

Enhanced form of solving real coded numerical optimization problem using ABC algorithm with linear crossover operator

Mrs. Shahana Gajala Qureshi

Department of computer science & Engineering, RITEE
CSVTU, Raipur
Chhattisgarh, India
shahana.rit@gmail.com

Mrs. Uzma Arshi Ansari

Department of computer science & Engineering, RITEE
CSVTU, Raipur
Chhattisgarh, India
arshi.uzma@gmail.com

Abstract— Artificial Bee Colony (ABC) algorithm is population-based swarm intelligence algorithm. There are many algorithms Present for solving numeric optimization problem. ABC is based on the intellectual behaviour of honey bee crowd. In this work, ABC algorithm is used with Genetic crossover machinist and tested on standard benchmark functions, and also result compared with the X-ABC algorithm with the advantage of fewer organize Parameters. Obtained results show that the performance of the ABC with crossover gives better results than the X-ABC algorithm.

Keywords- Swarm intelligence, Artificial bee colony, Genetic algorithm, Evolutionary algorithm, Numerical function optimization.

I. INTRODUCTION

Several optimization algorithms have been proposed to choose the best element from some set of available alternatives. Based on intelligent behaviour of honey bee swarm there is only one numerical optimization algorithm in the literature [1]. Swarm develops collective intelligence. In 2005 D.Karaboga has described an artificial bee colony (ABC) algorithm [2] for optimizing multivariable functions.

Several population based algorithm has been proposed to find near-optimal solutions to the difficult optimization problems like: scheduling and routing problems. A population-based algorithm refers the population consisting of possible solutions to the problem are modified by applying some operators on the solutions depending on the information of their fitness values. Population based algorithm is classified into two groups: evolutionary algorithms [3] and swarm intelligence-based algorithms [4]. Evolutionary algorithm is one of the most popular genetic algorithm (GA) based on the natural evolution. In the basic GA, a selection operation is applied to the solutions evaluated by the evaluation unit.

Linear real coded crossover operator is applied to ABC to check whether the algorithm works better on the Benchmark function or not. In this paper we performed an experiment on enhanced model of ABC with crossover operation. In this work a crossover step is added to the standard ABC. The crossover is performed between each individual food source best positions. After the crossover, the fitness of the individual food source best position is compared with that of the two off-springs, and the best one is taken as the new individual food source best position. Two food sources are selected as parents through selection process and calculate their fitness value. After selecting a crossover point randomly, new fitness values are generated using crossover probability. Both the above techniques perform a crossover by swapping the food source around the crossover point.

The rest of the paper is organized as follows: Genetic algorithm (GA) in second section. ABC with Crossover operator, Algorithm and Flowchart in section third. In section fourth, benchmark functions and parameter setup

are described. Results are presented in section five followed by conclusion in section eight and future scope in Section 7.

II. GENETIC ALGORITHM

Genetic algorithms play a significant role, as search techniques for handling complex spaces, in many fields such as artificial intelligence, engineering, robotic, etc. Genetic algorithms (GAs) are general purpose search algorithms which use principles inspired by natural genetic populations to evolve solutions to problems. Basically it consists of five components: a random number generator, a fitness evaluation unit and genetic operators for reproduction; crossover and mutation operations. The basic algorithm of GA is as below:

- 1: Initialize Population
- 2: repeat
- 3: Evaluation
- 4: Reproduction
- 5: Crossover
- 6: Mutation
- 7: until requirements are met

The initial population required at the start of the algorithm, is a set of food source generated by the random generator. Each position of a food source represents a possible solution of the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. A fitness value is a measure of the goodness of the solution that it represents. Basically the aim of the genetic operators is to transform this set of food source into sets with higher fitness values. The reproduction operator performs a natural selection function known as seeded selection. Individual food source are copied from one set (representing a generation of solutions) to the next according to their fitness values, the higher the fitness value, the greater the probability of a food source being selected for the next iteration. The crossover operator chooses pairs of food sources at random and produces new pairs. The simplest crossover operation is to cut the original food sources nectar amount at a randomly selected point and to exchange. The number of crossover operations is governed by a crossover rate. The mutation operator randomly mutates or reverses the values of food source. The number of mutation operations is determined by a mutation rate. A phase of the algorithm consists of applying the evaluation, reproduction, crossover and mutation operations. A new generation of solutions is produced with each phase of the algorithm [5].

