HBBABC: A Hybrid Optimization Algorithm Combining Biogeography Based Optimization (BBO) and Artificial Bee Colony (ABC) Optimization For Obtaining Global Solution Of Discrete Design Problems

Vimal Savsani

Department of Mechanical engineering, Pandit Deendayal Petroleum University, Gandhinagar 382 007, Gujarat, INDIA

Abstract

Artificial bee colony optimization (ABC) is a fast and robust algorithm for global optimization. It has been widely used in many areas including mechanical engineering. Biogeography-Based Optimization (BBO) is a new biogeography inspired algorithm. It mainly uses the biogeography-based migration operator to share the information among solutions. In this work, a hybrid algorithm with BBO and ABC is proposed, namely HBBABC (Hybrid Biogeography based Artificial Bee Colony Optimization), for the global numerical optimization problem. HBBABC combines the searching behavior of ABC with that of BBO. Both the algorithms have different solution searching tendency like ABC have good exploration searching tendency while BBO have good exploitation searching tendency. To verify the performance of proposed HBBABC, 14 benchmark functions are experimented with discrete design variables. Moreover 5 engineering optimization problems with discrete design variables from literature are also experimented. Experimental results indicate that proposed approach is effective and efficient for the considered benchmark problems and engineering optimization problems. Compared with BBO and ABC separately HBBABC performs better.

1. Introduction

Hybridization of algorithm means to combine the capabilities of different algorithm in a single algorithm. Hybridization is done to overcome the drawback in the existing algorithms and to obtain better solutions. Evolutionary Algorithms (EAs) are very popular for the hybridization due to their different capabilities in handling different types of problems. There is a continuous research to find new optimization techniques which are capable of handling variety of problems with high effectiveness, efficiency and flexibility and thus there are many such optimization algorithms like GA, SA, DE, PSO, ACO, SFLA, ABC, BBO etc. Hybridization is one of the popular methods to increase the effectiveness, efficiency and flexibility of the algorithm to produce better solution and convergence rates and thus saving computational times. Many such hybrid algorithms are available in the literature and continuous efforts are continued to develop new hybrid algorithms. Literature survey of some of the recently developed hybrid algorithm is given in Table 1.ABC is a simple and powerful population-based algorithm for finding the global optimum solutions. ABC divides the population in two main parts viz. employed bees and onlooker bees. Employed bees start the search with specific rules and onlooker bees follow the employed bees corresponding to the fitness of employed bees and it also updated the solution as employed bees. If there is no change in the fitness of employed bees for some number of generations than that bee is converted in scout bee which start for a new search and acts as a employed bees from then. Algorithm continues for predefined number of generations or until the best solution is found. So ABC finds the global solution by exploring the search space with specific rules followed by employed bees, onlooker bees and scout bees. Biogeography-Based Optimization (BBO), proposed by Simon (2008), is a new global optimization algorithm based on the biogeography theory, which is the study of distribution of species. BBO is also population-based optimization method. In the original BBO algorithm, each solution of the population is a vector of integers. BBO updates the solution following immigration and emigration phenomena of the species from one place to the other which is referred as islands by Simon. Simon compared BBO with seven other optimization methods over 14 benchmark functions and a real-world sensor selection problem. The results demonstrated the good performance of BBO. BBO has good exploitation ability as solution is updated by exchanging the existing design variables among the solution.

In order to combine the searching capabilities of ABC and BBO, in this paper, we propose a hybrid ABC with BBO, referred to as HBBABC, for the global numerical optimization problems. In HBBABC, algorithm starts by updating the solutions using immigration and emigration rates. Solution is further modified using the exploration tendency of ABC using employed, onlooker and scout bees.Experiments have been conducted on 14 benchmark functions with discrete design variables Simon (2008) and also on 5 engineering optimization problems.

2. Optimization Algorithms

2.1 Biogeography-based optimization (BBO*)*

BBO (Simon 2006) is a new population-based optimization algorithm inspired by the natural biogeography distribution of different species. In BBO, each individual is considered as a “habitat” with a habitat suitability index (HIS). A good solution is analogous to an island with a high HSI, and a poor solution indicates an island with a low HSI. High HSI solutions tend to share their features with low HSI solutions. Low HSI solutions accept a lot of new features from high HSI solutions. In BBO, each individual has its own immigration rate λ and emigration rate μ. A good solution has higher μ and lower λ and vice versa. The immigration rate and the emigration rate are functions of the number of species in the habitat. They can be calculated as follows

$$\lambda_k = I\left(1-\frac{k}{n}\right) \tag{1}$$

$$\mu_k = E\left(\frac{k}{n}\right) \tag{2}$$

where I is the maximum possible immigration rate; E is the maximum possible emigration rate; k is the number of species of the k-th individual; and n is the maximum number of species. In BBO, there are two main operators, the migration and the mutation.

2.1.1 Migration

Consider a population of candidate which is represented by design variable. Each design variable for particular population member is considered as SIV for that population member. Each population member is considered as individual habitat/Island. The objective function value indicates the HSI for the particular population member. Immigration and emigration rates are decided from the curve given in Figure 1. In Figure 1 the nature of curve is assumed to be same (linear) for immigration and emigration but with opposite slopes. S value represented by the solution depends on its HSI. S_1 and S_2 represent two solutions with different HSI. The emigration and immigration rates of each solution are used to probabilistically share information between habitats. If a given solution is selected to be modified, then its immigration rate λ is used to probabilistically modify each suitability index variable (SIV) in that solution. If a given SIV in a given solution S_i is selected to be modified, then its emigration rates μ of the other solutions is used to probabilistically decide which of the solutions should migrate its randomly selected SIV to solution S_i. The above phenomenon is known as migration in BBO. Because of this migration phenomenon BBO is well suited for the discrete optimization problems as it deals with the interchanging of design variables between the population members.

2.1.2 Mutation

In nature a habitat's HSI can change suddenly due to apparently random events (unusually large flotsam arriving from a neighbouring habitat, disease, natural catastrophes, etc.). This phenomenon is termed as SIV mutation, and probabilities of species count are used to determine mutation rates.

This probability mutates low HSI as well as high HSI solutions. Mutation of high HSI solutions gives them the chance to further improve. Mutation rate is obtained using following equation.

$$m(S) = m_{\max}\left(1-\frac{P_s}{P_{\max}}\right) \tag{3}$$

Where, m_{max} is a user-defined parameter called mutation coefficient.

