Solving Real Optimization Problem using Genetic Algorithm with Employed Bee (GAEB)

Deepak Singh
National Institute Of Technology
Raipur,India

Ankit Sirmorya
National Institute Of Technology
Raipur,India

ABSTRACT

Optimization is a research branch that models the maximization and minimization of a real function by systematically choosing the input values from within an allowed domain. Its concerned with finding the best solutions for a given problem irrespective of its discipline. The Artificial Bee Colony (ABC) algorithm is a swarm based meta-heuristic optimization algorithm inspired by the intelligent behavior of honey bee swarm. Genetic Algorithms (GA) is an attractive class of computational models that mimic the biological evolution process for solving problems in a wide domain. In this work, Genetic Algorithm with Employed Bee (GAEB) is proposed where, hybridization of the employed bee operator of ABC algorithm has been done in GA. The results obtained showed that GAEB provides better optimized results than classical GA.

General Terms

Real Optimization, Algorithm.

Keywords

Artificial Bee Colony; ABC; Genetic Algorithm; GA; Genetic Algorithm with Employed Bee; GAEB.

1. INTRODUCTION

Man has been involved with the process of optimization since long. Earlier, optimization comprised of prejudices and unscientific rituals like sacrificing animals and consulting oracles. When the circumstances were appropriate, the timing was considered to be auspicious (optimum) for plantation of crops or embarking on a war. With the passage of ages optimization faced a revolutionary change. With the advent of digital computers interest in optimization process has taken a giant leap. In recent years, the advancement of optimization techniques has reached new limits. Consequently, many complex optimization problem has been solved which seemed intractable few years back.

Optimization is the process of obtaining the ‘best’. The term ‘optimum’ is a technical term for quantitative measure. Therefore, the crux of optimization is finding the maximum or the minimum value. On the other hand, optimization practice is the collection of techniques, methods, procedures and algorithms that can be used to find the optima. Typical areas of its application are modeling, process control, quality control, production scheduling, inventory control, budgeting, etc. Numerous recent innovations like neural networks and adaptive system rely upon optimization theory.

Many optimization algorithms tend to be so elaborate that the time and space complexities related to their algorithmic solution seem unacceptable. However, in reality if the approximate or partial solution is admitted the set of techniques to cope with such problems gets extended as in case of Particle Swarm Optimization [1] technique. Algorithms that nearly give right answer or provide a solution not for all the instances of the problem are called *heuristic algorithm* [6].In this case the objective function is a function used to evaluate a quality of the generated solution. These algorithms aim at finding the optimal of all possible solutions that is one that minimizes and maximizes an objective function. For a given optimization problem the collection of possible solutions form a search space. Branch-and-Bound technique and dynamic programming algorithms are quite effective heuristic technique but are unacceptable for NP-complete problems on account of too high time complexities. The *hill-climbing* [7] technique which involves the local search within the search space is used as heuristics for intra group replications. There are optimization problems that belong to the classes NP-complete [8] or NP-hard, having no polynomial algorithms to solve such problems, therefore, for huge inputs heuristics is developed. Besides this, heuristic approach limits the design alternative being evaluated for their approach is scenario-by-scenario method (Caldas et al,2003).

In contrast to the heuristics search approach for optimization, the Meta heuristics [2] approach formulates a computational method which uses an iterative process to improve a candidate solution based on the given measure of quality for optimizing a problem. The search space involved is also quite larger for meta-heuristic approach compared to the heuristic approach. This approach aims at finding good heuristics. The implementation of this approach involves tuning of an optimization problem to suite particular combination of search problem, performance measure and algorithmic toolkit. The meta-heuristic approach forms an essential component in building and implementing population based algorithms: evolutionary algorithms (EA) and swarm intelligence based algorithms. Nils Aall Barricelli[3] introduced Genetic Programming with evolutionary algorithms in 1954. In 1992 Marco Dorigo[4] proposed Ant Colony Optimization algorithm as one of the swarm intelligence method.

This paper is organized as: Section 2 and 3 briefs about the classical GA and ABC algorithm, respectively. The proposed algorithm (GAEB), of hybridizing the GA with the operators of ABC algorithm is put forth in Section 4 along with the benchmark functions and experimental set up in Section 5 and 6 contiguously. The tabulated simulation and the experimental results are placed in Section 7.

