Enhanced Bee Colony Algorithm for Complex Optimization Problems

S.Suriya [1], R.Deepalakshmi [2], S.Suresh kannan [3], Dr.S.P.Shantharajah[4]

[1]*Assistant Professor,*
Velammal college of Engineering and technology,
Madurai.
suriyaS84@gmail.com

[2]*Assistant Professor,*
Velammal college of Engineering and technology,
Madurai.
jei.deepa@gmail.com

[3]*Under Graduate Student,*
SACS mavmm Engineering college,
Madurai.
suresh28kannan@gmail.com

[4]*Prof. and Head,*
Sona college of Engineering and Technology,
Salem.
spshantharaj@gmail.com

Abstract

Optimization problems are considered to be one kind of NP hard problems. Usually heuristic approaches are found to provide solutions for NP hard problems. There are a plenty of heuristic algorithms available to solve optimization problems namely: Ant Colony Optimization, Particle Swarm Optimization, Bee Colony Optimization, etc. The basic Bee Colony algorithm, a population based search algorithm, is analyzed to be a novel tool for complex optimization problems. The algorithm mimics the food foraging behavior of swarms of honey bees. This paper deals with a modified fitness function of Bee Colony algorithm. The effect of problem dimensionality on the performance of the algorithms will be investigated. This enhanced Bee Colony Optimization will be evaluated based on the well-known benchmark problems. The testing functions like Rastrigin, Rosenbrock, Ackley, Griewank and Sphere are used to evaluavate the performance of the enhanced Bee Colony algorithm. The simulation will be developed on MATLAB.

Keywords: *Swarm Intelligence, Optimization, Heuristics.*

I. INTRODUCTION

In computer science, an optimization problem is the problem of finding the best solution from all feasible solutions. More formally [I-1], an optimization problem A is a quadruple (I, f, m, g), where

- I is a set of instances;
- given an instance x€I, f(x) is the set of feasible solutions;
- given an instance x and a feasible solution y of x, m(x,y) denotes the measure of y, which is usually a positive real.
- g is the goal function, and is either min or max.

The goal is then to find for some instance x an optimal solution, that is, a feasible solution y with m(x, y) = g{ m(x, $y^{'}$) | $y^{'}$ Є f(x) }. For each optimization problem, there is a corresponding decision problem that appears as whether there is a feasible solution for some particular measure m_0. For example, if there is a graph G which contains vertices u and v, an optimization problem might be "find a path from u to v that uses the fewest edges". This problem might have an answer, say, 4. A corresponding decision problem would be "is there a path from u to v that uses 10 or fewer edges?" This problem can be answered with a simple 'yes' or 'no'.In the field of approximation algorithms, algorithms are designed to find near-optimal solutions to hard problems. The usual decision version is then an inadequate definition of the problem since it only specifies acceptable solutions. Even though suitable decision problems are introduced, the problem is more naturally characterized as an optimization problem. NP optimization problems: An NP-optimization problem (NPO) is an optimization problem with the following additional conditions.

- the size of every feasible solution yЄf(x) is polynomially bounded in the size of the given instance x,
- the languages { x | xЄI } and { (x,y) | yЄ f(x) } can be recognized in polynomial time, and
- m is polynomial-time computable.

This implies that the corresponding decision problem is in NP. In computer science, interesting optimization problems usually have the above properties and are therefore NPO problems. A problem is additionally called a P-optimization (PO) problem, if there exists an algorithm which finds optimal solutions in polynomial time. Often, when dealing with the class NPO, one is interested in optimization problems for which the decision versions are NP-hard. It is notable that hardness relations are always with respect to some reduction. Due to the connection between approximation algorithms and computational optimization problems, reductions which preserve approximation in some respect are for this subject preferred than the usual Turing and Karp reductions. An example of such a reduction would be the L-reduction. For this reason, optimization problems with NP-complete decision versions are not necessarily called NPO-complete.

The term heuristic is used for algorithms which find solutions among all possible ones. These algorithms, usually find a solution close to the best one and they find it fast and easily.

The Bee Colony algorithm is a swarm based meta-heuristic algorithm that was introduced by Karaboga in 2005 (Karaboga, 2005) for optimizing numerical problems. It was inspired by the intelligent foraging behavior of honey bees. The algorithm is specifically based on the model proposed by Tereshko and Loengarov (2005) for the foraging behaviour of honey bee colonies. The model consists of three essential components: employed and unemployed foraging bees, and food sources. The first two components, employed and unemployed foraging bees, search for rich food sources, which is the third component, close to their hive. The model also defines two leading modes of behaviour which are necessary for self-organizing and collective intelligence: recruitment of foragers to rich food sources resulting in positive feedback and abandonment of poor sources by foragers causing negative feedback. In Bee Colony algorithm, a colony of forager bees (agents) search for rich artificial food sources (good solutions for a given problem). To apply algorithm, the considered optimization problem is first converted to the problem of finding the best parameter vector which minimizes an objective function. Then, the bees randomly discover a population of initial solution vectors and then iteratively improve them by employing the strategies: moving towards better solutions by means of a neighbour search mechanism while abandoning poor solutions.