2.2 Artificial Bee Colony (ABC) technique

Artificial Bee Colony (ABC) Algorithm is an optimization algorithm based on the intelligent foraging behaviour of honey bee swarm. The colony of artificial bees consists of three groups of bees: employed bees, onlookers and scouts (Karaboga 2005, Basturk and Karaboga 2006). An employed bee searches the destination where food is available. They collect the food and returns back to its origin where they perform waggle dance depending on the amount of food available at the destination. The onlooker bee watches the dance and follows employed bee depending on the probability of the available food means more onlooker bee will follow the employed bee associated with the destination having more amount of food. The employed bee whose food source becomes abandoned convert into a scout bee and it searches for the new food source. For solving optimization problems the population is divided into two parts consisting of employed bees and onlooker bees. An employed bee searches the solution in the search space and the value of objective function associated with the solution is the amount of food associated with that solution. Employed bee updates its position using Equation (4) and it updates new position if it is better than the previous position, i.e it follows greedy selection.

$$v_{ij} = x_{ij} + R_{ij}(x_{ij} - x_{kj}) \tag{4}$$

Where v_{ij} is the new position of employes bee, x_{ij} is the current position of employed bee, k is a random number between *(*1, *N(population size)/2)* $\neq i$ and j =1, 2,...,*Number of design variables.* R_{ij} is a random number between (-1, 1).An onlooker bees chooses a food source depending on the probability value associated with that food source, p_i , calculated using Equation (5).

$$p_i = \frac{F_i}{\sum_{n=1}^{N/2} F_n} \tag{5}$$

Where F_i is the fitness value of the solution i and $N/2$ is the number of food sources which is equal to the number of employed bees.Onlooker bees also update its position using Equation (4) and also follow greedy selection. The Employed bee whose position of the food source cannot be improved for some predetermined number of cycles than that food source is called abandoned food source. That employed bee becomes scout and searches for the new solution randomly using Equation (6).

$$x_i^j = x_{\min}^j + rand(0,1)(x_{\max}^j - x_{\min}^j) \tag{6}$$

The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called "*limit*" for abandonment. The value of limit is generally taken as *Number of employed bees*Number of design variables* (Karaboga and Basturk 2007, Karaboga and Basturk 2008).

2.3 Hbbabc: Hybrid Biogeography Based Artificial Bee Colony Optimization

As mentioned above, ABC is good at exploring the search space and locating the region of global minimum. On the other hand, BBO has a good exploitation searching tendency for global optimization. Based on these considerations, in order to maximize the exploration and the exploitation a HBBABC approach is proposed. Step by step procedure for the implementation of HBBABC is given as follows.

Step 1: Initialize BBO and ABC parameters which are necessary for the algorithm to proceed. These parameters includes population size, number of generations necessary for the termination criteria, Maximum immigration and emigration rates, number of design variables and respective range for the design variables.

Step 2: Generate random population equal to the population size specified. Each population member contains the value of all the design variables. This value of design variable is randomly generated in between the design variable range specified. Every design variable in the population indicates SIVs for that respective population member (Habitat)

Step 3: Obtain the value of objective function for all population members. The value of objective function so obtained indicates the HSI for that Habitat (population member). If problem is constrained optimization problem than some penalty approach is used to convert constrained optimization problem to unconstrained optimization problem.

Step 4: Map the value of HSI to obtain the species count. High species count is allotted to the population member having high HSI for maximization optimization problem. If the optimization problem is of minimization type than low HSI member is given high species count.

Step 5: Modify population using the migration operator considering its immigration and emigration rates. If a given solution is selected to be modified, then its immigration rate λ is used to probabilistically modify each suitability index variable (SIV) in that solution. If a given SIV in a given solution Si is selected to be modified, then its emigration rates μ of the other solutions is used to probabilistically decide which of the solutions should migrate its randomly selected SIV to solution S_i. Pseudo code for migration is given as follows.

```
For i = 1 to NP
    Select X_i with probability proportional to λ_i
        if rand(0, 1) < λ_i
        For j = 1 to NP
            Select X_j with probability proportional to μ_j
            if rand(0, 1) < μ_j
            Randomly select a variable σ from X_j
            Replace the corresponding variable in X_i with σ
            Endif
        Endif
        End
End
```

Step 6: Divide the population into two equal parts to act as employed bees and onlooker bees. Obtain the value of objective function for employed bees. The value of objective function so obtained indicates the amount of nectar (food) associated with that destination (food source).

Step 7: Update the position of employed bees using Equation (4). If the value of objective function of the new solution is better than the existing solution , replace the existing solution with the new one.

Step 8: Calculate probability associated with the different solutions using Equation (5). Onlooker bee follows a solution depending on the probability of that solution. So more the probability of the solution more will be the onlooker bee following that solution.

Step 9: Update the position of onlooker bees using Equation (4). If the value of objective function of the new solution is better than the existing solution, replace the existing solution with the new one

Step 10: Identify abandon solution and replace it with the newly generated solution using Equation (6)

Step 11: Continue all the steps from step 3 until the specified number of generations are reached.
Detailed pseudo code is given below:

```
START
        Initialize required parameters necessary for the algorithm (as mentioned above)
        Generate the initial population N, Evaluate the fitness for each individual in N
For i=1 to number of generations
        ____________________________________________________________________
        BBO loop
        ____________________________________________________________________
        For each individual, map the fitness to the number of species
        Calculate the immigration rate λi and the emigration rate μi for each individual Xi
        For i = 1 to N
                Select Xi with probability proportional to λi
                if rand(0, 1) < λi
                For j = 1 to N
                        Select Xj with probability proportional to μj
                        if rand(0, 1) < μj
                        Randomly select a variable σ from Xj
                        Replace the corresponding variable in Xi with σ
                        Endif
                Endif
                End
        End
        ____________________________________________________________________
        ABC loop
        ____________________________________________________________________
        For i = 1 to N/2
                Produce new solutions υij for the employed bees and evaluate them
                Replace new solution if it is better than the previous one
        End
                Calculate the probability values pij for the solutions
                Identify onlooker bees depending on the probability pij
        For i = 1 to N/2
                Produce the new solutions υij for the onlookers
                Replace new solution if it is better than the previous one
        End
                Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced
        solution xij
        End
End
STOP
```

It can be seen from the pseudo code that HBBABC requires small extra calculation effort than BBO but it combines the strength of both the algorithms in searching the optimum solutions. To demonstrate the effectiveness of the proposed algorithm many experiments were conducted on benchmark problems which is discussed in the next section.