III. CROSSOVER OPERATOR

The crossover operator is a method for sharing information between chromosomes; it combines the features of two parent chromosomes to form two offspring, with the possibility that good chromosomes may generate better ones. The crossover operator is not usually applied to all pairs of chromosomes in the intermediate population. A random choice is made, where the likelihood of crossover being applied depends on probability defined by a crossover rate, the crossover probability. The crossover operator plays a central role in GAs. It combines parts of good solution to form new potential solution. Information contained in one solution combine with information contained in another solution and the resulting solution will either have good fitness or survive to exchange this information again. Crossover operators exist for both real coded and binary coded GA. Since this paper explores the application of crossover operator to ABC for numerical optimization problem therefore for this paper we discuss only real coded crossover operator. A number of real coded crossovers have been introduced for GA and other heuristic technique. Linear crossover we used listed in next section.

IV. PRAPOSED ARTIFICIAL BEE COLONY (ABC) ALGORITHM WITH CROSSOVER OPERATOR

ABC usually suffers from premature convergence, tending to get stuck in local optima, low solution precision and so on. In order to overcome these shortcomings and get better results, various improvements to ABC have been proposed. One of the approach ABC with crossover operator is proposed by adding a crossover step to the standard ABC to get improved result.

In this paper ABC with linear crossover is applied to the 3 benchmark functions, which gives the better optimum result at better probability as compare to X-ABC algorithm. Food sources generated by ABC are randomly

selected for crossover operation and two new offspring's are formed. The best offspring (in terms of fitness value) selected from the new offspring's. This new best offspring replaces the worst fitness value which is selected for crossover. The replacement is done if the new best offspring has the good fitness value than the parent food source.

Algorithm of proposed method is as follows:

- INITIALIZE.
- REPEAT.
 - (a) Place the employed bees on the food sources in the memory;
 - (b) Linear Crossover Operator
 - (c) Place the onlooker bees on the food sources in the memory;
 - (d) Send the scouts to the search area for discovering new food sources.
- UNTIL (requirements are met).

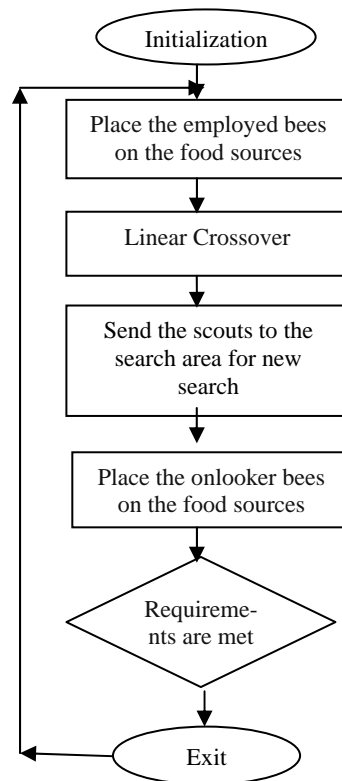


Figure 1. Flow diagram of ABC with Crossover

V. PARAMETER SETUP FOR ABC WITH LINEAR CROSSOVER AND BENCHMARK FUNCTIONS

In this paper 3 standard set of Benchmarks functions [6] are chosen to test and compare the performance of both ABC with linear crossover and X-ABC. All the benchmarks chosen have multivariable functions and have different degree of complexity. This set includes much different kind of problems like: multimodal, local minima in order to approximate, global optimum, and dimensionality.

A. Benchmark Functions

1) Griewank function

The Griewank function extensively used to test the convergence of optimization functions. The Griewank function of order n is defined by:

$$f_1(\bar{x}) = 1/400 \left(\sum_{i=1}^D (x_i^2) \right) - \left(\prod_{i=1}^D \cos(x_i/\sqrt{i}) \right) + 1$$

Its value is 0 at its global minimum (0,0,...,0) and local minima at $\pm x$. Initialization range for the function is [-600,600]. It has a global minimum of 0 at the point $x=0$.

