Hybridisation of Artificial Bee Colony Algorithm on Four Classes of Real-valued Optimisation Functions

Mudita Sharma
Department of Computer Science, University of York
United Kingdom
ms1938@york.ac.uk

Dimitar Kazakov
Department of Computer Science, University of York
United Kingdom
dimitar.kazakov@york.ac.uk

ABSTRACT

Hybridisation of algorithms in evolutionary computation (EC) has been used by researchers to overcome drawbacks of population-based algorithms. The introduced algorithm called mutated Artificial Bee Colony algorithm, is a novel variant of standard Artificial Bee Colony algorithm (ABC) which successfully moves out of local optima. First, new parameters are found and tuned in ABC algorithm. Second, the mutation operator is employed which is responsible for bringing diversity into solution. Third, to avoid tuning 'limit' parameter and prevent abandoning good solutions, it is replaced by average fitness comparison of worst employed bee. Thus, proposed algorithm gives the global solution thus improving the exploration capability of ABC. The proposed algorithm is tested on four classes of problems. The results are compared with six other population-based algorithms, namely Genetic Algorithm (GA), Particle Swarm Optimsation (PSO), Differential Evolution (DE), standard Artificial Bee Colony algorithm (ABC) and its two variants- quick Artificial Bee Colony algorithm (qABC) and adaptive Artificial Bee Colony algorithm (aABC). Overall results show that mutated ABC is at par with aABC and better than above-mentioned algorithms. The novel algorithm is best suited to 3 of the 4 classes of functions under consideration. Functions belonging to UN class have shown near optimal solution.

KEYWORDS

Swarm Intelligence, Numerical Optimisation, Artificial Bee Colony algorithm, Mutation

1 INTRODUCTION

Optimisation is everybody's part of life; for instance, optimising a number of tasks by giving priority to one over another such that neither of these degrades overall value of the problem. The industry is also dependent on it to a large extent. Engineering, software testing [12] [14] are few of them. Researchers in the field of evolutionary algorithms and swarm intelligence are working actively to find solutions to optimise problems and to overcome the challenges faced by optimisation algorithms when finding a solution(s) for numerical problems. Particular randomised algorithm has proved to be useful for a wide range of problems in different application areas [11] [6] [18]. Thus it would not be wrong to say that a particular algorithm is best suited for the non-exclusive class of problems. Algorithms such as Genetic Algorithms (GA) [4] [15], Particle Swarm optimisation (PSO) [13], Ant colony optimisation (ACO) [2] are among the efficient algorithms consistently used by researchers, as a starting point for the design of new algorithm, either to improve algorithm's performance by introducing their variants or hybridising with other algorithms sharing similar properties [19]; and modifying them to implement in a specific application of an area. Another research in this field is based on tuning and controlling parameters [3]. Parameters for an algorithm define its performance. Tuning parameter techniques are used to set parameter values before running the algorithm whereas parameters are set during the run of an algorithm in parameter control. Parameter control can further be done in three ways: Deterministic control [5], feedback control where feedback is taken from past generation to come up with a better parameter value for a problem and self-adaptive control [17] [1] where a parameter adapts itself according to the performance of the whole population.

The Artificial Bee Colony algorithm (ABC) [7] has shown better or similar results compared to the above-mentioned algorithms [8]. It was proposed by Dervis Karaboga in 2005 to imitate the foraging behaviour of bees. It uses intelligent behaviour i.e. decentralisation and self-organization of a swarm of bees and has successfully solved a range of mathematical problems [16]. The ABC algorithm is good at exploitation but poor at exploration. To improve its performance many modifications have been done in the way bee searches its neighbourhood [9]. qABC (quick ABC) modifies onlooker bee phase where each onlooker bee chooses a neighbour whose Euclidian distance from its position is less than the mean euclidean distance. Euclidean distance is used as a similarity measure. One common problem faced by the traditional algorithms is premature convergence. This paper introduces a novel variant of Artificial Bee Colony algorithm employing Genetic Algorithm operator-mutation. In this paper, mutation is used for interpolation and extrapolation for solving real-valued problems.

Further sections are organised as follows: Section 2 describes the Artificial Bee Colony algorithm in detail. Section 3 discusses the proposed novel algorithm. Section 4 gives experimental results. Finally, Section 5 provides conclusions and potential lines of future research.

2 ARTIFICIAL BEE COLONY ALGORITHM

Bees' honey foraging behaviour is interesting and intelligent enough to help optimise the range of problems, which mainly imitates the

GECCO '17 Companion, Berlin, Germany
© 2017 ACM. 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3082498

behaviour of employed bee, onlooker bee and scout bee. Employed bees explore food and return back with honey. They then dance on dance area to share the location and amount of honey present on that location with other bees in the hive. Types of dance are waggle dance and round dance, telling how far or near the food is! Onlooker bees observe the dance and reach the neighbourhood of selected employed bee for exploiting a new food source and collect honey from the destination. They repeat this process switching between each other's role.