3. Application On Benchmark Problems

In the field of optimization it is a common practice to compare different algorithms using different benchmark problems. In this work 14 different benchmark problems are considered having different characteristics like separability, multimodality and regularity (Simon 2008). A function is multimodal if it has two or more local optima. A function is separable if it can be written as a sum of functions of variable separately. Function is regular if it is differentiable at each point of their domain. Non seperable functions are more difficult to optimize and difficulty increases if the function is multimodel. Complexity increases when the local optima are randomly distributed. Moreover complexity increases with the increase in dimensionality. Description of all the benchmark problems with respect to Multimodality, Separability and regularity are given in Table 2. All the benchmark problems are explained as follows with their optimum value and optimum design parameters.Sphere Model $f(x) = \sum_{i=1}^{30} x_i^2$

$$-100 \le x_i \le 100, \quad \min(f) = f(0,0...0) = 0 \tag{7}$$

Schwefel's Problem 2.22

$$f(x)=\sum_{i=1}^{30}\left|x_i\right|+\prod_{i=1}^{30}\left|x_i\right|$$

$$-100\le x_i\le 100,\quad \min(f)=f(0,0...0)=0 \tag{8}$$

Schwefel's Problem 1.2

$$f(x)=\sum_{i=1}^{30}\left(\sum_{j=1}^{i}x_j\right)^2$$

$$-100\le x_i\le 100,\quad \min(f)=f(0,0...0)=0 \tag{9}$$

Schwefel's Problem 2.21

$$f(x)=\max_i\left\{\left|x_i\right|,1\le i\le 30\right\}$$

$$-100\le x_i\le 100,\quad \min(f)=f(0,0...0)=0 \tag{10}$$

Generalized Rosenbrock's Function

$$f(x)=\sum_{i=1}^{29}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$$

$$-30\le x_i\le 30,\quad \min(f)=f(1,1...1)=0 \tag{11}$$

Step Function

$$f(x)=\sum_{i=1}^{30}\left[x_i+0.5\right]^2$$

$$-100\le x_i\le 100,\quad \min(f)=f(0,0...0)=0 \tag{12}$$

Quartic Function

$$f(x)=\sum_{i=1}^{30}\left[ix_i^4\right]$$

$$-1.28\le x_i\le 1.28,\quad \min(f)=f(0,0...0)=0 \tag{13}$$

Generalized Schwefel's Problem 2.26

$$f(x)=-\sum_{i=1}^{30}\left(x_i\sin\left(\sqrt{\left|x_i\right|}\right)\right)$$

$$-500\le x_i\le 500,\quad \min(f)=f(420.9687,420.9687...420.9687)=-12569.5 \tag{14}$$

Generalized Rastrigin Function

$$f(x)=\sum_{i=1}^{30}\left[x_i^2-10\cos(2\pi x_i)+10\right]$$

$$-5.12\le x_i\le 5.12,\quad \min(f)=f(0,0...0)=0 \tag{15}$$

Ackley's function

$$f(x)=\sum_{i=1}^{30}-20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30}x_i^2}\right)-\exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos 2\pi x_i\right)$$

$$-32\le x_i\le 32,\quad \min(f)=f(0,0...0)=0 \tag{16}$$

Generalized Griewank function

$$f(x)=\frac{1}{4000}\sum_{i=1}^{30}x_i^2-\prod_{i=1}^{30}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$$

$$-600\le x_i\le 600,\quad \min(f)=f(0,0...0)=0 \tag{17}$$

Generalized penalized function 1

$$f(x) = \frac{\pi}{30}\left[10\sin^2(\pi y_1) + \sum_{i=1}^{29}(y_i - 1)^2\{1 + 10\sin^2(\pi y_{i+1})\} + (y_n - 1)^2\right] + \sum_{i=1}^{30} u(x_i, 10, 100, 4)$$

$$-50 \le x_i \le 50, \quad \min(f) = f(1,1 \ldots 1) = 0 \tag{18}$$

Generalized penalized function 2

$$f(x) = 0.1\left[\sin^2(\pi 3x_1) + \sum_{i=1}^{29}(x_i - 1)^2\{1 + \sin^2(3\pi x_{i+1})\} + (x_n - 1)^2(1 + \sin^2(2\pi x_{30})\right] + \sum_{i=1}^{30} u(x_i, 5, 100, 4)$$

$$-50 \le x_i \le 50, \quad \min(f) = f(1,1 \ldots 1) = 0$$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a, \\ 0, & -a \le x_i \le a, \\ k(-x_i - a)^m, x_i < -a & \end{cases}$$

$$y_i = 1 + 1/4(x_i + 1) \tag{19}$$

Fletcher-Powell

$$f_{14}(x) = \sum_{i=1}^{30}(A_i - B_i)^2$$

$$A_i = \sum_{j=1}^{30}(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j), \; B_i = \sum_{j=1}^{30}(a_{ij}\sin x_j + b_{ij}\cos x_j)$$

$$x_i, \alpha_i \in [-\pi, \pi]; a_{ij}, b_{ij} \in [-100, 100], \quad \min(f_{14}) = f_{14}(\alpha, \ldots \alpha) = 0 \tag{20}$$

3.1 Discrete optimization of benchmark problems

14 Benchmark problems were compared by implementing integer versions of all the optimization algorithms in Matlab. The granularity or precision of each benchmark function was 0.1, except for the Quartic function (Simon 2008). Since the domain of each dimension of the quartic function was only ±1.28, it was implemented with a granularity of 0.01. ABC was specially tuned to take integer value in each iteration modifying Equation (6) and (8) as.

$$v_{ij} = round(x_{ij} + R_{ij}(x_{ij} - x_{kj})) \tag{21}$$

And

$$x_i^j = round(x_{\min}^j + rand(0,1)(x_{\max}^j - x_{\min}^j)) \tag{22}$$

All the algorithms were run for 100 times considering different population size and number of generations depending on the complexity of problem. For BBO and HBBABC Habitat modification probability was taken as 1 and for ABC limit is set as Population size/2 * Number of design variables. Results for the discrete optimization of benchmark problems are given in Table 3 which shows the Best Value, Mean Value and T-Test Value for all the benchmark problems. Results are also shown considering different number of design variables viz. 30, 50, 100 and 200 (Dimensions). Population size is taken as 50 for all the problems and number of generation taken is equal to 50 for 30 and 50 dimensions, 100 for 100 dimensions and 200 for 200 dimensions.It is seen from the Table that out of 14 benchmark problems considering discrete design variables hybrid HBBABC has shown better result for 12 bench mark problems for the best and mean values for 100 runs. Only for Schwefel 1.2 and Schwefel 2.21 BBO has outperformed HBBABC though the best result obtained considering 200 design variables for HBBABC is better than BBO. Further statistical analysis is done to analyze the differences between different algorithms using T-test method (Hill 1970). T-test method is used to check whether the differences between the groups of data are statistically significant or not. T-test value also suggest that HBBABC is statistically significant than BBO and ABC. ABC has shown better performance than BBO for only Griewank function, but for Sphere, Schwefel 2.22, Rosenbrock, Step, and Quarticperformance of ABC has increased with the number of design variables. Moreover discrete version of ABC is not so effective in comparison with HBBABC expect for sone results of Scwefel 1.2 and Schwefel 2.21.urther comparison of the algorithms is done considering two more criterions like success rate and maximum number of generation required to reach the best solution. Results are shown in Table 4. Comparison is done considering the population size of 100 and number of generations as 500. If the optimum result is Not Reached than it is marked as NR. Algorithm is considered successful if the solution value is less than 1.0E-15 and also maximum number of generation required is counted for the same value. Out of 14 benchmark problems HBBABC was able to find optimum solution for 10 benchmark problems which is quite better than BBO