2) *Rosenbrock function*

Rosenbrock function is often used as a test problem for optimization algorithms.

$$f_2(\bar{x}) = \sum_{i=1}^D 100(x_i^2 - x_{i+1})^2 + (1-x_i)^2$$

It has a global minimum at $(x, y) = (1, 1)$ where $(x, y) = 0$. Initialization range for the function is [-15, 15].

3) *Sphere function*

It is the first function of De Jong's test set. Function is continuous, convex and unimodal. It has the following general definition

$$f_3(x) = \sum_{i=1}^n x_i^2$$

Initialization range for the function is $-5.12 \leq x_i \leq 5.12, i = 1, \dots, n$. Global minimum $f(x) = 0$ is for $x_i = 0, i = 1, \dots, n$.

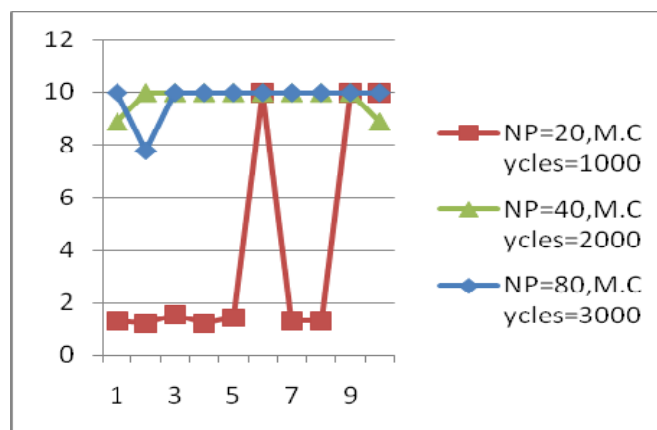
VI. PARAMETER SETUP FOR EXPERIMENT

In this paper the parameter setting are as colony size= 20, 40, 80, Maximum number of cycles (MCN) = 100, 2000, 3000 respectively for the dimension 10. Food sources equal the half of the colony size. The percentage of onlooker bees was 50% of the colony, the employed bees were 50% of the colony and the number of scout bees was selected as one. X_{max} and X_{min} are the upper and lower bounds of the decision variables. X_{min} (lb) and X_{max} (ub) are taken in the range -100 to 100. This paper presents an experiment on X-ABC and ABC with Crossover operator. For Crossover, linear crossover which is explained previously is chosen. For experiments, linear crossover's probability is taken in the range of 0.1 to 0.9.

Table I X-ABC vs. ABC with crossover operator

Benchmark Function	Fitness value of X-ABC	Fitness value of ABC with linear crossover operator								
		pr < .1	pr < .2	pr < .3	pr < .4	pr < .5	pr < .6	pr < .7	pr < .8	pr < .9
Griewank	1.332268	1.221245	1.554312	1.221245	1.44329	9.991007	1.332268	1.332268	9.992007	9.992007
Griewank	8.881784	9.992007	9.992007	9.992007	9.992007	9.992007	9.992007	9.992007	9.992007	8.881784
Griewank	9.992007	7.771561	9.992007	9.992007	9.992007	9.992007	9.992007	9.992007	9.992007	9.992007

X-ABC vs. ABC with crossover operator indicate the results that obtained after 1000 cycles with a population 20 for the dimension 10

