Solution of Unconstrained Non-Linear Programming Problems using Differential evolution, Genetic algorithm and Artificial bee colony evolutionary algorithms: A comparative study

Priyavada[#1], Binay Kumar [*2]

[1, 2] Department of Mathematics, Lingaya's Vidyapeeth, Faridabad (Delhi NCR)
[1] priyavada.parihar@gmail.com
[2] binay@lingayasvidyapeeth.edu.in
*Corresponding author: Dr. Binay Kumar

Abstract
Evolutionary computation (EC) is a set of global optimization algorithms inspired by biological evolution. EC techniques, which are based on principle of evolution and survival of the fittest Darwinian Theory. It is a sub-field of Artificial Intelligent techniques and soft computing. In Evolutionary computation, we use non classical methods based on natural evolution instead of classical methods. When fitness function has several local extreme or not known derivative then it is very difficult to use classical methods for solution of optimization problem in such situation non classical methods gives very convenient and faster solution. In this paper we present Genetic algorithms, Artificial bee colony algorithm and Differential evolution algorithm of Evolutionary computation technique for solution of unconstrained non-linear programming problem. Numerical examples have solved using software MATLAB and the result has compared among evolutionary algorithms as Genetic algorithm, differential algorithm and artificial bee colony algorithm.

Keywords: Evolutionary computation, Genetic algorithm, Differential evolution algorithm and Artificial bee colony algorithm.

1. Introduction

Last few decades there are a lot of evolutionary algorithms subset of evolutionary computational approach have been proposed and used to many real world complex optimization problems and got the success in finding the near best optimal solution. Solution of non-linear optimization problems is one of the most important issues in many field like complex research problem and real world problems. Evolutionary algorithm uses the computational model of evolutionary processes as key element and share a usual concept based on the processes of selection and reproduction. Mostly all evolutionary algorithms are inspired by the natural concept of Darwinian Theory and this theory tells that EAs are not static but dynamic as they can evolve over time. Evolutionary algorithms are population based local search algorithm. During the last twenty years most of the research has done in the complex objective function without using constraints. Genetic algorithm, differential evolution, Particle swarm optimization, Artificial Bee Colony algorithm, Ant colony optimization

and Spider monkey optimization etc. are the evolutionary algorithms and all type of non-linear complex optimization problem can be easily solved by these algorithms. Among all Differential evolution is very easy and effective to apply. GA is evolutionary algorithm based on the theory of natural selection of Charles Darwin. GA and DE both uses same operator's mutation, crossover and selection but in GA crossover is used first then mutation while in DE mutation is used first then crossover. DE algorithm solved continuous optimization problems first and depends on the differentials of two vectors. ABC algorithm introduced by Dervis Karaboga in 2005. It is inspired by the behaviour of natural honey bees and uses three main phases employed bee phase, onlooker bee phase and scout bee phase. In ABC algorithm the aim of the bees is to find the best place of food sources with high nectar amount.

2. Literature Review

2.1 Genetic algorithms

Nuno Neves et al. [1] discussed the study of NON-LINEAR Optimization problem using a distributed GA. GA is used in large space for the optimization problem. Dr. Yuyi Lin et al. [2] proposed solution of non-linear optimization problems using GA with penalty method. This paper discussed the techniques of applying GAs with penalty function to solve highly constrained nonlinear engineering design problems. Subrata Datta et al. [3] proposed the efficient genetic algorithm on programming problems for fittest chromosomes. Chhavi Mangla et al. [4] proposed the genetic algorithm based optimization problem for solution of system of non-linear equations. D.E technique is applied for the solution of non-linear optimization problem. This type of problems is changed first in to multi objective problem then a new fitness function is used further process and both result is compared. Dr. Amanpreet Singh et al. [5] discussed some special type of Unconstrained Optimization Problem with using G.A and main focus of G.A is on the simulation of natural evolutionary process which can use for certain standard optimization problems. It is evolutionary algorithm which works according to the basis of natural genetics on a population string structures. The global and population based optimization technique is used to avoid the problem like stagnation of the solution to a local minimum. Punam S Mhetre et al. [6] introduced G.A for the solution of both linear and Non-linear optimization problems and check the major benefits acquired as a result of using G.A. Gauss- Legendre integration method as a technique is used for the solution of NLPP and to get the result without changing the non -linear equation to linear. G.A result are very good for all tasks requiring optimization and highly successful in any situation where many variables interact to produce a large number of possible solutions. T Yokota et al. [7] discussed non- linear optimization problems of integer programming and also its applications. T. Yokota also introduced the new method for the solution of NLPP for better comparison. Some methods and holding property are discussed with the help of G.A. From the result it is observed that proposed method is best then the property. Chhavi Mangla et al. [8] discussed solution of non- linear optimization problem by using Genetics algorithm. Both type of problems singles and multi-objective optimization problem based on engineering, social sciences and medical sciences are solved by using standard benchmark problems. This paper carries out a comprehensive review to analysis the work of GA.