II. LITERATURE SURVEY

Swarm Intelligence employs the collective behaviors in the animal societies to design algorithms. Bee Colony algorithm is based on a particular intelligent behavior of honeybee swarms. Bee Colony algorithm is developed purely based on inspecting the behaviors of real bees on finding nectar and sharing the information of food sources to the bees in the hive.

Li-Pie Wong[1] demonstrates solution for travelling salesman problem using bee colony optimization involving evaluation of state transition probability using the parameters namely arc fitness and the distance between the cities i and j respectively. The dance duration of the bee is evaluated using three parameters namely waggle dance scaling factor, profitability score of a bee and bee colony’s average profitability.

D.T.Pham [2] describes the use of Bee colony algorithm for the Environmental/Economic problem. The problem is formulated as a nonlinear constrained multi-objective optimization problem. Simulation results presented for the standard IEEE 30-bus system using the Bees algorithm with Pareto optimality are compared to those obtained using other approaches. The comparison shows the superiority of the Bees algorithm and confirms its suitability for solving the multi-objective problem.

D.T.Pham [3] discusses the application of he Bee colony algorithm to solve feeder arrangement and component placement sequencing problems associated with a printed circuit board assembly machine. The experimental results proves that the Bee colony algorithm outperforms by significant reduction in assembly time compared to that of genetic algorithm and Evolutionary Programming on a benchmark assembly task.

D.T.Pham [4] introduces Bee colony algorithm as a new application to optimize Support Vector Machine (SVM) for the problem of classifying defects in plywood. The goal is to find the best combination of

SVM parameters and data features to maximize defect classification accuracy. The experimental results demonstrate the strengths of the Bee colony algorithm as an optimization tool.

D.T.Pham [5] describes the application of Bee Colony algorithm to combinatorial optimization problem. The experimental results compare Bee colony algorithm results with other existing optimization techniques to demonstrate the efficiency and robustness of the Bee colony algorithm.

D.T.Pham [6] focuses on the application of Bee Colony algorithm for both combinatorial optimization and functional optimization. The experimental results obtained by benchmark problems demonstrating the efficiency and robustness of the Bee colony algorithm.

III. BEE COLONY ALGORITHM

The bee colony optimization algorithm is inspired by the behaviour of a honey bee colony in nectar collection. This biologically inspired approach is currently being employed to solve continuous optimization problems, training neural networks, mechanical and electronical components design optimization, combinatorial optimization problems such as job shop scheduling, the internet server optimization problem, the travelling salesman problem, etc. There are three phases in Bee Colony algorithm namely:

1. The Employed Bee Phase
2. The Onlooker Bee Phase
3. The Scout Bee Phase

The Employed Bee Phase: It stays on a food source and provides the neighborhood of the source in its memory. The Onlooker Bee Phase: It gets the information of food sources from the employed bees in the hive and select one of the food source to gathers the nectar. The Scout Bee Phase: It is responsible for finding new food, the new nectar, sources.

```
1.    Initialization Phase
2.    REPEAT
3.    Employed Bees Phase
4.    Onlooker Bees Phase
5.    Scout Bees Phase
6.    Memorize the best solution
7.    UNTIL(Cycle=Maximum Cycle Number
   or a Maximum CPU time)
```

Table 1: Pseudo code for Existing Bee Colony Algorithm

IV. PROPOSED ENHANCED BEE COLONY ALGORITHM

A global optimization problem can be defined as finding the parameter vector x that minimizes an objective function, f(x), as below:

Minimize $f(x)$, $x=(x_1 , x_2 , \dots , x_i ,\dots, x_{n-1}, x_n) \in R^n$ ------------(1)

which is constrained by the following inequalities and/or equalities:

$l_i \le x_i \le u_i$, where $i=1,\dots,n$, ------------(2)

$g_i(x)\le 0$, for $j=1,\dots,p$ ------------(3)

$h_j(x)=0$, for $j=p+1,\dots\dots\dots$ ------------(4)

f(x)is defined on a search space, **S**, which is a **n** dimensional rectangle in R^n ($S \in R^n$). The variable domains are limited by their lower and upper bounds. This problem is also known as constrained optimization problem. If it is an unconstrained optimization problem, then both p=0 and q=0.

