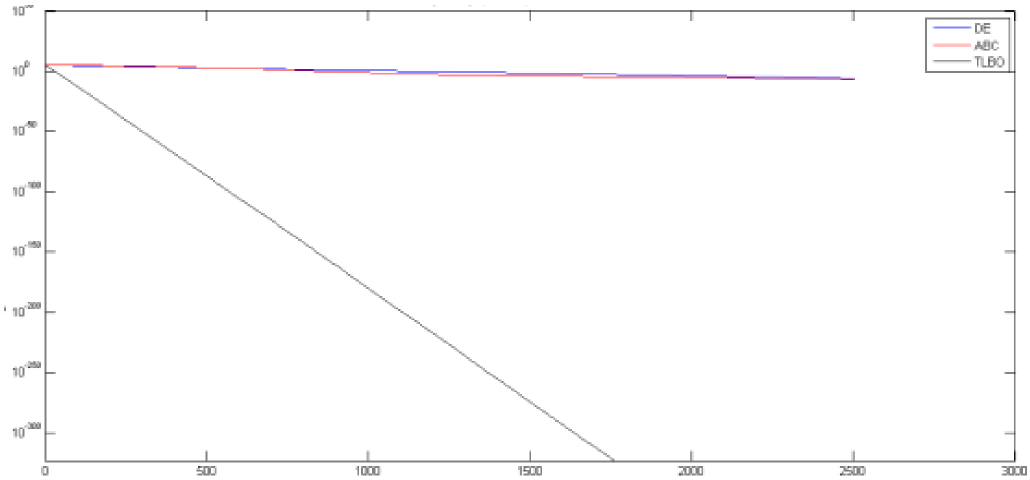

Fig. 1: Converge graph for Sphere Function with D=100

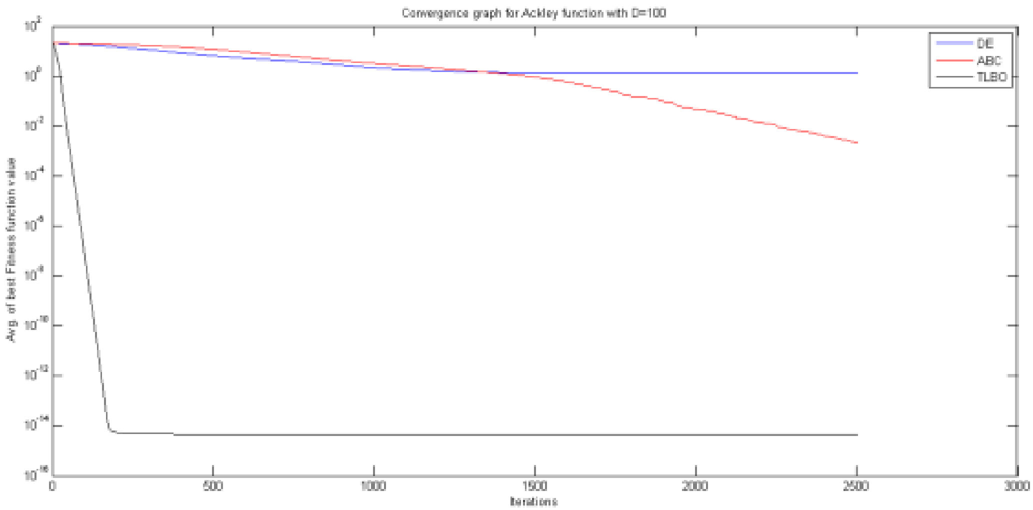

Fig. 2: Converge graph for Ackley Function with D=100

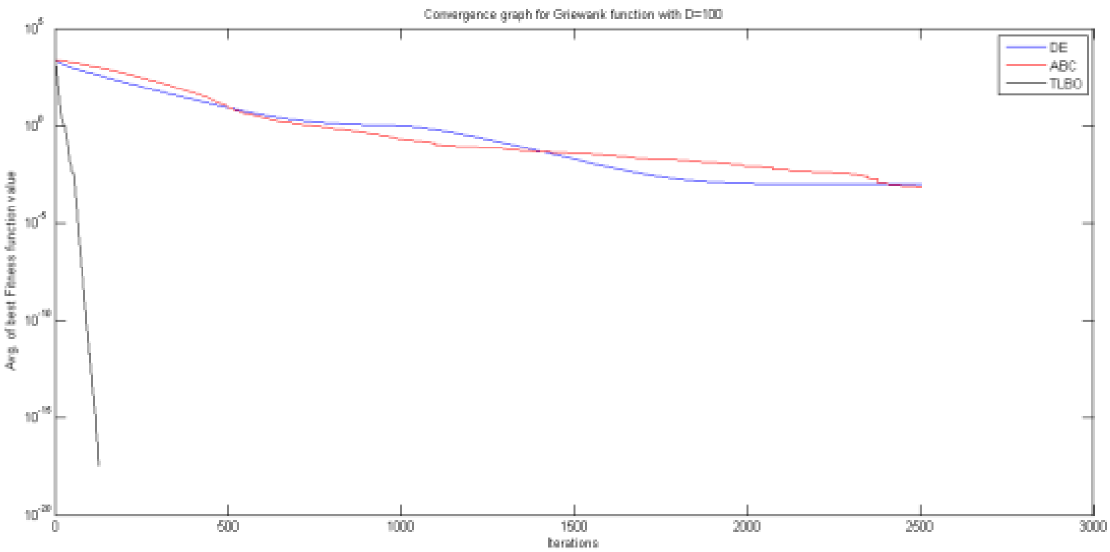

Fig. 3: Converge graph for Griewank Function with D=100

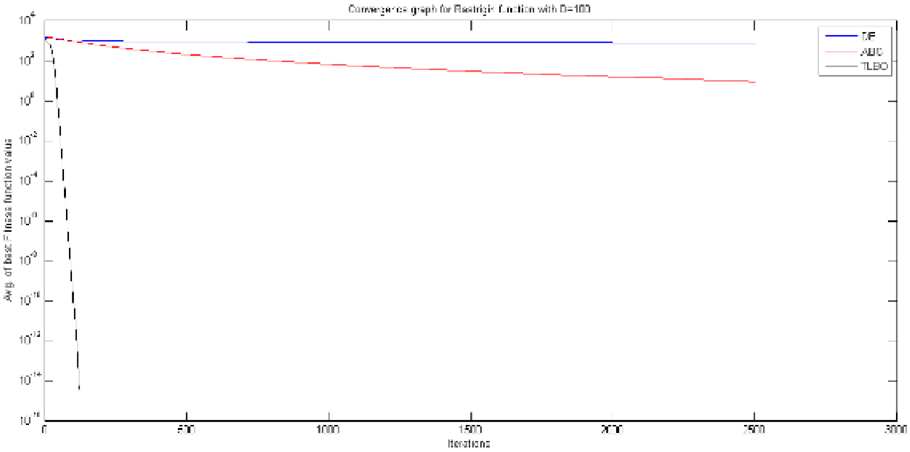

Fig. 4: Converge graph for Rastrigin Function with D=100

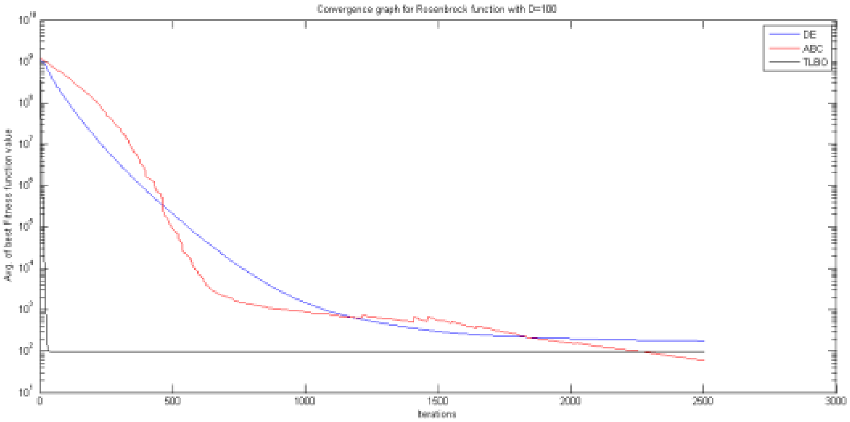

Fig. 5: Converge graph for Rosenbrock Function with D=100

Table II: Result of Benchmark Problems with Dimensionality 5

Function/Algorithm	TLBO		ABC		DE	
	Mean	Sta.Dev.	Mean	Sta.Dev.	Mean	Sta.Dev.
Sphere	**0.00E+00**	0.00E+00	1.61E-17	7.02E-18	2.03E-212	0.00E+00
Ackley	**8.88E-16**	0.00E+00	3.97E-15	1.23E-15	**8.88E-16**	0.00E+00
Griewank	**0.00E+00**	0.00E+00	2.73E-13	1.46E-12	**0.00E+00**	0.00E+00
Rastrigin	1.23E-01	4.78E-01	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
Rosenbrock	2.29E-08	5.01E-08	1.47E-01	1.65E-01	**0.00E+00**	0.00E+00

TABLE III: Result of Benchmark Problems with Dimensionality 30.