2. GENETIC ALGORITHM

In 1975, Professor John Holland [9] suggested an attractive class of computational models, called Genetic Algorithms (GA) , that replicates the human genetic process to solve problems. The mechanism under GA have been analyzed and explained later by Goldberg [10], De Jong[11], Davis[12] and many others. Over the last 30 years, GA has been used in three major applications, namely, intelligent search, optimization and machine learning. Recently, it has developed usage in more complex problems like neural networks and fuzzy logic. On account of its application in manifold problems it is coined by a generic name: “soft computing”.

The crux of Holland’s idea was based on the concept of natural selection of the biological evolution process. The process starts with generating a random population of possible solutions. This population creates a generation population which indeed creates another generation and the process continues. The best elements of the previous generation are used to create the next generation hoping for a better population. A new generation of possible solutions for a given problem in each GA cycle. At the starting phase, the selection process is performed from the initial population representing possible solutions based on the fitness of each population element. Once the selection process is done, subsequent genetic manipulation process is carried out. The genetic manipulation process is carried out in two steps. The first step comprises of crossover operation which recombines the dimensions of the two selected population elements. The underlying idea behind crossover is that the new chromosome may be better than both of parents in case it takes the best characteristics from each of parents. The crossover operator summarizes good solution parts creating a new potential solution. Information contained in one solution is combined with the information in another solution formulating a resulting solution having better fitness. In the GAEB, linear crossover [15] methodology is opted which is one of the earliest operator in real coded crossover. It generates three solutions from two parents and the best two offspring replace parents. Let ($y_1^{(1,s)}$,$y_2^{(1,s)}$,…,$y_n^{(1,s)}$) and ($y_1^{(2,s)}$,$y_2^{(2,s)}$,…,$y_n^{(2,s)}$) are two parent solutions of dimension ‘n’ at generation ‘s’. Linear crossover generates three offspring from these two parents as shown in equation (2.1),(2.2) and (2.3) and the best two offspring being chosen as offspring.

$$0.5(y_i^{(1,s)} + y_i^{(2,s)}) \qquad (2.1)$$

$$1.5y_i^{(1,s)} - 0.5y_i^{(2,s)} \qquad (2.2)$$

$$-0.5y_i^{(1,s)} + 0.5y_i^{(2,s)} \qquad (2.3)$$

i=1,2, …,n

The second step of genetic mutation process is termed as mutation, where the bits at one or more randomly selected population element are altered for optimizing the solution. In this paper uniform mutation is used where the value of the chosen dimension is replaced with a uniform random value within the specified range, consequently following Gaussian distribution [14]. Mutation is an essential feature of genetic search as it ensures that the search do not stagnates in the local optima. The offspring produced by the mutation process act as the initialization population for the next cycle of Genetic Algorithm,

3. ARTIFICIAL BEE COLONY ALGORITHM

In 2005 Dervis Karaboga[5] proposed a global optimization algorithm, Artificial Bee Colony algorithm for numerical optimization. The algorithm put forth a model in which the system global behavior is determined from the individual’s local behavior where the interactions and coordination lead to an organized team work. The model defines three leading modes of behavior for self-organizing and collective intelligence: recruitment of employed bees to search rich food sources, selection of the food sources after sharing information with the employed bees and determining the scout bees for exploration of the possible food sources. First, the scout bees explores the food sources then returns back and announces the information about the food sources in the colony ,i.e. the nectar amount and the direction. Secondly, after the information is shared the employed bees visits the food source area visited by her in the previous cycle which already resides in her memory and chooses a new food source based on the neighborhood’s visual information. If the nectar amount of the new food source is higher than the previous source, the employed bee memorizes the position of the new source position and forgets the old one. Thirdly, the employed bees share the information about the nectar amount in the hive. The probability with which the onlooker chooses a food source increases proportionately with the nectar amount. The employed bees whose food sources gets abandoned and exhausted becomes scout bees in search of new food sources. Each employed bee is associated with only one food source; hence the number of employed bees is equal to the number of food sources.