2.2 Artificial Bee Colony Algorithm

Lalit Kumar et al. [9] proposed solution of NP-hard Problem using ABC algorithm. ABC is considered as a newest nature-inspired swarm intelligence-based optimization algorithms. S. T S. Talatahari et al. [10] discussed solution of Parameter Identification of Nonlinear Problems by Artificial Bee Colony Algorithm. Dervis Karaboga et al. [11] discussed the ABC algorithm for solution of non -linear unconstraint optimization problem. Modified result of ABC algorithm has been compared to state of art algorithm. Result of modified ABC algorithm are tested for real engineering problems and found that it can efficiently applied for solving constrained optimization problem. Weifang Gao et al. [12] introduced a framework of constraint optimization problem by novel ABC algorithm. Bi -objective optimization problem is taken in which first part remains same (objective function) and the second part is the degree of constraint. Multi-strategy technique is used which consists three types of strategies, and play the main role between diversity and convergence. Soudeh Babaeizadeh et al. [13] discussed Constrained based ABC Algorithm for Optimization Problems. It is an optimization algorithm and compared to other optimization based algorithms. ABC has one poor exploitation ability that can be removed by using three types of phases employee bees, onlooker bees and scout bees phase by a new search technique. Soudeh Babaeizadeh et al. [14] discussed non-linear optimization problem related to enhanced Constrained ABC Algorithm. From last few years ABC with less parameters and strong global optimization ability researcher paid their attention on these two problems. But still there is one drawback in ABC algorithm that is poor in exploitation. Numerical examples results are shown with several benchmark functions and result are compared. Nadezda Stanarevic et al. [15] proposed improved ABC algorithm for constrained optimization problem. Artificial bee colony algorithm is applicable only on unconstrained optimization problems but later a lot of modification in ABC was introduced of constraints problems. ABC modified version introduced for constraint optimization problem.

2.3 Differential Evolution Algorithm

Jouni Lampinen et al. [16] discussed the mixed variable non- linear optimization by differential evolution. This paper solving non-linear programming problems containing integer, discrete and continuous variables. Md. Abul Kalam Azad et al. [17] proposed the solution of non -linear continuous global optimization problems by modified differential evolution. This paper discussed about modified differential evolution in which some self -adaptive parameter is introduced. Mutation is modified and then use its inversion operator. Some non-linear continuous optimization problems are tested. Inversion operator has used in this proposed differential evolution algorithm. Musrrat Ali et al. [18] discussed simpler differential evolution algorithm. Simple and modified version of differential evolution algorithm (NSDE) is presented. In simple DE & NSDE there is only one difference of initial population between them. NSDE uses non -linear simplex method to create the initial population. Twenty benchmark problem are taken with box constraint and numerical result are compared with traditional DE and other DE. Zhang Xiao Fei et al. [19] proposed different concept related to reactive power optimization of power system based on niching differential evolution algorithm, As the problem is non -linear integer type. In this Zhang introduced the niche theory and some improvement in the differential evolution which aim of the modified RPO. Local search ability and broaden search range also increased. Different degrees of improvement are also used for speed and precision of convergence. Chiha Ibtissem et al. [20]

discussed differential evolution algorithm for special multi -objective optimization problems. A new variant is discussed for DE. Proposed algorithm named based cost differential evolution generate the mutant vector. This vector is created by adding a weighted difference of two cost functions. Then performance of Bc.DE is calculated on several bound constrained non -linear and non -differentiable numerical problems and compare with the DE and other DE variants. Jeerayut Wetweerapong et al. [21] Proposed an improved differential evolution algorithm with a restart technique to solve system of non -linear equations, which often used for the solution of complex computational problems with variables. In DE –R a new approach is used for mutation and restart technique to stop premature convergence. Proposed technique is applied on various world and synthetic problems and compare with other methods, it found that DER gives fast convergence and high quality solutions. Saeed Nezhadhosein et al. [22] Introduced integrating differential evolution algorithm with modified hybrid GA for solving non -linear optimal control problems. Two phase algorithm is used based on IDE. Associate non- linear problems with hybrid GA of optimal control problem is discussed. In First phase DE is completely depends on initial population. In 2nd phase MHGA starts with new population.

3. Evolutionary Computational Approach

3.1 Genetic Algorithm

Genetic Algorithms are metaheuristic optimization method and based on the biological theory introduced by Charles Darwin and provides an extended concept survival of the fittest [23]. GA is developed by Professor Johan Holland along with his students and colleagues at the University of Michigan during the 1960s and 1970s. [24] Dr Johan Holland then developed the theoretical framework for applying genetic algorithm many complex computational problems needed searching through huge search domain. In that cases G.A gives best and efficient tool to reach at an optimal solution and gives an efficient way for solving computationally hard non-linear optimization problems. GA starts with a set of possible solutions where each solution in GA, is represented as a chromosome. Set of operators are applied to the initial set of solution space. GA does not work on a single trial solution at a time but work on entire population. In special generation, the whole population comprises to the set of solution then select the fittest member and share their qualities. GA is based on the principal of Natural selection and applied to discrete optimisation problem. It is frequently used to find optimal or near optimal solution to difficult problem. GA can solve both types of problems unconstrained and constrained optimization problems [25].

GA has three main operators:

3.1.1 Selection Operator

This operator gives preference to better solution allowing them to pass on their genes to the next generation of the algorithm. The main objective is to select the good solution and discard the bad solution. The individual with a higher fitness must have a higher probability of having offspring. There are different selection techniques in GA.