Artificial Bee colony algorithm has an advantage compared to other nature inspired algorithm as it uses less control parameters compared to others. Thus there are fewer parametes to tune. Parameter set for Artificial Bee Colony algorithm is shown below:

$(SN, D, MCN, limit)$

SN- Number of employed Bees or Number of onlooker bees,

D- Dimension in search space,

MCN- Maximum Cycle Number,

$limit$- Threshold cycle number to abandon the not-changing-position of a bee.

Following are the steps of Artificial Bee Colony algorithm:

Initialization of bees:

All bees in colony are randomly assigned a position between lower and upper bounds for each dimension of a function using following formulae:

$$x_{ij} = l_j + rand(0,1) \cdot (u_j - l_j) \tag{1}$$

x_{ij}- i^{th} employed bee with j^{th} dimension

l_j- Lower limit for j^{th} dimension of a function

u_j- Upper limit for j^{th} dimension of a function

rand(0,1)- Random number between 0 and 1

REPEAT

Employed bee phase:

Each employed bee is assigned a random position in the search space using formulae:

For any randomly chosen j-dimension of i^{th} bee,

$$v_{ij} = x_{ij} + rand[-1,1] \cdot (x_{ij} - x_{kj}), k \in [1, SN], i \neq k \tag{2}$$

v_{ij}- New candidate solution for i^{th} employed bee with j^{th} dimension

$rand[-1,1]$- Random number between -1 and 1

x_{kj}- Randomly selected k^{th} employed bee with j^{th} dimension

Greedy selection-For each i^{th} bee, x_{ij}, past best fitness and fitness of new position, v_{ij}, are compared to attain best position in current cycle by selecting the one with best out of these two positions.

Onlooker bee phase:

Onlooker bees are assigned a random position in the neighbourhood of probabilistically selected employed bee using formulae (2). Greedy selection is applied same as above.

$$P_i = \left(\frac{fitness(i)}{\sum_{i=1}^{SN} fitness(i)} \right), i \in [1, SN] \tag{3}$$

P_i- Probability of i^{th} employed bee

$fitness(i)$- Fitness of i^{th} employed bee

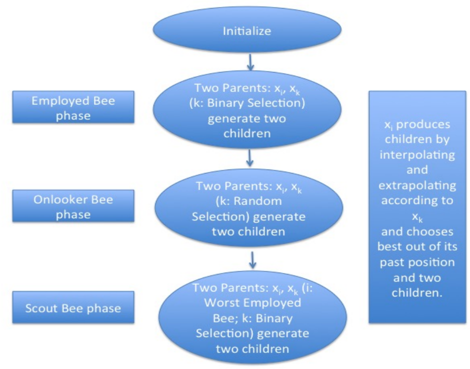

Figure 1: Mutated Artificial Bee Colony algorithm

Scout bee phase:

There can be either one or zero scout bee in each cycle. Employed bee whose position has not changed for predefined number of cycles, abandons its position and is assigned a random position in the search space using formula (1).

UNTIL (stopping criteria is met)

3 PROPOSED ALGORITHM

The performance of an algorithm depends on its parameters; thus tuning parameters of an algorithm is an important task as it gives an insight of how an algorithm works. Optimum parameter values for a problem can be different for another problem considering a particular algorithm. Artificial Bee Colony algorithm has few hidden parameters which play important role in working of ABC algorithm. This paper explores some of them. Parameter set for mutated ABC is shown below:

$(SN, D, MCN, EB:OB, SB, Select_{EB}, Select_{OB}, RNG)$

- SN- Number of employed Bees or Number of onlooker bees
- D- Dimension in search space
- MCN- Maximum Cycle Number
- $EB:OB$- Ratio of Employed Bee to Onlooker Bee
- SB- Number of Scout Bee
- $Select_{EB}$- Selction method of k^{th} bee in Employed Bee phase
- $Select_{OB}$- Selction method of k^{th} bee in Onlooker Bee phase
- RNG- Random Number Generator in v_{ij}

Figure 1 shows Mutated Artificial Bee Colony algorithm where x_i and x_k are employed bees in the previous cycle. In the Initialization phase, all bees in the colony are assigned a random position between their dimensional limits. To further describe proposed algorithm a comparison of ABC and GA is made in this section and their best features are hybrid. Also, parameter are tuned to best suit four classes of problems, namely Unimodal Separable (US), Multimodal Separable (MS), Unimodal Non-separable (UN), and Multimodal Non-separable (MN).