(4 benchmark problems) and ABC (3 benchmark problems) also HBBABC has shown 100% success for 9 benchmark problems and about 62% for Rastrigin function which is also better than BBO and ABC. NR* indicates that though optimum solution has not reached but still the best solution achieved is better than other algorithms. It is also seen that with increase in the dimension of the problem still HBBABC is better than BBO and ABC on 12 benchmark problems. Moreover average number of generations required to reach the optimum solution (less than 1.0E-15) for 100 different runs was calculated and was rounded off in multiples of 10 (e.g. if average number of generation equals to 147 than it is rounded off to 150). These rounded values of average number of generation are shown in the Table 4. It is seen from the Table 4 that HBBABC requires less number of generations to reach the optimum solution than BBO and ABC, which indirectly indicates that HBBABC requires less computational effort than BBO and ABC.Simon (2008), founder of BBO, has compared the results of BBO with different evolutionary algorithms such as ACO, DE, ES, GA, PBIL, PSO and SGA for 14 different benchmark problems considered in this paper. The results were presented for best as well as the mean solutions. Results show that for mean solution SGA is better than BBO for Griewank, Penalty 1, Penalty 2, Quartic, Rosenbrock, Schwefel 2.26 and Step functions. For Best solution SGA is better for Fletcher, Griewank, Quartic, Schwefel 1.2, Schwefel 2.21, Sphere and Step functions while ACO is better for Penalty 1, Penalty 2 and Schwefel 2.26. To compare HBBABC with different Evolutionary algorithms results were compared with the best performing optimization algorithm in comparison with BBO considering same parameters used by Simon (2008) (i.e population size of 50, number of generations of 50, 100 number of runs, ACO parameters and SGA parameters) . Results are present in Table 5 for the comparison of HBBABC with SGA and ACO. It is seen form the results that for the mean solution HBBABC has outperformed SGA for all the considered benchmark problems and for the Best solution HBBABC has also outperformed SGA and ACO except for Penalty 1 , Penalty 2 and Schwefel 1.2. Here it is interesting to note that though for Penalty 1 and Penalty 2 ACO has produced better results than HBBABC but mean results for ACO is poor than HBBABC which indicates that average performance of HBBABC is better than ACO. Further investigation is done for the convergence rate which gives the value of objective function with respect to number of generations. Figure 2 shows the convergence graph for some of the benchmark problems for 100 generations. It is seen from the Figure 2 that convergence rate for HBBABC is better than BBO and ABC.*3.2 Discrete optimization of engineering problems*To check the validity of the proposed hybrid algorithm 5 different real-life engineering design problems with discrete variables is considered from the literature. All the problems considered are constrained optimization problems and so it is required to convert constrained optimization problem into unconstrained optimization problem. Penalty approach is used to change the constrained optimization problem into unconstrained optimization problem. Consider an optimization problem as

$$Minimize\ f(X), Subjected\ to\ g_i(X) \geq 0$$

This problem is converted in unconstrained form as

$$Minimize\ f(X)\ if\ g_i(X) \geq 0, else$$

$$Minimize\ f(X) + R\sum_{i=1}^{n} g_i(X)$$

Where *R* is very large number and *n* is number of constraints.
In this work one more term is added in the above equation to make sure that all the constraints are satisfied.

$$Minimize\ f(X) + R\sum_{i=1}^{n} g_i(X)\ + R_1(G)$$

Where R_1 is also a very large number and *G* is the number of constraints which are not satisfied.

3.2.1 Example 1: Gear Train Design

This problem was introduced by Pomrehn and Papalambros (1995) to minimize the total weight of the gear train. There are 22 discrete design variables with three different types of discreteness. The number of teeth should be integer, gear/pinion shafts must be located in discrete locations, and gears are to be manufactured from four available gear blanks. There are 86 inequality constraints considered for gear-tooth bending fatigue strength, gear-tooth contact strength, gear-tooth contact ratio, minimum pinion size, minimum gear size, gear housing constraints, gear pitch and kinematic constraints. The above problem was attempted by Khorshid and Seireg (1999) and Dolen et al. (2005). The best solution found so far is 38.13 cm^3 by Khorshid and Seireg (1999).

3.2.2 Example 2: Welded Stiffened Cylindrical Shell

This problem was introduced by Jarmai et. al. (2006) to minimize the cost of welded orthogonally stiffened cylindrical shell. The problem have 5 discrete design variables with 5 inequality constraints for shell buckling, panel stiffener buckling, panel ring buckling and manufacturing limitations. The best solution given by Jarmai et. al. (2006) is f*(x)=55326.3 which is the global solution for the considered problem.

3.2.3 Example 3: 10- Bar Truss Structure

This problem is taken from Rajeev and Krishnamoorthy (1992). In these problems the objective function is to minimize the weight of the structure. Constraints are based on allowable stresses and deflections with 10 discrete design variables for each bar in the structure. The above problem was also solved by many methods such as Improved Penalty

Function Method (Cai and Thiereu, 1993), Genetic Algorithms (Rajeev and Krishnamoorthy, 1992), Difference Quotient Method (Thong and Liu, 2001), Genetic Algorithms (Coello, 1994) and Simulated Annealing (Kripa 2004). The best result shown was f*(x)=5490.74 which is the global optimum solution for the problem.