In ABC, a population based algorithm, the food sources indicates the number of possible solutions for the optimization problem which equals the number of the artificial bees. Besides this, the nectar amount implies the quality (or fitness) of the associated solution. In the initialization, the artificial bees discover the set (or population) of initial solution vectors. The artificial scout bees initializes vectors of all the population sources, $\vec{y}_m$’s (m=1, 2, 3…SZ, SZ is the population size). Each solution vector $\vec{y}_m$, comprises of n variables, (y_{mj} where j=1…n) which are to be optimized to minimize the objective function. The initialization might follow the expression (3.1):

$$y_{mj} = l_j + random(0,1) \times (u_j - l_j) \qquad (3.1)$$

where u_j and l_j are upper and lower limits of y_{mj}, respectively. At the second stage, the strategy aims at moving towards the better solution by means of a neighbor search algorithm abandoning the poor solutions. The artificial employed bees select the new food source, $\vec{v}_m$ using the expression 3.2 as:

$$v_{mj} = v_{mj} + \varnothing(v_{mj} - v_{kj}) \quad (3.2)$$

where $\varnothing_{mj}$ is a random number generated between [-a,a] and y_k is a randomly selected food source. At the third phase, the onlooker bees chooses the food source vector $\vec{y}_m$ depending on the fitness values using the expression 3.3 as:

$$p_m = \frac{fit_m(\vec{y}_m)}{\sum_{m=1}^{SZ} fit_m(\vec{y}_m)} \quad (3.3)$$

where p_m is the probability with which $\vec{y}_m$ solution vector of optimization problem in selected. After $\vec{y}_m$ is probabilistically chosen, the neighborhood vector $\vec{v}_m$ is determined using 3.2.

4. GENETIC ALGORITHM WITH EMPLOYED BEE

Despite of the fact that Genetic Algorithms can be used to solve any multi-dimensional optimization problem which can be described with the chromosome encoding, it has no absolute assurance to provide a global optimum. Often the result obtained from GA gets limited to local optimal solution when the population size is large. This work aims at overcoming this shortcoming of GA. In this work, the classical Genetic Algorithm is hybridized by including operators from the Employed Bee Phase of the Artificial Bee Colony algorithm with an objective to find the global optimum. The number of possible solutions is termed as population. Every population element is called chromosome. The main steps involved in the algorithm are given below:

- Initialization of values assigned randomly.
- Selection of solution chromosomes randomly.
- Crossover of selected solution chromosomes linearly.
- Mutation of the resultant population element, chromosomes.
- Hybridization using employed bee phase.

Each cycle of the GAEB produces new generation of possible solutions for the optimization problem. At the initialization step, each of the N_p population elements, chromosomes are initialized. Every individual chromosome has D dimensions, each dimension implying an individual generation represented by Y_{ij} where i=1…N_p and j=1…D. After the first step, the chromosomes are selected for the subsequent genetic manipulation process. The selection process is completely random in which chromosomes. After the selection of chromosomes is over, the genetic manipulation process comprising of two steps is carried out. Initially, linear crossover which recombines each of the dimensions of two selected chromosomes are executed. Consequently, two child chromosomes are produced increasing the population elements to N_p+2. Based on the fitness of individual chromosome, Np population elements are selected. The second step of genetic manipulation process is mutation, where one of the randomly selected dimension (Y_{ij}) is altered which helps to overcome trapping at local optimum to some extent. Once the mutation process gets over, hybridization with the employed bee phase is carried out satisfying eligibility criteria. In the hybridization process, each of the dimensions of every population element gets updated. The dimension (Y_{ij}) gets updated following the expression mentioned in 3.2. Here the fitness value of each dimension is analogous to the nectar amount of the food source for the bee hive. Similar to the ABC algorithm where the artificial employed bee modify the solution vector the fitness value of each dimension of GA gets updated in order to achieve better optimal solution using the neighbor search algorithm. In this case, the neighbor is the dimension having the same index in a randomly selected population element. A particular dimension gets modified only if the outcome of the comparison of its dimension value with another dimension of the same index after being operated by the random function updates the dimension value. In addition to this, the time overhead is minimized by defining probability criteria for the application of employed bee phase after the mutation process. X_{max} and X_{min} are upper and lower bounds of the dimension values. If the updated dimension value exceeds X_{max} or lowers X_{min}, the dimension value is set to X_{max} or X_{min} respectively.

```
do
    For   i=1   to   $N_p$
        For j=1   to D
            Initialize the values randomly
            Calculate fitness of each
            population
            element
    Select the chromosome randomly for crossover
        Applying linear crossover
    Mutation process
    If Probability criteria met
        Employed bee phase.
Until stopping criteria met
```