(a) **Roulette Wheel Selection.** Here selection is done by Roulette wheel**.**
(b) **Rank Based Selection:** Rank the all individual firstly and then give the fitness ranking to each.

(c) **Tournament Selection**: A group of n individuals is randomly selected from the current population and from them best is selected. Here n is the tournament size the higher n the higher the pressure to select above average quality individual. A lot of tournaments are held between the individual and this process is repeated as often as desired.

3.1.2 Crossover Operator

The crossover between two parents string produce offspring by swapping genes of the chromosomes. With the help of crossover, we can combine partial solution from different candidate.

 In GA three types of crossover are used

 1-point Crossover,

 2-point crossover and

 3-Uniform- Crossover.

3.1.3 Mutation Operator

 GA get better solution by using this operator. This operator is used to maintain genetic diversity from one generation of population to the coming generation. After mutation the solution may change entirely from the last solution. Mutation operator changes 1 to 0 or 0 to 1.

 Bit Flip Mutation: We can select one or more random bits and flip them to each other

 Swap Mutation: In swap mutation we select two position on the chromosome at random and interchange the values

Scramble Mutation: This is also famous with permutation representation choose, select a subset of genes and shuffled their values.

Inverse Mutation: In Inverse mutation, we select a subset of genes and inverse that selected subset.

3.2 Artificial Bee Colony Algorithm

 ABC is same as swarm based intelligence, metaheuristic and global optimization algorithm and advanced proposed algorithm by Dervis Karaboga in 2005. Dervis Karaboga with his research members studied on ABC algorithm and some of its applications. The first journal article of ABC was presented by Karaboga and Basturk in 2007 in which the performance of ABC was compared to GA and PSO. First conference paper is published in 2006. In ABC algorithm [26, 27], the swarm of artificial bees are categorised in three groups of bees: employed bees, onlookers and scout bee. 50% are the employed bees and 50% are onlooker's bees to the total bees and only 1 and 2 are scout bees. The total numbers of employed bees are equal to the number of onlooker's bees. Every food source is a possible solution to the optimization problems. Here the main aim of the bees is to find

the best place of food source with high amount of nectar. Fitness of bees is determined in terms of best quality of the food source. Initially first group of employed bees goes to random places for searching the food. At that time onlooker bees are at bee hive. They are waiting for the employed at the hive. When the employed returns to beehive then they share all the information like quality, quantity and distance with onlooker bees regarding the food source. After receiving all information from the employed bees the onlooker bees also start searching in neighbourhood of same food source. Food sources which are abandoned changed by scout bees in to new randomly generated food source.

Behaviour of Honey Bee in Nature

ABC algorithm is inspired by honey bees, where honey bee's swarm works in collective intelligence way, while searching the food. Bees have many qualities, like bees can communicate in their own language, can memorize all the information and take best own decisions based on that gathered information. By the intelligence behaviour of bees, many researchers motivate to simulate above foraging behaviour of the bees. The nature of honey bees is listed as

Employed Bees: Every employed Bee exploit only single food source and store every information regarding that food source like richness, direction and distance.

Unemployed bees: These bees divided in to two groups of bees, onlooker and scout bees. Only these bees found and shape out all information from 1st group of bees based on certain probability.

Foraging behaviour: Foraging behaviour is the important characteristic of the bees. In this process, bees go away from their hive and starts searching best food with high nectar amount, extracts all nectar and stores in her abdomen. She can store nectar till 30-120 minutes, on the basis of nectar quantity and distance. Finally, they share all information.

Four Phases of ABC Algorithm

1. **Initialization Phase:** In this phase we select the food sources randomly with the expression.

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

Here x_{min}^j & x_{max}^j are the upper and lower bound of the solution space (Domain) of objective function, rand (0 1) is a random number lies between [0 1]. And fitness function is determined by this formula

Evaluate the fitness function of all new solution with the help of this formula.

$$fit = \{\frac{1}{1+f}, \; if \; f \geq 0 \; 1 + fabs(f), if \; f < 0$$

2. **Employed Bee Phase**: Three steps are used mainly in this phase a) Generate a new solution

b) Calculate new fitness, c) Apply greedy selection

The neighbourhood food source is determined and calculated by the following equation.

$$x_{new}^{j} = x^{j} + \phi (x^{j} - x_{p}^{j}), \phi \in [-1,1]$$

Where x = Current solution

x_{p}^{j} =Random partner, x_{new}^{j} =New solution

f=Objective function value of a solution.

3. Onlooker Bee phase

Firstly, calculate the probabilities of every food source. Then generate a new solution that depends on probability value after that Calculate new fitness by applying greedy selection

$$\text{Probability} = 0.9 \times \frac{fit_i}{fit} + 0.1$$

Here fit_i is the fitness value of the solution i. Onlooker bees search the neighborhood food sources according to this expression.

$$x_{new}^{j} = x^{j} + \phi (x^{j} - x_{p}^{j})$$

4. Scout Bee Phase

First of all, found the abandoned solution which based on limit value. Then discard that abandoned solution and generate a new solution randomly to replace that. The new food sources are randomly generated by the scout bees. The new solution will be discovered by the scout bees by using the expression.

$$x_{i}^{j} = x_{min}^{j} + \text{rand } (0, 1) \times (x_{max\_}^{j} x_{min}^{j})$$