Artificial Bee Colony algorithm differs from Genetic Algorithm in following ways: ABC algorithm is a two-tier process (Employed bee phase and Onlooker Bee phase) whereas GA is the one-tier process. In ABC, employed bee phase selects two bees from the last cycle called parents, operates them to produce two children and after comparing children's fitness with bee's past fitness, adds best to current cycle whereas onlooker bees phase selects two employed bees from current generation called parents, operates them to produce two children and after comparing their fitness with its past fitness, it selects best to add in current cycle. GA selects candidates from the current generation as parents, operates them producing children and adds best in next population.

All optimisation algorithm needs a right balance between Intensification and Diversification. Employed bees are responsible for diversification and onlooker for intensification. ABC is good at exploiting a solution but poor in exploration. To approach this problem equation (2) is divided into following two parts:

$$v_{ij} = x_{ij} + rand[-1, 0] \cdot (x_{ij} - x_{kj}), k \in [1, SN], i \neq k \tag{4}$$

$$v_{ij} = x_{ij} + rand[0, 1] \cdot (x_{ij} - x_{kj}), k \in [1, SN], i \neq k \tag{5}$$

In the employed bee phase, bee attains a position which is away from k^{th} bee using first equation. The second equation takes bee towards k^{th} bee. Here k^{th} bee is selected using binary tournament selection. At the end of employed bee phase, each employed bees chooses the best position among x_i and two children produced by parents x_i and x_k.

Onlooker bee phase exploits the neighbourhood of selected employed bees. Each onlooker bee probabilistically selects an employed bee from the current population as the first parent and a k^{th} employed bee is selected randomly as the second parent. Using equation (4) and (5) two children are produced and finally attaining a position that is best among three positions- its past position and two children. Unlike employed bee phase, this process is applied to a randomly selected dimension of chosen employed bee.

Proposed algorithm is called mutated ABC because it works like a mutation operator where each dimension (or each gene in GA) is selected with some probability and changed to another bit if selected for change. This is responsible for exploration in the Genetic algorithm. Mutation probability is set to 0.01. Thus in both phases, x_i either goes towards or away from the k^{th} bee and the best position is chosen according to fitness.

In traditional ABC, employed bee whose position has not been changed for a predefined number of cycles (limit) becomes scout bee. If the value of limit is too small, there is a chance that good bee position is loosed. If limit value is large, it results in unnecessary delays in convergence. There needs to be 'right' limit value which is hard to predict as it can be problem dependent. Due to confusion in 'limit' value, the concept of 'limit' is not used in proposed algorithm and instead is replaced by average fitness which is calculated in each cycle for colony size and if worst employed bee's fitness is greater than average fitness (in case of minimization problems) its position is perturbed as in employed bee phase unlike standard ABC algorithm where a new random position is assigned without considering 'goodness' of current position as this can assign a position farther away from optima.

4 EXPERIMENTAL RESULTS

Experiments are conducted on a set of benchmark functions given in Table 1 of paper [10]. Each function belongs to one of the classes of problem: US, MS, UN and MN. Each problem was carried out for 30 runs and the number of function evaluations is 500,000 i.e. colony size is 50 and number of cycles are 10,000. The values less than 10^{-15} is taken as 0. Each function problem is set for fixed dimension (D) and interval for all experiments. The parameter $EB : OB$ is set same as the original ABC algorithm that is 1:1. The number of scout bee is equal to 1 in all cycles for all experiments. These specifications are matched with Karaboga's (2016) paper [10]. All simulation results are obtained by an algorithm implemented in Python 3.5.

Table 1 gives results using proposed algorithm. Mean, Standard Deviation (SD), the best solution and the worst solution obtained by mutated ABC are shown along with last column showing best results using aABC [10]. This table shows results when the algorithm is run 30 times. Mutation probability is set to 0.01 for all functions except for Dixon-Price (Unimodal Nonseparable) function whose mutation probability is 1.0 which means that each dimension of an employed bee undergoes change as it needs more exploration of search space. Two children are produced, one away from employed bee and other towards employed bee. These new positions are compared with bee's previous cycle position and the best is chosen as the new position of that employed bee. Thus, compared to aABC algorithm best solution is obtained for US, MS and MN class functions. Near optimal solutions are obtained for UN class of functions.