In Bee Colony algorithm, the colony of bees contains three groups of bees: employed bees associated with specific food sources, onlooker bees watching the dance of employed bees within the hive to choose a food source, and scout bees searching for food sources randomly. Both onlookers and scouts are also called unemployed bees. Initially, all food source positions are discovered by scout bees. Thereafter, the nectars of food sources are exploited by employed bees and onlooker bees, and this continual exploitation will ultimately cause them to be exhausted. Then, the employed bee which was exploiting the exhausted food source becomes a scout bee in search of further food sources once again. In other words, the employed bee whose food source has been exhausted becomes a scout bee. In Bee Colony algorithm, the position of a food source represents a possible solution to the problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of employed bees is equal to the number of food sources (solutions) since each employed bee is associated with one and only one food source.

The general scheme of the Bee Colony algorithm is as follows:

```
1. Initialization Phase
2. REPEAT
3. Employed Bees Phase
      Modified fitness function
4. Onlooker Bees Phase
5. Scout Bees Phase
6. Memorize the best solution
7. UNTIL(Cycle=Maximum Cycle Number or a Maximum CPU time)
```

Table 2: Pseudo code for Enhanced Bee Colony algorithm

Initialization Phase :

All the vectors of the population of food sources, x_m 's, are initialized (m=1…SN , SN: the population size) by scout bees and control parameters are set. Since each food source, x_m , is a solution vector to the optimization problem, each x_m vector holds n variables, (x_{mi} , i=1…..n) , which are to be optimized so as to minimize the objective function. The following definition might be used for initialization purpose equation 5:

$X_{mi} = l_i + \text{rand}(0,1)*(u_i-l_i)$ --------------(5) where l_i and u_i are the lower and upper bound of the parameter x_{mi} , respectively.

Employed Bees Phase

Employed bees search for new food sources (v_m) having more nectar within the neighbourhood of the food source (x_m) in their memory. They find a neighbour food source and then evaluate its profitability (fitness). For example, they can determine a neighbour food source $\vec{\upsilon_m}$ using the formula given by equation (6):

$v_{mi} = x_{mi} + \Phi_{mi}(x_{mi}-x_{ki})$ --------------------(6)

where x_k is a randomly selected food source, i is a randomly chosen parameter index and ϕ_{mi} is a random number within the range [-a,a]. After producing the new food source v_m, its fitness is calculated and a greedy selection is applied between v_m and x_m. The fitness value of the solution, $\text{fit}_m(x_m)$, might be calculated for minimization problems using the following formula (7)

$\text{fit}_m(x_m) = \{ 1/(1+ f_m(x_m))$ if $f_m(x_m) \geq 0$

$1+\text{abs}(f_m(x_m))$ if $f_m(x_m)<0 \}$--------(7)

where $f_m(x_m)$is the objective function value of solution x_m . The fitness function is modified as

$\text{fit}_m(x_m) = \{ f_m(x_m)/(1+ f_m(x_m))$ if $f_m(x_m) \geq 0$

$1+\text{abs}(f_m(x_m))$ if $f_m(x_m)<0 \}$-----------(8)

Onlooker Bees Phase

Unemployed bees consist of two groups of bees: onlooker bees and scouts. Employed bees share their food source information with onlooker bees waiting in the hive and then onlooker bees probabilistically choose their food sources depending on this information. In Bee Colony algorithm, an onlooker bee chooses a food source depending on the probability values calculated using the fitness values provided by employed bees. For this purpose, a fitness based selection technique can be used, such as the roulette wheel selection method (Goldberg, 1989). The probability value p_m with which x_m is chosen by an onlooker bee can be calculated by using the expression given in equation (9):

$P_m = \text{fit}_m(x_m) / \Sigma_{m=1}^{SN} \text{fit}_m(x_m)$ -----------------------(9)

After a food source x_m for an onlooker bee is probabilistically chosen, a neighbourhood source v_m is determined by using equation (6), and its fitness value is computed. As in the employed bees phase, a greedy selection is applied between v_m and x_m . Hence, more onlookers are recruited to richer sources and positive feedback behaviour appears.

Scout Bees Phase

The unemployed bees who choose their food sources randomly are called scouts. Employed bees whose solutions cannot be improved through a predetermined number of trials, specified by the user of the Bee Colony algorithm and called “limit” or “abandonment criteria” herein, become scouts and their solutions are abandoned. Then, the converted scouts start to search for new solutions, randomly. For instance, if solution x_m has been abandoned, the new solution discovered by the scout who was the employed bee of x_m can be defined by equation(5). Hence those sources which are initially poor or have been made poor by exploitation are abandoned and negative feedback behaviour arises to balance the positive feedback.