3.3 Differential Evolution Algorithm

Differential Evolution Algorithm is very simple and easy stochastic optimization technique for the solution of non- linear optimization problems which evolves a population of potential solutions (individuals) in order to increase the convergence to optimal solutions. DE is population based stochastic search algorithm developed by Storn and Price in 1995 for the solution of Chebyshev fitting problems [28]. DE is different from other algorithm in the sense that distance and direction from current population is used to guide the search process. DE has evolutionary operators like mutation, crossover and selection and totally depends on difference of vectors.DE is not nature inspired algorithm like PSO, ABC etc. But in DE global search algorithm employ investigate to allow escaping from local minima. Here three main operators are used mutation, crossover and last selection. The special work of DE algorithm is generating a trial vector, for this three different vectors are selected from the initial population and got the trial vector by adding a weighted difference of two randomly selected solutions to a third one which is target vector. This process is known as mutation operator of evolutionary algorithm. After mutation operator vectors are merged

with another vector and this operation is called crossover operators. After this process we get a new trial vector and this trial is accepted after applying greedy selection in new population if its fitness function value is better than the target vector fitness value. It evolves generation to generation until the stopping criteria would be met. There is one difference between GA and DE algorithm is the mutation and recombination phase. In DE position of the vectors give important information about the best vectors. Diversity of the present population is determined by the distance between the individuals. If distance is large then steps size are also large for the individuals and they can explore much search space in another side if distance is small, then step size should be small to exploit local areas. Minimum population size in differential evolution should be 4.

Three Main Operators of Differential Evolution

3.3.1 Mutation Operator

In DE trial vector plays an important role and it is created by mutation operator.

Trial vector is created for each solution as follow

Trial Vector=Target vector + Scale factor× (Randomly selected solution1- Randomly selected solution12)

Five mutation strategies are proposed by Storn and Price initially. Later Storn and Price discussed ten different working strategies of DE and these are derived from mutation schemes. Every mutation strategy either used exponential type crossover or the binomial type crossover.

DE/rand/1: $U_{ij} = x_{r_1 j} + F(x_{r_2 j} - x_{r_3 j})$

DE/best/1: $U_{ij} = x_{best j} + F(x_{r_2 j} - x_{r_3})$

DE/target to best/2: $U_{ij} = x_{ij} + F(x_{best j} - x_{ij}) + F(x_{r_1 j} - x_{r_2 j})$

DE/best/2: $U_{ij} = x_{best j} + F(x_{r_1 j} - x_{r_2}) + F(x_{r_3 j} - x_{r_4})$

DE/rand/2: $U_{ij} = x_{r_1 j} + F(x_{r_2 j} - x_{r_3 j}) + F(x_{r_4 j} - x_{r_5})$

3.3.2 Crossover Operator

Crossover is the discrete recombination of parent and trial vector and increase the diversity. Crossover is of two type Binomial and Exponential crossover. Binomial crossover is commonly used by the researchers as it gives best result. Any calculation work is not used in crossover operator only some conditions which are user defined gives the trial vector. Trial vector is generated as

$U_j = \{v_j$, $if\ r \leq p_c\ OR\ j = \delta\ x_j$ $if\ r > p_c\ AND\ j \neq \delta$

Here p_c is the Crossover probability

δ=Randomly selected variable $\delta \in \{1,2,3,4,\dots\dots,D\}$

$r = random\ number\ between\ 0\ and\ 1$

U_j jth variable of trial vector

v_j jth variable of donor vector

x_j jth variable of target vector

3.3.3 Selection operator

In this operator apply greedy selection and find the best fitness vector which can take part in mutation operator to produce best trial vector which will continue in next iteration. It constructs the new generation for the next iteration and apply deterministic selection. The offspring replaces the parent if the fitness of the offspring is better than its parent otherwise the parent survive to the next generation. This ensure that the average fitness of the population does not deteriorate.

4. Non Linear Programming Problems

There are different types of non –linear optimization problems like continuous optimization, Bound constraint, discrete, global derivative free and non -differentiable optimization problems. Among these problems some are solved by traditional method which give the approximate solution result rather than accurate solution and some are solved by evolutionary algorithm which are derivative free algorithm. As everyone know that solution of NLPP are difficult in comparison to LPP. Many traditional methods are available for the solution of LPP and these are easy to apply. There are two types of NLPP constraints and without constraints. NLPP without constraints could be designed with several methods like steepest descent, Newton's method. They are taking less time in comparison to the NLPP with constrains. On the other hand, various methods are available like KKT conditions, Lagrangian multiplier, penalty method, barrier method for the solution of NLPP with constraints and there are special problems of quadratic NLPP that cab solved by Beal's method, Wolfe's method etc. Special method is not available for the solution of both types of problems LPP and NLPP. Many non-traditional methods have been introduced for the solution of different optimization problems. These methods are very powerful, easy, populations based and metaheuristic algorithm, by which we can solve NLPP without constrains and with constraints like PSO, DE, ABC and GA to implement for the solution of complex research problems and others problems. DE is very easy to apply because this algorithm used less parameters to others algorithm. In NLPP the function may be convex or non-convex. A convex optimization problem maintains the properties of a linear programming problem and a non-convex problem the properties of a nonlinear programming problem. The basic difference between the two categories is that in a) convex optimization there can be only one optimal solution, which is globally optimal or you might prove that there is no feasible solution to the problem, while in b) non convex optimization may have multiple locally optimal points and it can take a lot of time to identify whether the problem has no solution or if the solution is global.