3.2.4 Example 4: 25- Bar Truss Structure

This problem is also taken from Rajeev and Krishnamoorthy (1992). In these problems the objective function is to minimize the weight of the structure. Constraints are based on allowable stresses and deflections with 8 discrete design variables for different sets of bar in the structure. The above problem is also solve by many methods such as Improved Penalty Function Method (Cai and Thiereu, 1993) , Genetic Algorithms (Rajeev and Krishnamoorthy, 1992), Brach and Bound (Zhu, 1986), Difference Quotient Method (Thong and Liu, 2001), Genetic Algorithms (Coello, 1994), Simulated Annealing (Kripa 2004). The best result is equal to f*(x)=484.85

3.2.5 Example 5: Multispeed Planetary Transmission

This problem is taken from Simionescu et. al. (2006) for the teeth number synthesis of Multispeed Planetary Transmission. The objective function is to minimize the error between imposed and actual transmission ratios. There are 12 kinematic constraints with 10 discrete design variables. The best global solution for the problem is f*(x)=0.526. The result for the above considered problems are shown in Table 6. It can be seen that for all the considered problems HBBABC has given global solutions. The results given by HBBABC are better than BBO and ABC in terms of mean solutions and success rate. Algorithm is considered to be successful if the solution has reached 99% of the global solution. Only for Example 1 algorithm is considered successful if it gives feasible solution. For Example 1 HBBABC has given better results than reported in the literature.

4. Conclusions

A hybrid HBBABC algorithm using two well known algorithms viz. Biogeography Based Optimization and Artificial Bee colony Optimization is proposed in this work. It combined the exploitation capacity of BBO and exploration capacity of ABC. To verify the performance of the proposed method it was experimented on 14 benchmark problems considering discrete design variables and 5 engineering design optimization problems. Comparison of different algorithm is done considering different criteria such as Best solution, Mean Solution, T-test, Success rate, Average number of generations required to reach the optimum solution and Convergence rate. Experimental results show that the overall performance of HBBABC is better than BBO and ABC considering above criteria.

References

[1]. Ali HK, Huseyin C, Tamer A, Gurhan G (2010) PSOLVER: A new hybrid particle swarm optimization algorithm for solving continuous optimization problems. Expert Systems with Applications 37: 6798–6808

[2]. Ali RY (2009) A novel hybrid immune algorithm for global optimization in design and manufacturing. Robotics and Computer-Integrated Manufacturing 25: 261-270

[3]. Basturk B, Karaboga D (2006) An Artificial Bee Colony (ABC) Algorithm for Numeric function Optimization. *IEEE Swarm Intelligence Symposium*, Indianapolis, Indiana, USA

[4]. Behnamian J, Zandieh M, Fatemi Ghomi SMT (2009) Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm. Expert Systems with Applications 36: 9637-9644

[5]. Behnamian J, Fatemi Ghomi SMT (2010) Development of a PSO–SA hybrid metaheuristic for a new comprehensive regression model to time-series forecasting. Expert Systems with applications 37: 974–984

[6]. Berna D, Fulya A, Onder B (2010) Design of reliable communication networks: A hybrid ant colony optimization algorithm. IIE Transactions. 42: 273 - 287

[7]. Cai JB, Thiereut G (1993) Discrete optimization of structures using an improved penalty function method. *Engineering Optimization* 21: 293-306

[8]. Changsheng Z, Jiaxu N, Shuai L, Dantong O, Tienan D (2009) A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. Operations Research Letters 37: 117-122

[9]. Chin KH, Hong TE (2009) GACO - a hybrid ant colony optimization metaheuristic for the dynamic load-balanced clustering problem in ad hoc networks. Applied Artificial Intelligence 23: 570 - 598

[10]. Coello CA (1994) Discrete Optimization of Trusses using Genetic Algorithms. Expert Systems Applications and Artificial Intelligence, *J.G. Cheng, F.G. Attia and D.L. Crabtree (Editors), (I.I.T.T. International, Technology Transfer Series)*: 331-336

[11]. Cuneyt O, Zafer B (2009) Application of heuristic and hybrid-GASA algorithms to tool-path optimization problem for minimizing airtime during machining. Engineering Applications of Artificial Intelligence. 22: 389-396

[12]. Dolen M, Kaplan H, Seireg A (2005) Discrete parameter-nonlinear constrained optimisation of a gear train using genetic algorithms. *International Journal of Computer Applications in Technology*. 24(2): 110 - 121

[13]. Dong HK, Ajith A, Jae HC (2007) A hybrid genetic algorithm and bacterial foraging approach for global optimization. Information Sciences 177: 3918-3937

[14]. Fan SKS, Zahara E (2007) A hybrid simplex search and particle swarm optimization for unconstrained optimization. European Journal of Operational Research 181: 527–548

[15]. Guohui Z, Xinyu S, Peigen L, Liang G (2009) An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. Computers & Industrial Engineering 56: 1309-1318

[16]. Hill G (1970) Algorithm 395: Students t-distribution'. Communication of ACM 13: 617-619

[17]. Hui L, Zixing C, Yong W (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Applied Soft Computing 10: 629–640

[18]. Jarmai K, Snyman JA, Farkas J (2006) Minimum cost design of a welded orthogonally stiffened cylindrical shell. *Computers and Structures* 84: 787–797

[19]. Jerome HK, Darren R (2009) A hybrid CMA-ES and HDE optimisation algorithm with application to solar energy potential. Applied Soft Computing 9: 738-745

[20]. Jing-Ru Z, Jun Z, Tat-Ming L, Michael RL (2007) A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. Applied Mathematics and Computation 185: 1026-1037

[21]. Karaboga D (2005) An Idea Based On Honey Bee Swarm For Numerical Optimization. *Technical Report-TR06, Erciyes University, Engineering Faculty*, Computer Engineering Department

[22]. Karaboga D, Basturk B (2007) Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. *P. Melin et al. (Eds.): IFSA 2007, LNAI 4529*, *Springer-Verlag Berlin Heidelberg* , pp. 789–798.

[23]. Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing* 8: 687–697

[24]. Karen I, Yildiz AR, Kaya N, Ozturk N, Ozturk F (2006) Hybrid approach for genetic algorithm and Taguchi's method based design optimization in the automotive industry. International Journal of Production Research 44: 4897 - 4914

[25]. Khorshid E, Seireg A (1999) Discrete nonlinear optimisation by constraint decomposition and designer interaction. *International Journal of Computer Applications in Technology* 12: 76-82

[26]. Kripa M (2004) Discrete Optimization of Trusses by Simulated Annealing. *Journal of the Brazilian Society of Mechanical Science & Engineering* 26: 170-173

[27]. Liao TW (2010) Two hybrid differential evolution algorithms for engineering design optimization. Applied Soft Computing : doi:10.1016/j.asoc.2010.05.007

[28]. Ling W (2005) A hybrid genetic algorithm–neural network strategy for simulation optimization. Applied Mathematics and Computation 170: 1329-1343

[29]. Nenzi W, Yau-Zen C (2002) A Hybrid Search Algorithm for Porous Air Bearings Optimization. Tribology Transactions 45: 471 – 477