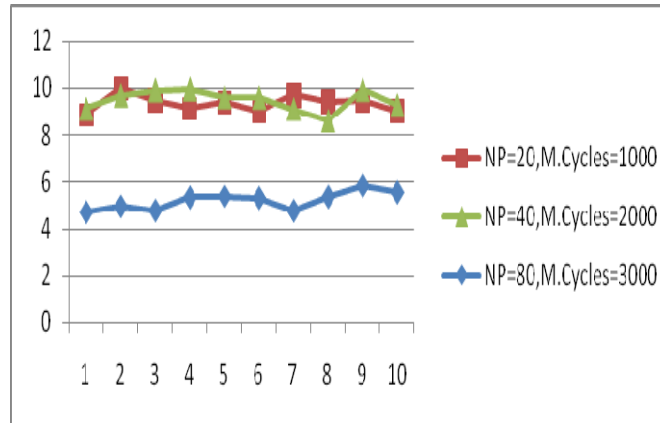


Graph 1 Fitness value of X-ABC vs. ABC with linear crossover operator

Table II X-ABC vs. ABC with crossover operator

Benchmark Function	Fitness value of X-ABC	Fitness value of ABC with linear crossover operator								
		pr < .1	pr < .2	pr < .3	pr < .4	pr < .5	pr < .6	pr < .7	pr < .8	pr < .9
Rosenbrock	8.881784	9.992476	9.502416	9.123774	9.409717	9.004217	9.715304	9.428769	9.480702	9.0259
Rosenbrock	9.109312	9.676996	9.899817	9.944221	9.620697	9.580132	9.105082	8.607993	9.918184	9.264489
Rosenbrock	4.737206	4.976909	4.80848	5.359311	5.41167	5.337368	4.799651	5.387644	5.846728	5.574773

X-ABC vs. ABC with crossover operator indicate the results that obtained after 2000 cycles with a population 40 for the dimension 10

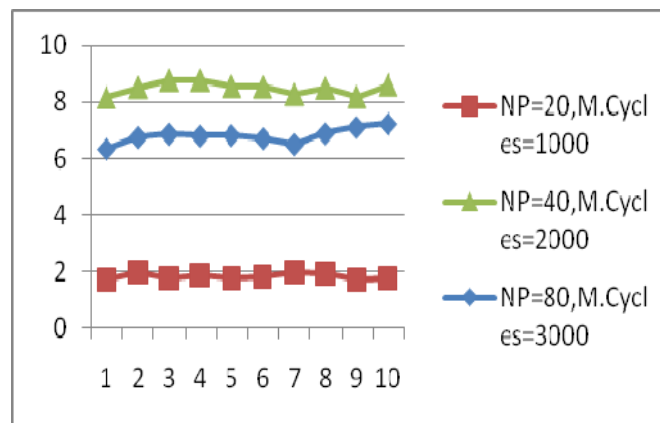


Graph 2 Fitness value of X-ABC vs. ABC with linear crossover operator

Table 3 X-ABC vs. ABC with crossover operator

Benchmark Function	Fitness value of X-ABC	Fitness value of ABC with linear crossover operator								
		pr < .1	pr < .2	pr < .3	pr < .4	pr < .5	pr < .6	pr < .7	pr < .8	pr < .9
Sphere	1.724582	1.975405	1.762315	1.868645	1.769938	1.831868	1.982908	1.909434	1.727958	1.761739
Sphere	8.16916	8.519625	8.777245	8.774533	8.561635	8.539031	8.274195	8.498855	8.191727	8.614061
Sphere	6.350367	6.769752	6.887722	6.818324	6.841564	6.728867	6.515446	6.90444	7.125173	7.24512

X-ABC vs. ABC with crossover operator indicate the results that obtained after 3000 cycles with a population 80 for the dimension 10



Graph 3 Fitness value of X-ABC vs. ABC with linear crossover operator

VII. EXPERIMENTAL RESULT

In Experiments we compared the X-ABC and ABC with crossover operator algorithms on a set of 3 benchmark functions given in Table 1, 2, 3. The X-ABC and ABC with linear crossover is implemented in C++. Recorded

simulated results are presented in Tables 1, 2, 3 for each benchmark function. In this experiment mean fitness value is calculated for both the algorithms with each 4 benchmark function. In the tables the bold readings are improved result over the ABC with linear crossover. Obtained results indicated in tables that ABC with crossover provides better results than original ABC algorithm. Here we also observed that Sphere function provides the better improved result than the Griewank, Rosenbrock, and Rastrigin.

VIII. CONCLUSION

In this paper, ABC is compared with enhanced ABC algorithm. Enhanced ABC is obtained by an additional step added to ABC that is a crossover technique of GA. Code of ABC and Enhanced ABC is developed in C++ and results are shown in Table 1, 2, 3. The experiments are performed on a set of 3 benchmark problems available in the literature. The crossover probability range is set to 0.1 to 0.9. For each benchmark function results are tabulated for crossover probability 0.1 to 0.9 in terms of mean fitness which describes the efficiency and accuracy are depicted in bold. From the result obtained it is concluded that Enhanced ABC has been found to have successful performance on Griewank, Rosenbrock, and Sphere benchmark problems.