Suppose we are given an NLP with constraint

Min. f (x)

Subject to linear or non-linear constraints

$g(x) \leq 0;$ and $h(x) = 0;$

5. Numerical Examples

1.objective function (min) = $x_1 - x_2 + 2x_1^2 + 2x_1 x_2 + x_2^2$

$$0 \leq x_1, x_2 \leq 10$$

2. objective function(min) $= 20x_1 + 26x_2 + 4x_1x_2 - 4x_1{}^2 - 3x_2{}^2$

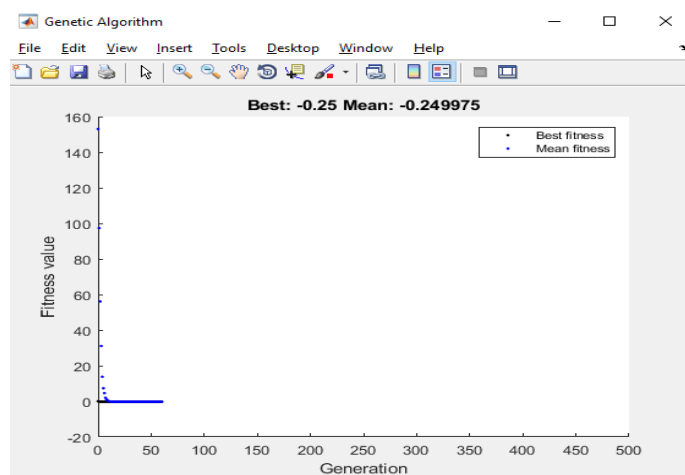$$-10 \leq x_1, x_2 \leq 10$$

Genetic Algorithm problems

1. Objective function (min) $= x_1 - x_2 + 2x_1{}^2 + 2x_1 x_2 + x_2{}^2$

$$0 \leq x_1, x_2 \leq 10$$

1ˢᵗ Table of G.A

| S.No	Population	No. of Variable	Generation	Function Value	X1	X2
1	10	2	50	-0.22185	0	0.667
2	20	2	10	-0.25	0	0.5
3	50	2	50	-0.2599	0	0.5
4	100	2	50	-0.25	0	0.5
5	100	2	100	-0.25	0	0.5
6	500	2	100	-0.25	0	0.5
7	500	2	500	-0.25	0	0.5

Figure 1 (GA)

2. Objective function(min) $= 20x_1 + 26x_2 + 4x_1x_2 - 4x_1{}^2 - 3x_2{}^2$

$$-10 \leq x_1, x_2 \leq 10$$

2nd Table of G.A

S.No	Population	No. of Variable	Generation	Function Value	X1	X2
1	10	2	50	-1072.0593	9.998	-9.292
2	20	2	10	-1144.975	9.971	-9.903
3	50	2	50	-1159.695	9.998	-9.999
4	100	2	50	-1159.999	10	-10
5	100	2	100	-1159.999	10	-10
6	500	2	100	-1159.999	10	-10
7	500	2	500	-1159.999	10	-10

Figure 2 (GA)
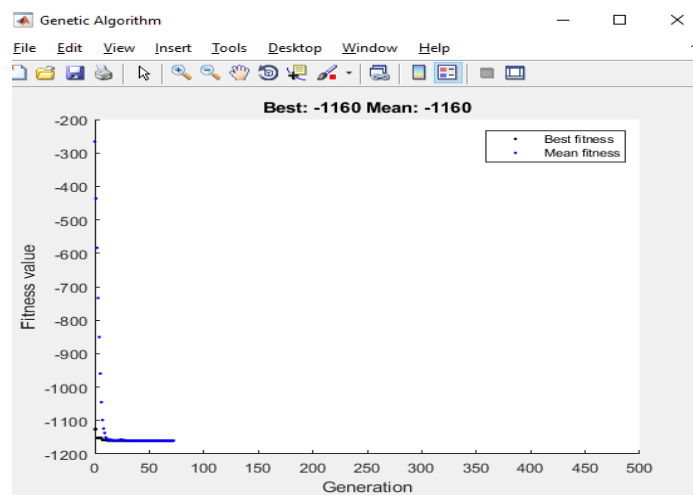
Artificial Bee Colony Algorithm problems

1.Objective function (min) = $x_1 - x_2 + 2x_1^2 + 2x_1 x_2 + x_2^2$

$0 \le x_1, x_2 \le 10$

1st Table of ABC

Pop Size	Variable	Iteration	Function value	X1	X2	Fnew
10	2	50	-0.25(38ite)	0	0.5006	-0.2500
20	2	10	-0.2103(1it)	0	5.0616	20.5585
50	2	50	-0.25(40ite)	0.0011	0.4290	-0.2429
100	2	50	-0.25(10ite)	0	0.3876	-0.25
100	2	100	-0.25(39ite)	0	0.5000	-0.25
500	2	100	-0.25(4ite)	0	0.5000	-0.25
500	2	500	0(2iter)	0	0.5000	-0.25

For ABC solution is -0.2103 in 10 iterations with 20 populations but when iteration and population are increasing ABC gives the best optimal solution is -0.25 near to 0 in 50 iterations and all other iteration with different population. Values of x1 =0 and x2=0.5.