[30]. Nourelfath M, Nahas N, Montreuil B (2007) Coupling ant colony optimization and the extended great deluge algorithm for the discrete facility layout problem. Engineering Optimization 39: 953 - 968

[31]. Pomrehn LP, Papalambros PY (1995) Discrete optimal design formulations with application to gear train design. *ASME Journal of Mechanical Design* 117: 419-424

[32]. Pradeep KG, Ranjan G (2005) An automated hybrid genetic-conjugate gradient algorithm for multimodal optimization problems. Applied Mathematics and Computation 167: 1457-1474

[33]. Qian-jin G, Hai-bin Y, Ai-dong X (2006) A hybrid PSO-GD based intelligent method for machine diagnosis. Digital Signal Processing 16: 402-418

[34]. Rajeev S, Krishnamoorthy CS (1992) Discrete Optimization of Structures Using Genetic Algorithms. *Journal of Structural Engineering* 118: 1233-1250

[35]. Shahla N, Mohammad EB, Nasser G, Mehdi HA (2009) A novel ACO–GA hybrid algorithm for feature selection in protein function prediction. Expert Systems with Applications 36: 12086-12094

[36]. Shu-kai SF, Yun-chia L, Erwie Z (2004) Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions. Engineering Optimization 36: 401 - 418

[37]. Shun-Fa H, Rong-Song H (2006) A hybrid real-parameter genetic algorithm for function optimization. Advanced Engineering Informatics 20: 7-21

[38]. Simionescu PA, Beale D, Dozier GV (2006) Teeth Number Synthesis of Multispeed Planetary Transmission Using An Estimation of Distribution Algorithm. Journal of mechanical design 128: 108-115

[39]. Simon D (2008) Biogeography based optimization. *IEEE transactions on Evolutionary Computations* 99(1) : 1-13

[40]. Taher N (2010) A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem. Applied Energy 87: 327-339

[41]. Taher N, Babak A (2010) An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. Applied Soft Computing 10: 183-197

[42]. Tamer A, Ali HK, Huseyin C, Gurhan G (2009) Hybridizing the harmony search algorithm with a spreadsheet 'Solver' for solving continuous engineering optimization problems. Engineering Optimization 41(12): 1119 - 1144

[43]. Thong WHT, Liu GR (2001) An optimization procedure for truss structures with discrete design variables and dynamic constraints. Computers and Structures 79: 155-162

[44]. Victoire TAA, Jeyakumar AE (2004) Hybrid PSO–SQP for economic dispatch with valve-point effect Electric Power Systems Research 71: 51–59

[45]. Vincent K, Florin C, Olivier L, Louis W (2008) A hybrid optimization technique coupling an evolutionary and a local search algorithm. Journal of Computational and Applied Mathematics 215: 448-456

[46]. Wen-Yi L (2010) A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage . Mechanism and Machine Theory 45: 1096–1107

[47]. Xiaoxia Z, Lixin T (2009) A new hybrid ant colony optimization algorithm for the vehicle routing problem. Pattern Recognition Letters 30: 848-855

[48]. Yannis M, Magdalene M (2010) A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem. Expert Systems with Applications 37: 1446-1455

[49]. Ying-Pin C (2010) An ant direction hybrid differential evolution algorithm in determining the tilt angle for photovoltaic modules. Expert Systems with Applications 37: 5415–5422

[50]. Yi-Tung K, Erwie Z (2008) A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Applied Soft Computing 8: 849-857

Table 1 Details of different hybrid algorithms

Hui et. al. (2010)	PSO + DE
Behnamian and Ghomi (2010)	PSO + SA
Ali et. al. (2010)	PSO + Spread Sheet 'Solver'
Wen (2010)	GA + DE
Ying (2010)	ACO + DE
Liao (2010)	DE + Random walk DE + Harmony Search
Berna et. al. (2010)	ACO +SA
Taher and Babak (2010)	ACO + Fuzzy adaptive PSO + k-means Algorithm
Taher (2010)	Fuzzy Adaptive PSO + Nelder–Mead simplex Search
Yannis and Magdalene (2010)	GA + PSO
Chin and Hong (2009)	GA + ACO
Tamer et. al. (2009)	Harmony Search + Spread Sheet 'Solver'
Changsheng et. al. (2009)	DE + PSO
Guohui etl al. (2009)	PSO + Tabu Search Algorithm
Shahla et. al. (2009)	GA + ACO
Ali (2009)	Immune + Hill climbing algorithm
Cuneyt and Zafer (2009)	GA + SA
Jerome and Darren (2009)	Covariance Matrix Adaptation Evolution Strategy + DE + Backwards Ray Tracing Technique
Xiaoxia and Lixin (2009)	ACO + Scatter Search
Behnamian et. al. (2009)	ACO + SA + Vaiable Neighborhood Search (VNS)
Vincent et. al. (2008)	GA + Local Search Interior Point Method
Yi and Erwir (2008)	GA + PSO
Fan and Zahara (2007)	PSO + Simplex search
Nourelfath et. al. (2007)	ACO + Extended Great Deluge (EGD) Local Search Technique
Jing et. al. (2007)	PSO + Back Propogation Algorithm
Dong et. al. (2007)	Genetic Algorithm (GA) + Bacterial Foraging (BF)
Karen et. al. (2006)	Taguchi's method + GA
Qian et. al. (2006)	PSO + Gradient Descent (GD) Methods
Shun and Rong (2006)	GA + SA
Pradeep and Ranjan (2005)	GA + Local Optimizing Gradient Based Algorithm
Ling (2005) [34]	GA + Neural Network Strategy
Shu et. al. (2004)	Nelder-Mead (NM) + PSO
Victoire and Jeyakumar (2004)	PSO + SQP
Nenzi and Yan (2002)	GA + Simplex Method (SM)

Table 2

Characteristics of benchmark problems

Name	Multimodal?	Separable?	Regular?
Sphere	no	yes	yes
Schwefel 2.22	no	no	no
Schwefel 1.2	no	no	yes
Schwefel 2.21	no	no	no
Rosenbrock	no	no	yes
Step	no	yes	no
Quartic	no	yes	yes
Schwefel 2.26	yes	yes	no
Rastrigin	yes	yes	yes
Ackley	yes	no	yes
Griewank	yes	no	yes
Penalty #1	yes	no	yes
Penalty #2	yes	no	yes

Fletcher-Powell	yes	no	no

Table 3
Best results and Mean results for the benchmark problems considering discrete design variables for different dimensions. **'Bold value'** indicates better solution found. T-Test value is significant with 49 degree of freedom at α=0.05 by two tailed test