IX. FUTURE SCOPE

In this experiment we compared X-ABC and ABC with linear crossover operator algorithms on a set of 3 benchmark functions and concluded that enhanced algorithm gives the improved result than X-ABC algorithm. In the future work X-ABC is compared with enhanced ABC with naive crossover, simulated crossover and laplus crossover operators of GA on a set of 4 benchmark functions Griewank, Rosenbrock, Rastrigin and Sphere .

REFERENCES

- [1] Yang, X.S. ,“Engineering optimizations via nature- inspired virtual bee algorithms” ,Lecture Notes in Computer Science, pp. 317–323. Springer, GmbH (2005)
- [2] Karaboga, D. ,“An idea based on honey bee swarm for numerical optimization” ,Technical Report- TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005
- [3] A.E. Eiben, J.E. Smith, “Introduction to Evolutionary Computing’ , Springer, 2003.
- [4] R.C. Eberhart, Y. Shi, J. Kennedy, “Swarm Intelligence”, Morgan Kaufmann, 2001.
- [5] D.T. Pham, D. Karaboga, “Optimum design of fuzzy logic controllers using genetic algorithms” , Journal of Systems Engineering 1 (1991) 114–118
- [6] Srinivasan, D., Seow, T.H., “ Evolutionary computation” , CEC ’03, 8–12 Dec. 2003, 4(2003), Canberra, Australia, pp. 2292–2297.
- [7] D. Karaboga, B. Akay, “Artificial bee colony (ABC), Harmony search and bees algorithms on numerical optimization”, Erciyes University, the Dept. of Computer Engineering, 38039, Melikgazi, Kayseri, Turkiye.
- [8] Dervis KARABOGA, “An idea based on honey bee swarm for numerical optimization”, Erciyes University, Engineering Faculty Computer Engineering Department, Kayseri/Türkiye, TECHNICAL REPORT-TR06, OCTOBER, 2005.
- [9] Adil Baykaso lu, Lale Özbakır and Pınar Tapkan, “Artificial bee colony algorithm and its application to generalized assignment problem”, University of Gaziantep, Department of Industrial Engineering 2Erciyes University, Department of Industrial Engineering Turkey.
- [10] J.G. Digalakis, K.G. Margaritis, “An experimental study of benchmarking functions for genetic algorithms” , International Journal of Computer Mathematics 79 (4) (2002) 403–416.
- [11] Teodorovi´c, D., “Transport modeling by multi-agent systems: a swarm intelligence approach” , Transport. Plan. Technol. **26**(4), 289–312 (2003)
- [12] Teodorovi´c, D., Dell’Orco, M., “ Bee colony optimization—a cooperative learning approach to complex transportation problems” , In: Proceedings of the 10th EWGT Meeting, Poznan, 13–16 September 2005
- [13] Tereshko, V., “Reaction-diffusion model of a honeybee colony’s foraging behaviour” , In: Schoenauer M. (ed.) Parallel Problem Solving from Nature VI. Lecture Notes in Computer Science, vol. 1917, pp. 807–816, Springer, Berlin (2000)
- [14] Holland, J.H., “Adaptation in natural and artificial systems”, University of Michigan Press, Ann Arbor, MI (1975)
- [15] Lucic, P., Teodorovi´c, D., “Transportation modeling: An Artificial Life Approach” , ICTAI, pp.216–223.Washington D.C. (2002)

AUTHORS PROFILE



Mrs Shahana Gajala Qureshi , Asst.professor in Computer Science & Engineering department ,Raipur Institute of Technology(RIT) afiliated by CSVTU,Bhilai,Chhattisgarh,India, Pursuing M.Tech(4th sem.)in Computer Science & Engineering from RIT collrge,Raipur(C.G.),India. I have 4 yers of experiance in CSE department.My area of interest is Softcomputing, AI,Neural Network.

Mrs.Uzma Arshi Ansari,Reader(HOD) in Computer Science & Engineering department ,Raipur Institute of Technology(RIT) afiliated by CSVTU,Bhilai,Chhattisgarh,India. M.E.in Computer Technology & Application from Shri Shankaracharya College of Engineering & Technology,Bhilai,Afiliated by CSVTU,Bhilai,Chhattisgarh,India.