Figure 3(ABC)
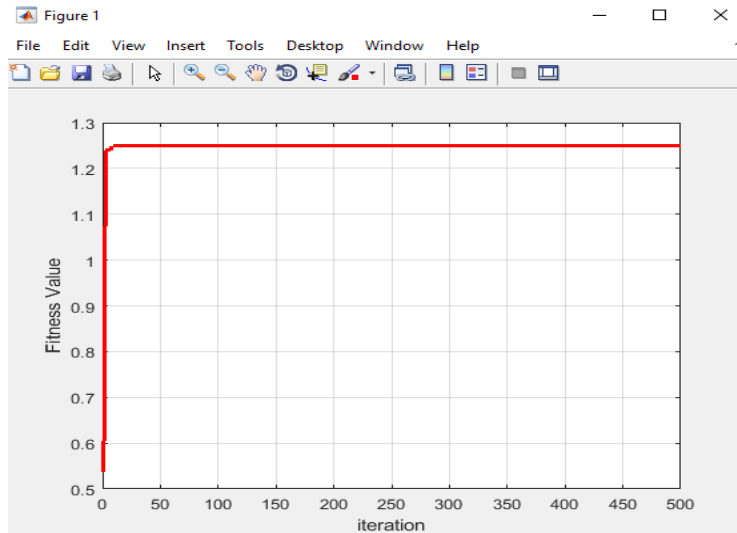
2.Artificial Bee Colony Algorithm problems

2.objective function(min) $= 20x_1 + 26x_2 + 4x_1x_2 - 4x_1^2 - 3x_2^2$

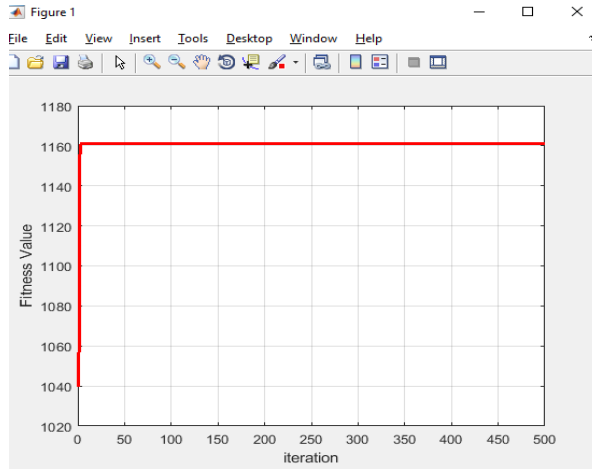$\qquad -10 \leq x_1, x_2 \leq 10$

2ⁿᵈ Table of ABC

Pop size	Variable	Iteration	Function value	X1	X2	fnew
10	2	50	-865(50iter)	-9.9823	-4.5912	-597.5201
20	2	10	-1080.337(5it)	-9.9987	6.4894	-817.0251
50	2	50	-1160(20ite)	10	-8.6092	-990.5670
100	2	50	-1160(8ite)	10	-10	-1160
100	2	100	-1160(6ite)	-10	-2.5203	-583.7715
500	2	100	-1160(2ite)	-9.7807	10	-1.0095e+03
500	2	500	-1160(3iter)	10	-10	-1160

For ABC solution is -0.2103 in 10 iterations with 20 populations but when iteration and population are increasing ABC gives the best optimal solution is -0.25 near to 0 in 50 iterations and all other iteration with different population. Values of x1 =0 and x2=0.5. In the same way for 2ⁿᵈ problem we got the optimal solution -1160 when we take 50 iterations with 50 populations and gives the best solution in 20 iterations. When iterations are increasing solution obtained in less number of iterations.

Figure 4

3.Differential Evolution Algorithm

1.Objective function (min) = $x_1 - x_2 + 2x_1^2 + 2x_1 x_2 + x_2^2$

$0 \leq x_1, x_2 \leq 10$

1st Table of DE

Pop Size	Variable	Iteration	Best Cost	X1	X2	New Sol
5	2	50	29.2154(54)	2.6437	2.0023	2.6437, 2.0023
20	2	10	-0.2498(3)	0.0104	0.5143	0,0.5221
50	2	50	0(1it)	0	0.5000	0,0.5000
100	2	50	-0.25(20ite)	0	0.5000	0,0.5000

In DE result is faster than ABC and GA here optimal solution is near to 0 in 3rd iteration with 20 populations and x1=0.0104, x2= 0.5143. when iterations and population are increasing than we got optimal point in minimum no of iterations.

2.objective function(min) = $20x_1 + 26x_2 + 4x_1x_2 - 4x_1^2 - 3x_2^2$

$-10 \leq x_1, x_2 \leq 10$

2nd Table of DE

Pop Size	Variable	Iteration	Best Cost	X1	X2	New Sol
5	2	50	-1.160(3ite)	10	-10	10,-10
20	2	10	-1160	10	-10	10,-10
50	2	50	-1.160	10	-10	10,-10
100	2	50	-1160(2ite)	10	-10	10,-10

Result of the problems

In GA 1ˢᵗ problem we got the optimal solution -0.25 near to 0 because problem is of minimization in 10 number of iterations with 20 population and values of x1=0, x2=0.5. These values are same for all other iterations in the table. so we get our optimal solution in 10 number of iterations. 2ⁿᵈ problem gives the optimal solution -1154.695 in 10 number of iteration with 20 populations, value of x1 and x2 are near to 10 and -10. These value remain same for all other iteration also in the table.