		BBO		ABC		HBBABC		TTEST	
		Best	Mean	Best	Mean	Best	Mean	BBO-HBBOABC	ABC-HBBABC
Sphere	30	1.078	2.504	3.215	7.815	**0.050**	**0.143**	23.430	29.171
	50	15.692	20.509	18.615	22.885	**3.296**	**4.460**	13.085	16.615
	100	45.958	61.804	27.373	40.423	**7.781**	**9.968**	17.866	13.356
	200	117.133	149.820	59.827	80.398	**19.351**	**22.552**	16.962	15.481
Schwefel 2.22	30	3.600	7.128	4.000	6.720	**0.000**	**0.066**	38.917	40.539
	50	32.300	41.890	30.800	37.780	**8.100**	**11.160**	17.982	15.549
	100	82.000	97.790	58.600	72.260	**19.500**	**25.220**	21.983	18.078
	200	194.700	214.280	123.800	138.030	**49.300**	**60.100**	33.147	18.453
Schwefel 1.2	30	**1687**	**4971**	5926	13556	7369	12276	-23.507	2.671
	50	**21581**	**30233**	38196	55858	51359	75257	-7.424	-2.901
	100	**92822**	**113219**	156135	226183	178278	228703	-7.208	-0.113
	200	303333	**355509**	571897	728556	**64230**	440653	-0.991	2.924
Schwefel 2.21	30	**25.500**	**47.082**	37.900	60.934	36.900	58.062	-8.525	2.559
	50	**63.900**	**66.868**	78.800	79.914	81.500	79.874	-1.306	0.004
	100	**69.400**	**82.640**	86.500	92.610	85.800	91.070	-3.862	1.053
	200	**88.300**	**92.390**	94.500	96.150	94.300	96.380	-4.888	-0.423
Rosenbrock	30	48.997	117.287	107.642	205.920	**19.963**	**28.034**	25.096	28.686
	50	517.953	698.683	406.147	582.530	**94.113**	**148.841**	13.327	9.478
	100	1572.754	1915.736	796.838	1001.431	**237.627**	**311.123**	16.249	11.844
	200	3617.898	4706.759	1534.069	1857.636	**559.278**	**668.473**	17.562	20.332
Step	30	302	912	742	2828	**16.000**	**75.080**	21.255	27.929
	50	7304	11185	5636	7105	**1460.000**	**2083.600**	12.164	15.180
	100	15464	22224	10566	14859	**3826.000**	**5361.400**	11.514	7.596
	200	38448	52700	24749	29931	**11132.000**	**13777.100**	17.361	11.279
Quartic	30	0.019	0.094	0.037	0.195	**0.000**	**0.000**	15.777	18.343
	50	3.695	9.161	0.850	1.351	**0.062**	**0.140**	6.199	9.436
	100	19.336	40.968	2.877	4.984	**0.331**	**0.870**	8.778	8.410
	200	169.447	239.505	18.221	24.612	**2.674**	**5.074**	17.581	12.171
Schwefel 2.26	30	-11138	-10749	-8907	-8258	**-11963.61**	**-11410.21**	17.710	65.308
	50	-16262	-15778	-12171	-11514	**-18411**	**-18016**	11.659	22.303
	100	-32854	-30841	-24535	-22835	**-35535**	**-34632**	8.903	34.811
	200	-59440	-58118	-47483	-44980	**-65880**	**-64368**	10.302	22.015
Rastrigin	30	18.769	35.180	90.180	129.927	**16.141**	**30.555**	4.725	56.078
	50	144.350	159.983	467.228	536.967	**96.676**	**147.285**	1.290	26.909
	100	355.272	398.947	1020.308	1115.391	**234.206**	**280.601**	7.932	36.907
	200	838.478	907.235	2162.628	2353.384	**458.595**	**541.145**	20.615	61.021
Ackley	30	6.037	8.169	7.691	16.911	**0.231**	**1.772**	58.213	36.810
	50	10.609	12.483	19.923	19.930	**5.335**	**6.822**	14.892	50.080
	100	12.825	13.621	19.923	19.930	**6.736**	**8.197**	18.019	46.156
	200	13.927	14.501	19.925	19.931	**7.605**	**8.441**	27.032	70.353
Griewank	30	3.642	9.014	2.064	5.826	**1.101**	**1.434**	25.258	19.911
	50	53.643	72.828	25.408	39.197	**8.034**	**13.328**	11.365	6.976
	100	155.753	211.611	67.133	80.182	**29.081**	**35.830**	18.913	12.919
	200	385.648	484.405	127.770	161.764	**86.818**	**113.142**	19.470	4.391
Penalty 1	30	6.024	125269	15722	1422426	**0.336**	**1.983**	1.900	5.799
	50	767251	5781793	1342867	4503053	**12.175**	**101.366**	5.194	5.923
	100	4319626	16831372	5053622	11357689	**50.224**	**556.008**	5.718	5.933
	200	29233477	57270741	3980933	10125942	**238.834**	**9106.955**	6.687	6.496
Penalty 2	30	248	362503	172745	5265971	**3.573**	**31.627**	5.919	12.768
	50	6234168	18773181	6881823	16165647	**296.413**	**45020.198**	6.299	6.449
	100	54117223	97087010	15026298	34967350	**45358**	**123262**	7.160	6.761
	200	114373923	194879393	17566913	35550460	**236611**	**727595**	9.840	8.964
Fletcher	30	24917	68364	40390	105468	**4829**	**28120**	14.217	18.507

50	789748	1154052	1844496	2626204	**499324**	**646130**	6.500	9.428
100	3500122	4555957	10863455	12910759	**2164663**	**2974089**	5.744	20.746
200	16068045	20570768	45431035	53360024	**9868405**	**14269770**	5.199	16.630

Table 4
Success rate and Average minimum number of generation required for benchmark problems considering discrete design variables. **'Bold value'** indicates better value found.

Benchmark Functions	BBO		ABC		HBBABC	
	SR*	AMG*	SR	AMG	SR	AMG
Sphere	0.550	400	NR	NR	**1**	**110**
Schwefel 2.22	0.360	300	0.920	45	**1**	**40**
Schwefel 1.2	**NR***	**NR***	NR	NR	NR	NR
Schwefel 2.21	**NR***	**NR***	NR	NR	NR	NR
Rosenbrock	NR	NR	NR	NR	**NR***	**NR***
Step	NR	NR	NR	NR	**1**	**160**
Quartic	**1**	180	NR	NR	**1**	**150**
Schwefel 2.26	NR	NR	NR	NR	**NR***	**NR***
Rastrigin	0.520	**210**	NR	NR	**0.620**	350
Ackley	NR	NR	0.900	350	**1**	**50**
Griewank	NR	NR	**1**	310	**1**	**90**
Penalty 1	NR	NR	NR	NR	**1**	**210**
Penalty 2	NR	NR	NR	NR	**1**	**220**
Fletcher	NR	NR	NR	NR	**NR***	**NR***

SR* Success Rate
AMG* Average number of Maximum Generations Required
Table 5
Comparison of HBBABC with other optimization techniques. **'Bold value'** indicates better value found.