5. BENCHMARK FUNCTIONS

A multimodal function has two or more local optima. A function of variables is separable if it can be rewritten as a sum of functions of just one variable [13]. The search process for a multimodal function is difficult if the local optima are randomly distributed. The optimization problem focuses on removing the constraint on local optima towards global optimum. There are standard set of benchmark functions mentioned in the literature which are used for comparative study of various optimization process. In this work, four important benchmark functions has been chosen to test and compare the performance of classical GA and Hybrid GA. All the chosen benchmark function has discrete variables and different degree of complexity. The benchmark functions are listed below:

5.1 Sphere Function:

This function also known as De Jong's function,f_1 is a unimodal function. This is a separable function which is simple and strongly convex. The function has been widely used in the development of evolutionary algorithms as in case of Genetic Algorithm. The function fails in case of algorithms which use the gradient as those algorithms converge very slowly, since the gradient is not oriented along their axis due to separable variables. The initialization range of the function is [-5.12, 5.12]. The function has the value 0 at its global minima (0,0…0)(5.1.1)

$$f_1(\vec{x}) = \sum_{i=1}^{D} x_i^2 \qquad (5.1.1)$$

5.2 Rosenbrock Function:

This function is widely used to solve optimization problems. The global optimum is inside a long, narrow, parabolic flat shaped valley. The gradient usually does not point towards the optimum. The variables are highly interdependent as its difficult to converge the global optimum. The function has value equal to 0 at its global minimum (1,1,…1)(5.2.1). The initialization range of the function is [-15,15].

$$f_3\left(\vec{x}\right) = \sum_{i=1}^{D} 100(x_i^2 - x_{i+1})^{2} + (1-x_i)^2 \qquad (5.2.1)$$

5.3 Griewank Function:

The Griewank function is a test function having several local optima. It focuses on developing interdependency among variables. In this way, it overcomes the failure of optimization technique for each variable independently. The function has value equal to 0 at global minimum (0,0,…,0)(5.1.1) . With increase in dimensionality the number of local optima increases, making this function highly multimodal. However, the optima of Griewank function are regularly distributed in the search space. The search space is largely dominated by gradual slope. At a considerably higher dimensionality, the multimodality faints away and the problem becomes unimodal. The initialization range of this function is [-600,600].

$$f_1(\vec{x}) = \frac{1}{4000}\left(\sum_{i=1}^{D}\left(x_i\right)^2\right) - \left(\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)\right) + 1 \qquad (5.3.1)$$

5.4 Rastrigin Function:

This function is a multimodal function as it is based on Sphere function with the addition of cosine modulation to produce numerous local minima. The limitation lies in getting trapped in a local optima on its way of finding a global optimum value. However, to overcome this shortcoming the local optima are regularly distributed in the search space. By doing so, the performance of the search process is increased.

$$f_3(\vec{x}) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10) \qquad (5.4.1)$$

6. EXPERIMENTAL SETUP

In GAEB algorithm, the colony population (employed bees and onlooker bees) is taken 60. The number of food sources is taken half of the colony size. The limited trials through which the food source could be improved is considered 100 failing which the food source would be abandoned. The number of cycles for foraging (stopping criteria). The number of parameters of the problem to be optimized (dimension of each population element) is 30. The random number generated for updating the dimension values varies from -100 to 100. The algorithm is run 30 times in order to check its robustness.