For ABC solution is -0.2103 in 10 iterations with 20 populations but when iteration and population are increasing   ABC gives the best optimal solution is -0.25 near to 0 in 50 iterations and all other iteration with different population. Values of x1 =0 and x2=0.5. In the same way for 2ⁿᵈ problem we got the optimal solution -1160 when we take 50 iterations with 50 populations and gives the best solution in 20 iterations. When iterations are increasing solution obtained in less number of iterations. In DE result is faster than ABC and GA here optimal solution is near to 0 in 3ʳᵈ iteration with 20 populations and x1=0.0104, x2= 0.5143. when iterations and population are increasing than we got optimal point in minimum no of iterations. In 2ⁿᵈ problem optimal point -1160 is obtained in second iteration when population is 100 and iteration are 50. Value of x1, x2 are 10 and -10 So we can conclude that DE gives faster convergence point in less iterations. Result of DE is better in comparison of ABC and GA.

COMPARISON   TABLE OF 1ˢᵗ problem

Pop	Variable	Gen	GA			ABC			DE		
			X1	**X2**	**F.V**	X1	X2	F.V	X1	X2	F.V
10	2	50	0	0.667	0.22185	0	0.5006	-0.25(38iter)	2.6437	2	29.2154(54ite)
20	2	10	0	0.5	-0.25	0	5.0616	-0.2103(1it)	0.0104	0.5143	-0.2498(3)
50	2	50	0	0.5	-0.2599	0.0011	0.4290	-0.25(40ite)		0.5	0 (1st)
100	2	50	0	0.5	-0.25	0	0.3876	-0.25(10ite)		0.5	0
100	2	100	0	0.5	-0.25	0	0.5000	-0.25(39ite)		0	0
500	2	100	0	0.5	-0.25	0	0.5000	-0.25(4ite)	0	0	0
500	2	500	0	0.5	-0.25	0	0.5000	0(2iter)	0	0	0

COMPARISON   TABLE OF 2nd problem

Population	No. of Variable	Gen	GA			ABC			DE		
			X1	X2	F.V	X1	X2	F.V	X1	X2	F.V
10	2	50	9.998	-9.292	-1072.059	-9.9823	-4.5912	-865(50iter)	10	-10	-1.160(3ite)
20	2	10	9.971	-9.903	-1144.975	-9.9987	-6.4894	-1080.3371(5it)	10	-10	-1160
50	2	50	9.998	-9.999	-1159.695	10	-8.6092	-1160(20ite)	10	-10	-1.160
100	2	50	10	-10	-1159.999	10	-10	-1160(8ite)	10	-10	-1160(2iter)
100	2	100	10	-10	-1159.999	-10	-2.5203	-1160(6ite)	10	-10	-1160
500	2	100	10	-10	-1159.999	-9.7807	10	-1160(2ite)	10	-10	-1160
500	2	500	10	-10	-1159.999	10	-10	-1160(3iter)	10	-10	-1160

7.Conclusion

This paper discussed the solution of unconstrained non-linear optimization problems by using evolutionary computational approaches Genetic algorithm, Artificial bee colony algorithm, and Differential evolution by using MATLAB. For this comparison unconstrained non-linear optimization problems are solved and result are compared. According to 1st comparison table result of DE at 50 number of iteration and 50 populations is 0 in 1st iteration but GA and ABC result are – 0.2599 and -0.25 in 40 number of iteration, so result is faster than both algorithms. In 2nd comparison table we can see result is better in with 20 particles and 10 number of iterations DE result is -1160 which is minimum but GA and ABC result are -1144.975 and -1080. So DE result is better in comparison GA and ABC algorithm in less number of iterations. These algorithms are computational algorithms and efficient for the solution of non-linear optimization problems. From tables and figures it is concluded that Differential evolution gives better result in less number of iterations. Optimal solution obtained from the DE algorithm is much better than Genetic algorithm and ABC algorithm.

References:

[1] Nuno Neves, Anthony-Trung Nguyen, Edgar L. Torres. A study of non-linear   optimization problem using a distributed Genetic algorithm, in proceedings of the 25th International Conference on Parallel Processing, August 1996.

[2] Dr. Yuyi Lin Yang Shen. Dealing with non-linear optimization problems using   algorithm   with   penalty function.

[3] Subrata Datta, Chayanika Garai and Chandrani Das. The efficient genetic algorithm on programming problems for fittest chromosomes, Volume 3, No. 6, June 2012 Journal of Global Research in Computer Science, ISSN-2229-371X.

[4] Chhavi Mangla, Moin Uddin and Musheer Ahmad. G.A based optimization for system of Non –Linear equation. International Journal of advanced Technology and engineeringexploration,vol(5144),ISSN:2394,5443.http:dx.doi.org/10.191901/, IJATEE,2018.543009.