	Mean			Best		
	ACO	**SGA**	**HBBABC**	**ACO**	**SGA**	**HBBABC**
Griewank	-	7.63059	**1.434**	-	2.4531	**1.1006**
Penalty 1	67850279.99	7.98258	**1.982893**	**0.23161**	-	0.33582
Penalty 2	159465175.9	17083.97	**31.62655**	**0.24281**	-	3.5728
Quartic	-	0.023493	**0.000479**	-	0.001395	**0.00005127**
Rosenbrock	-	107.2454	**28.03401**	-	-	-
Schwefel 2.26	-	-8410.10	**-11410.2**	**-3953.9**	-	**-11963.61**
Step	-	618.6	**75.08**	-	222	**16**
Schwefel 1.2	-	-	-	-	**3484.908**	7369
Fletcher	-	-	-	-	20876.64	**4829**
Sphere	-	-	-	-	0.9543	**0.050393**

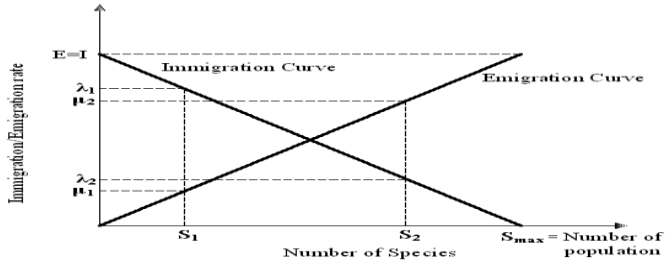
Table 6
Best results , Mean results and Success rate for the engineering optimization problems considering discrete design variables. **'Bold value'** indicates better solution found

		BBO			ABC			HBBABC		
		Best	Mean	SR*	Best	Mean	SR	Best	Mean	SR
Gear Train	N50G100*	6.7E+05	1.6E+08	0.00	2.0E+06	1.9E+08	0.00	**46.20**	**1.3E+07**	**0.08**
	N100G100	43.64	4.6E+06	0.24	40.41	2.2E+07	0.36	**35.36**	**52.49**	**1.00**
Welded Structure	N50G100	55698.95	57160.11	0.04	55326.29	55980.17	0.60	**55326.29**	**55729.30**	**0.72**
	N100G100	55326.29	56877.94	0.12	55326.29	55769.47	0.72	**55326.29**	**55381.83**	**0.96**
10- Truss	N50G100	5556.28	5776.11	0.00	5498.37	5797.23	0.16	**5490.74**	**5594.76**	**0.68**
	N100G100	5559.91	5695.93	0.04	5491.72	5646.59	0.36	**5490.74**	**5513.63**	**0.92**
25- Truss	N50G100	494.83	510.92	0.00	484.85	489.20	0.80	**484.85**	**484.95**	**1.00**
	N100G100	485.77	501.17	0.12	484.85	487.16	0.88	**484.85**	**484.85**	**1.00**

Planetary Transmission	N50G100	0.527	0.629	0.04	0.536	0.556	0.00	**0.526**	**0.536**	**0.48**
	N100G100	0.533	0.607	0.04	0.530	0.538	0.12	**0.526**	**0.529**	**0.84**

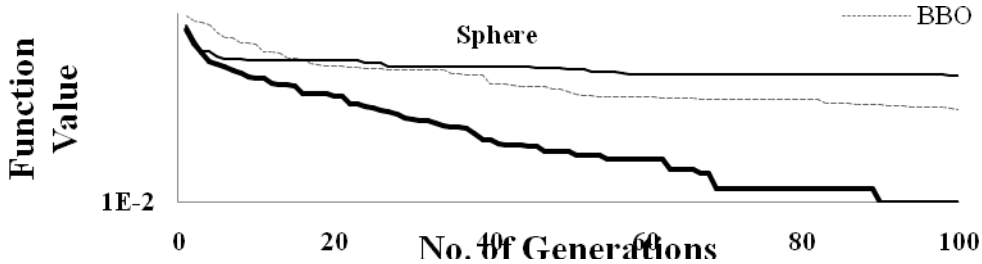
SR* Success Rate
N50G100* Population size = 50, Number of Generation = 100

Figure 1 Species model for single habitat showing two candidate solutions (Simon ,

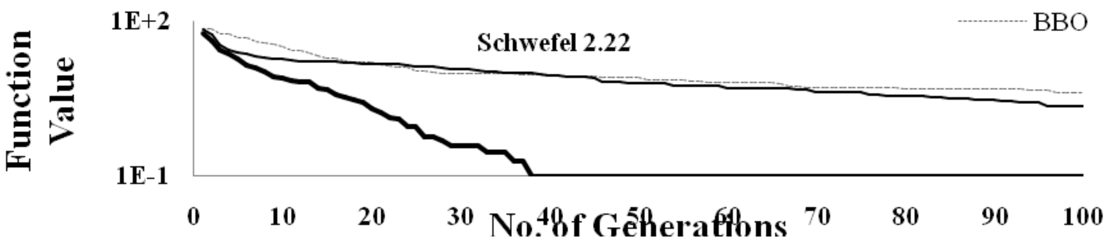


2008)

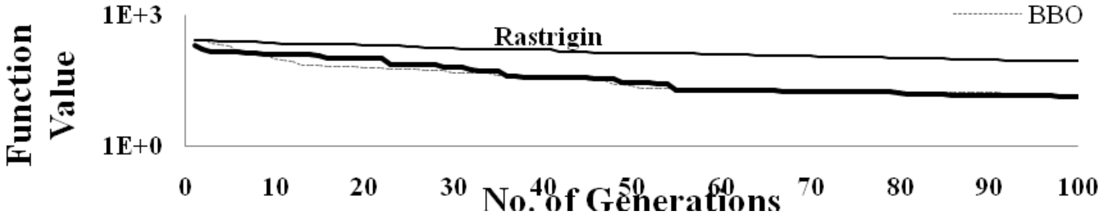
Figure 2
Convergence curve for different benchmark problems considering discrete design variables
(a) Sphere, (b) Schwefel 2.22, (c) Rastrigin (d) Fletcher
(a)



(b)



(c)



(d)