V. EXPERIMENTAL RESULTS

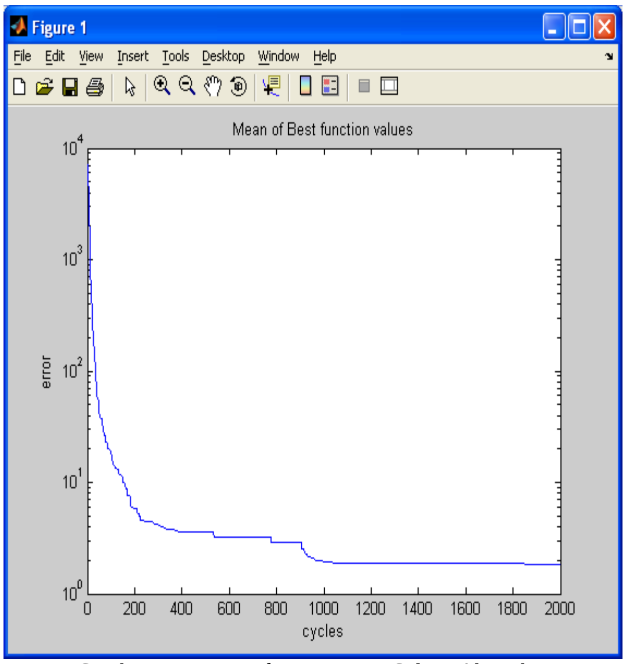
Existing Bee Colony System:

The performance of the existing bee colony system is investigated by applying the algorithm to testing functions like Rosenbrock, Rastrigin, Sphere , Griewank and Ackley respectively

S.No	Testing Function	Mean & Standard Deviation	D=5	D=10	D=30	D=50	D=100
1	**Rosenbrock**	Mean	**0.07140**	**1.08633**	**9.50849**	**63.269**	**305.03**
		Std. Dev.	**0.08736**	**1.61697**	**10.3805**	**44.1715**	**141.723**
2	**Rastrigin**	Mean	**0**	**2.10e-012**	**1.98992**	**10.2956**	**38.1608**
		Std. Dev.	**0**	**3.66e-012**	**5.822e-006**	**3.76604**	**4.03784**
3	**Sphere**	Mean	**3.07e-017**	**1.95e-016**	**1.3508e-015**	**5.1e-014**	**0.00580**
		Std. Dev.	**2.05e-017**	**9.68e-017**	**6.7474e-016**	**8.1e-014**	**0.01004**
4	**Griewank**	Mean	**2.32e-006**	**4.40e-011**	**4.2138e-007**	**2.2e-010**	**4.67e-005**
		Std. Dev.	**4.02e-006**	**7.60e-011**	**7.2952e-007**	**3.4e-010**	**7.55e-005**
5	**Ackley**	Mean	**4.44e-015**	**3.52e-014**	**0.00021105**	**0.586847**	**0.73723**
		Std. Dev.	**0**	**3.48e-014**	**0.000365563**	**0.507194**	**0.268632**

Table 3: Existing Bee Colony System

The above table3 shows the mean and standard deviation values for each testing functions described above by varying the dimensionality parameter D for 5, 10, 30, 50 and 100 respectively. The graph revealing the error rate of the existing bee colony algorithm is as follows:



Graph 1: Error Rate of Existing Bee Colony Algorithm

The above graph reveals the error rate of existing bee colony algorithm. The error rate is a term that describes the degree of errors encountered. Higher the error rate then the performance is slow. The error rate gets saturated as it approaches 1000th cycle nearing to 10^0 .

Enhanced Bee Colony System:

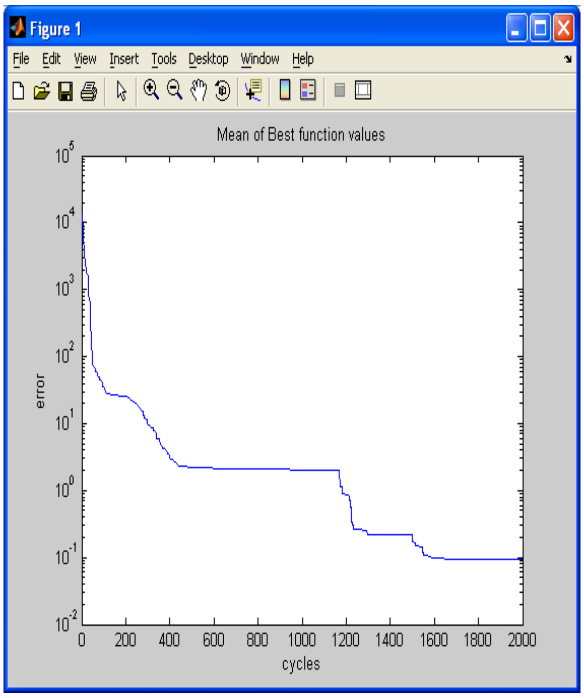
The performance of the enhanced bee colony system is also investigated by applying the algorithm to testing functions like Rosenbrock, Rastrigin, Sphere, Griewank and Ackley respectively. The same procedure is followed to determine the performance of Enhanced Bee Colony system.