[5] Dr. Amanpreet Singh, Shivani Sanan , Amit Kumar. Solving Unconstrained Rosenbrock Optimization Problem Using Genetic Algorithms at High Technology Letters. ISSN NO: 1006-6748http://www.gjstx-e.cn/, Volume 27, Issue 3, 2021

[6] Punam S Mhetre. Genetic algorithm for the solution of both linear and Non-linear optimization problems . International Journal of Advanced Engineering Technology, E-ISSN 0976-3945, IJAET/Vol.III/ Issue II/April- June, 2012/114-118 (2012).

[7] T. Yokota, M. Gen. Solving non-linear Integer Programming problem using. Proceeding of IEEE International conference on system, Man and cybernetics 5oct 1994, DOI:10.1109/ICSMC.1994.400076.

[8] Chhavi Mangla, Moin Uddin and Musheer Ahmad. Optimization of complex Non –Linear system using Genetic algorithm. International Journal of information technology, ISSN 2511-2104, DOI 10,1007/541870-020-00421-z ,2020.

[9] Lalit Kumar, Dr. Dheerendra Singh. Solution of NP-hard Problem using ABC algorithm. International journal of engineering and technology. ISSN 0976 – 6367(Print) ISSN 0976 – 6375(Online) Volume 4, Issue 1, January- February (2013), pp. 171-177.

[10] S. Talatahari, H. Mohaggeg, Kh. Najafi and A. Manafzadeh. Solving Parameter Identification of Nonlinear Problems by Artificial Bee Colony Algorithm, Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2014, Article ID 479197, 6 pages http://dx.doi.org/10.1155/2014/479197.

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

[12] Weifeing Gao, Lingling Huang, Yuting Luo, Zhifang Wei, and Sanyang Liu. Constrained Optimization by Artificial Bee Colony Framework. Digital Object Identifier 10.1109/ACCESS.2018.2880814, 2169-3536 2018 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission.

[13] Soudeh Babaeizadeh and Rohanin Ahmad. Constrained Artificial Bee Colony Algorithm for Optimization Problems. AIP Conference Proceedings 1750, 020008 (2016); https://doi.org/10.1063/1.4954521 Published Online: 21 June 2016.

[14] Soudeh Babaeizadeh and Rohanin Ahmad. Enhanced Constrained Artificial Bee Colony Algorithm for Optimization Problems, The International Arab Journal of Information Technology, Vol. 14, No. 2, 2017.

[15] Nadezda Stanarevic, Milan Tuba, and Nebojsa Bacanin. Modified artificial bee colony algorithm for constrained problems optimization. An International journal of mathematical models and methods in applied science, issue 3, Volume 5, 2011.

[16] Jouni Lampinen, Ivan Zelinka. Mixed Variable non- linear optimization by differential evolution. Corpus ID: 11285973

[17] Md. Abul Kalam Azad, Edite MGP, Fernandes Ana MAC Rocha. Non-linear continuous global optimization by modified differential evolution. Department of production and system school of Engineering, University of Minho, 1710-057 Braga, Portugal.10.1063/1.3498653,2010.

[18] Musrrat Ali, Millie Pant, Ajith Abraham. Simplex differential evolution. Department of paper technology, Indian institute of technology Roorkee Saharanpur capus, Saharanpur 247001 India vol 6, No.5, 2009.

[19] Zhang Xiao Fei, Guo Xiang Fulb, Yuan Li Hua. Reactive power optimization of power system based on niching differential evolution algorithm. International symposium on computers and informatics (ISCI),2015.

[20] Chiha ibtissem, Liouqne Hend. In based cost differential evolution algorithm for multi objective optimization problem. Vol 05. Electrica; Engineering department ENIM. Monastir, Tunisia. Asian general of Applied sciences (ISSN 2321-0893),2017.

[21] Jeerayut Wetweerapong and Pikul Puphasak 2020: In an improved DEA with the restart technique to solve system on non-linear equations. An International general of optimization and control theories and applications. ISSN: 2146-0957, vol.10, No.1, PP.118-136 2020.

[22] Saeed nezhadhosein, Aghile Hejdari, Raza Ghanbari. Integrating DEA with modified Hybrid GA for solving non-linear optimal control problems. International journal of mathematical science and informatics. Vol 12, No.1 2017, PP (47-67)

[23] JR Koza, D.Andre, F. H. Bennett and M.Keane. Genetic algorithm Programming (3) of Darwinian Invention and Problem Solving.

[24] David E. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Longman, USA; 1989.

[25] Dr. Amanpree Sings,Shivani Sanan, Amit Kumar. Solving Unconstrained Rosenbrock Optimization Problem Using Genetic Algorithms. ISSN NO : 1006-6748, ISSN NO : 1006-6748, http://www.gjstx-e.cn,2021.

[26] D. Karaboga. An Idea Based On Honey Bee Swarm for Numerical Optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[27] B. Basturk, D. Karaboga. An Artificial Bee Colony (ABC) Algorithm for Numeric Function Optimization, IEEE Swarm Intelligence Symposium 2006, May 12- 14, 2006, Indianapolis, Indiana, USA.

[28] J. Ilonen, J. K. Kamarainen and J. Lampinen. Differential evolution training algorithm for feed-forward neural networks, Neural Processing Letters 17 (2003), no. 1, 93-105.