Func/Algo	TLBO		ABC		DE	
	Mean	Sta.Dev.	Mean	Sta.Dev.	Mean	Sta.Dev.
Sphere	**0.00E+00**	0.00E+00	8.35E-16	1.61E-16	5.58E-26	4.02E-26
Ackley	**4.44E-15**	0.00E+00	6.07E-14	7.64E-15	9.31E-14	5.27E-14
Griewank	**0.00E+00**	0.00E+00	2.69E-04	1.37E-03	**0.00E+00**	0.00E+00
Rastrigin	1.63E+00	5.41E+00	**1.15E-10**	6.06E-10	1.47E+02	3.00E+01
Rosenbrock	2.51E+01	4.58E-01	**6.39E-01**	1.05E+00	6.28E+00	1.21E+00

Table IV: Result of Benchmark Problems with Dimensionality 100.

Function/Algorithm	TLBO		ABC		DE	
	Mean	Sta.Dev.	Mean	Sta.Dev.	Mean	Sta.Dev.
Sphere	**0.00E+00**	0.00E+00	1.64E-07	1.30E-07	1.37E-06	6.40E-07
Ackley	**4.44E-15**	0.00E+00	2.19E-03	1.41E-03	1.33E+00	5.06E+00
Griewank	**0.00E+00**	0.00E+00	7.28E-04	2.87E-03	9.87E-04	3.08E-03
Rastrigin	**0.00E+00**	0.00E+00	9.32E+00	2.28E+00	7.82E+02	4.66E+01
Rosenbrock	9.69E+01	7.46E-01	**5.78E+01**	3.56E+01	1.78E+02	5.57E+01

CONCLUSION

The comparison measures are the mean solution and standard deviation from various individual runs. The mean solution portrays the normal capacity of the algorithm to locate the global result. The standard deviation portrays the variation in results from the mean solution. In this simulation, the algorithm runs for a predefined number of iterations. The outcomes are gotten for various independent runs, and the mean and standard deviation are determined for the outcomes got in multiple runs. The dimensions of the benchmark function are taken as five, thirty and hundred for all the problem. So, the analysis is performed for small scale to large scale problems. The result shows that TLBO is out of form to find the optimum value with an increase in dimension. Another advantage of TLBO is that it uses a smaller number of control parameters to implement the algorithm. TLBO algorithm delivered sensible outcomes in any event for high dimensions for both uni-modal and multimodal functions.

REFERENCES

1. R. V. Rao, V. J. Savsani, and D. P. Vakharia, “Teaching-Learning- Based Optimization: An optimization method for continuous non-linear large scale problems,” Inf. Sci. (Ny)., vol. 183, no. 1, pp. 1–15, Jan. 2012.
2. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, “A comprehensive survey: Artificial bee colony (ABC) algorithm and applications,” Artif. Intell. Rev., vol. 42, no. 1, pp. 21–57, 2014.
3. J. Vesterstrom and R. Thomsen, “A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems,” Proc. 2004 Congr. Evol. Comput. CEC2004, vol. 2, pp. 1980–1987, 2004.
4. M. Kaufmann et al., “Artificial Intelligence: Genetic Programming,” Wiley, 1990.
5. B. Fogel, “An introduction to simulated evolutionary optimization,” Evol. Comput. Foss. Rec., vol. 5, no. 1, pp. 3–14, 1998.
6. R. Storn and K. Price, “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” J. Glob. Optim., vol. 11, no. 4, pp. 341–359, 1997.
7. Waghmare, “Comments on ‘A note on teaching-learning-based optimization algorithm,’” Inf. Sci. (Ny)., vol. 229, pp. 159–169, Apr. 2013.
8. M. Črepinšek, S. H. Liu, and L. Mernik, “A note on teaching-learning-based optimization algorithm,” Inf. Sci. (Ny)., vol. 212, pp. 79–93, Dec. 2012
9. S. Das and P. N. Suganthan, “Differential evolution: A survey of the state-of-the-art,” IEEE Trans. Evol. Comput. vol. 15, no. 1, pp. 4–31, 2011.
10. A. P. Piotrowski, “Review of Differential Evolution population size,” Swarm Evol. Computer, vol. 32, pp. 1–24, Feb. 2017.
11. S. Singh Jadon, R. Tiwari, H. Sharma, and J. Chand Bansal, “Hybrid Artificial Bee Colony Algorithm with Differential Evolution.” Applied Soft Computing Journal, vol. 58, pp. 11–24, sep 2017.