TABLE1: COMPARED RESULTS FOR GA AND GAEB

Benchmark Function	G.A (mean)	GENETIC ALGORITHM WITH EMPLOYED BEE PHASE								
		p>0.1	p>0.2	p>0.3	p>0.4	p>0.5	p>0.6	p>0.7	p>0.8	p>0.9
Griewank	1.65E+01	**5.37E-16**	**5.96E-16**	**1.17E-15**	**6.33E-14**	**2.62E-11**	**8.61E-09**	**2.87E-06**	**8.57E-04**	**1.91E-01**
Rosenbrock	2.75E+10	**6.96E+01**	**7.22E+01**	**7.58E+01**	**8.15E+01**	**1.10E+02**	**1.99E+02**	**2.06E+02**	**2.15E+04**	**2.37E+07**
Rastrigin	6.43E+04	**1.00E-01**	**3.66E-01**	**1.07E+00**	**2.34E+00**	**4.67E+00**	**9.59E+00**	**2.06E+01**	**7.32E+01**	**9.81E+02**
Sphere	6.69E+04	**6.77E-16**	**2.54E-15**	**7.39E-13**	**2.86E-10**	**1.48E-07**	**3.72E-05**	**7.27E-03**	**2.61E+00**	**7.78E+02**

7. EXPERIMENTAL RESULTS

This section focuses on the efficiency of the classical Genetic Algorithm and the Genetic Algorithm with Employed bee phase of the Artificial Bee Colony with linear crossover operators tested on 4 benchmark functions with 30 dimensions as mentioned in TABLE-1. In order to avoid the choice of a particular initial population and to conduct a fair comparison, 30 runs are considered initiating from the randomly selected points in the search space. The simulated results are recorded in TABLE-1 for each benchmark functions. The mean fitness objective function value for classical Genetic Algorithm and the optimized fitness value for GAEB of 30 runs were calculated and compared. The bold reading in the table is the improved result over the classical Genetic Algorithm. Obtained results indicated in the table proves that Genetic Algorithm with Employed Bee Phase of ABC provides better results than normal Genetic Algorithm on all four benchmark functions over any probability criteria.

8. CONCLUSION

In this work, Hybrid version of GA (GAEB) is compared with GA. The GA is hybridized by implementing the Employed Phase of ABC algorithm over the dimensions of each population element obtained from the mutation process. The results shown in TABLE-1 are obtained from the experiments performed on a set of 4 Benchmark functions available in the literature. The range of probability criteria for implementing the Employed Bee phase of ABC algorithm varies from 0.1 to 0.9 to reduce the time overhead. The mean fitness value of the classical Genetic Algorithm and optimized fitness value of Hybrid Genetic Algorithm over the probability criteria of 0.1 to 0.9 are tabulated for each Benchmark function. The better fitness evaluation values of the GAEB algorithm which describes the efficiency are depicted in bold. From the simulations, it can be concluded that the GAEB algorithm has the ability to get the classical Genetic Algorithm out of local optimum providing better optimized results as shown by the Benchmark functions.

9. REFERENCES

[1] Kennedy, J.; Eberhart, R. (1995), *Particle Swarm Optimization*. Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948

[2] Luke, S. (2009), *Essentials of meta heuristics*.

[3] Barricelli, Nils Aall (1954), *Esempi numerici di processi di evoluzione*, Methodos, pp. 45–68

[4] M. Dorigo, Optimization, *Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano, Italie, 1992

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

[6] Newell, A. & Simon, H. A. (1976), *Computer science as empirical inquiry: symbols and search*. Comm. Of the ACM. 19, 113-126.

[7] Alexandre Temporel & Tim Kovac, *A heuristic hill climbing algorithm for Mastermind*.

[8] Garey, M. R.; Johnson, D. S. (1979), Victor Klee. ed. *Computers and Intractability: A Guide to the Theory of NP-Completeness.*

[9] Holland, John H (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

[10] Goldberg, David E (1989), *Genetic Algorithms in Search, Optimization and Machine Learning,* Kluwer Academic Publishers, Boston, MA.

[11] De Jong, K. A., *An analysis of behavior of a class of genetic adaptive systems*, Doctoral dissertation, University of Michigan, 1975.

[12] Davis, T. E. and Principa, J. C., *A Markov chain framework for the simple genetic algorithm*, Evolutionary Computation, vol. 1, no. 3, pp. 269-288, 1993.

[13] Hadley G., Nonlinear *and Dynamics Programming*. Addison Wesley, Reading, MA (1964).

[14] L. Davis(1989). *Adapting operator probabilities in genetic algorithm.*

[15] Wright, A. Genetic Algorithms for Real Parameter Optimization, Foundations of Genetic Algorithms, G. Rswlins(Ed.), morgen Kaufmann publishers, CA, 1991, pp. 205-218 .