The table4 shows the mean and standard deviation values for each testing functions described above by varying the dimensionality parameter D for 5, 10, 30, 50 and 100 respectively.

S.No	Testing Function	Mean & Standard Deviation	D=5	D=10	D=30	D=50	D=100
1	**Rosenbrock**	Mean	**0.4350**	**0.702883**	**11.9789**	**71.2483**	**300.724**
		Std. Dev.	**0.689928**	**0.910426**	**9.92203**	**47.987**	**78.4931**
2	**Rastrigin**	Mean	**0**	**2.17e-012**	**.12003**	**9.7392**	**41.2585**
		Std. Dev.	**0**	**3.75e-012**	**1.02276**	**3.62295**	**3.56687**
3	**Sphere**	Mean	**3.66e-017**	**1.93e-016**	**7.28e-015**	**3.02e-014**	**1.66e-005**
		Std. Dev.	**3.18e-017**	**8.5e-017**	**8.95e-015**	**1.96e-014**	**1.31e-005**
4	**Griewank**	Mean	**3.70e-017**	**5.18e-016**	**2.0e-012**	**1.62e-010**	**5.22e-005**
		Std. Dev.	**6.40e-017**	**7.05e-016**	**3.e-012**	**2.76e-010**	**8.74e-005**
5	**Ackley**	Mean	**4.44e-015**	**2.57e-014**	**0.00115**	**0.435242**	**0.547046**
		Std. Dev.	**0**	**6.15e-015**	**0.00200**	**0.439589**	**0.187945**

Table 4: Modified Bee Colony System

The graph 2 reveals the error rate of the enhanced bee colony algorithm is as follows: the error rate even after reaching 1000th cycle nearing 10^0 does not saturate it approaches towards 10^{-1} to saturate near 1600th cycle respectively. This shows how the enhanced Bee Colony algorithm outperforms the existing Bee Colony algorithm.



VI. CONCLUSION

The experimental results prove that the enhanced bee colony algorithm has some effective role on solving complex optimization problems. It works more efficiently than the original bee colony algorithm. The experimental proves how effectively the enhanced bee colony algorithm outperforms the existing bee colony algorithm.

REFERENCES

[1] Li-Pei Wong, Malcolm and Yoke Hean Low, Chin Soon Chong, “ A Bee Colony Optimization algorithm for Travelling salesman Problem” , Second Asia International Conference on Modelling and Simulation, 2008.

[2] D.T.Pham, J.Y.Lee, A.Haj Darwish and A.J.Soroka, “Multi-objective Environmental/ Economic Power Dispatch using the Bees Algorithm with Pareto optimality”, 4th International Virtual Conference on Intelligent Production Machine and Systems, 2008.

[3] D.T.Pham, S.Otri, A.Haj Darwish, “Application of the Bees Algorithm to PCB assembly optimization”, 3rd International Virtual Conference on Intelligent Production Machine and Systems, 2007.

[4] D.T.Pham, Z.Muhamad, M.Mahmuddin, A.Ghanbarzadeh, E.Koc, S.Otri, “Using the Bees Algorithm to Optimize a Support Vector Machine for Wood Defect Classification”, International Virtual Conference on Intelligent Production Machine and Systems, 2007.

[5] D.T.Pham, E.Koc, J.Y.Lee, J.Phrueksanant, “Using the Bees Algorithm to Schedule jobs for a machine”, Proceedings of 8th International Conference on Laser Methodology, CMM and Machine Tool Performance, LAMDAMAP, page 430-439 , 2007.

[6] D.T.Pham, A.Ghanbarzadeh, E.Koc, S.Otri, S.Rahim, M.Zaidi, “The Bees Algorithm – A Novel Tool for Complex Optimization Problems”, International Virtual Conference on Intelligent Production Machine and Systems, 2006.

[7] Internet references from various referred sites.