The mean value for each function generated by proposed algorithm are compared with six other algorithms- GA, PSO, DE, ABC, qABC and aABC algorithm as shown in Table 2. These values for above mentioned six algorithms are taken directly from paper [10] for comparison. The results by best algorithm for a function is marked by boldface. Unimodal Separable- Sphere function, when tested on these algorithms, has given best result i.e. 0 for all algorithms including mutated ABC except GA. Unimodal non-separable function includes Rosenbrock and Dixon-Price. aABC has outperformed all algorithms for these two functions but for Dixon-Price, ABC has also shown the best result which is 0. Comparing results for Multimodal Sparable functions- Rastrigin, Schwefel and Branin, it can be noticed from Table 2 that for all these functions mutated ABC has shown best results compared to other algorithms. Considering last class which is Multimodal Non-Separable includes Griewank, Schaffer, Ackley and SixHumpCameBack, it is noticed that ABC and all its variants have shown best mean results for Griewank function. For Schaffer function, PSO, DE, aABC and mutated ABC have shown best results. For Ackley function, DE, ABC along with all variants of ABC have shown best results. Lastly, for SixHumpCameBack, GA has outperformed all other algorithms. Thus, it can be generalised that proposed algorithm is successful in moving out of local optima to reach global solution.

5 CONCLUSION AND FUTURE WORK

In this paper a novel variant of Artificial Bee Colony algorithm has been proposed which explores new parameters of ABC and

Table 1: Test on benchmark functions using mutated ABC

Test function	Global Minima	Mean	SD	Worst	Best-mutated ABC	Best-aABC
Sphere	$F_{min} = 0$	0	0	0	**0**	**0**
Rosenbrock	$F_{min} = 0$	0.085543	0.127395	0.681654	0.001961	**2.1913E-005**
Rastrigin	$F_{min} = 0$	0	0	0	**0**	**0**
Griewank	$F_{min} = 0$	0	0	0	**0**	**0**
Schaffer	$F_{min} = 0$	0	0	0	**0**	**0**
Dixon-Price	$F_{min} = 0$	0.000116504	0.000367662	0.001937097	1.44909E-08	**2.2822E-015**
Ackley	$F_{min} = 0$	0	0	2.84E-14	**0**	2.2204E-014
Schwefel	$F_{min} = -12569.5$	-12569.487	5.55E-12	-12569.487	**-12569.487**	**-12569.487**
SixHumpCameBack	$F_{min} = -1.03163$	-1.0316284	0	-1.0316284	**-1.0316284**	**-1.0316284**
Branin	$F_{min} = 0.398$	0.409121	0.020377	0.482377	**0.3978949**	0.398874

Table 2: Mean result on benchmark functions using ABC and mutated ABC

Test function	GA	PSO	DE	ABC	qABC	aABC	Mutated ABC
Sphere	1.11E+03	**0**	**0**	**0**	**0**	**0**	**0**
Rosenbrock	1.96E+05	15.088617	18.203938	0.1766957	0.1329198	**0.0246333**	0.085543
Rastrigin	52.92259	43.977137	11.716728	**0**	**0**	**0**	**0**
Griewank	10.63346	0.0173912	0.0014792	**0**	**0**	**0**	**0**
Schaffer	0.004239	**0**	**0**	1.04E-10	8.66E-06	**0**	**0**
Dixon-Price	1.22E+03	0.6666667	0.6666667	**0**	1.15E-12	**0**	1.91E-02
Ackley	14.67178	0.1646224	**0**	**0**	**0**	**0**	**0**
Schwefel	-11593.40	-6909.1359	-10.266	**-12569.49**	**-12569.49**	**-12569.49**	**-12569.49**
SixHumpCameBack	**-1.03163**	-1.0316285	-1.031628	-1.0316284	-1.0316284	-1.0316284	-1.0316284
Branin	**0.397887**	0.3978874	0.3978874	0.3978874	0.3978874	0.3978874	0.409121

tunes few of them to enhance exploration ability of ABC. The mutation operator borrowed from Genetic Algorithm has proved useful by interpolating and extrapolating the position of bees in finding new better solutions in their neighbourhood. Replacing 'limit' parameter with the average fitness of bees has been successful in perturbing position of employed bee and finding global minima. Mutated ABC is used on four classes of problems and shown best results for Unimodal Separable, Multimodal Separable, Multimodal Non-separable function and near optimal solution for Unimodal Separable Non-separable fnction. Proposed algorithm is at par with aABC algorithm and has shown better results than GA, PSO, DE, ABC and qABC for numerical optimisation.
Mutated ABC is required to be tested on a real-world problem. Identifying and implementing the proposed variant of ABC on other classes of problems to analyse what properties of mutated ABC are best suited for a particular class of problem.

REFERENCES

[1] Th Bäck, Agoston Endre Eiben, and Nikolai AL van der Vaart. 2000. An emperical study on GAs fiwithout parametersfi. In *International Conference on Parallel Problem Solving from Nature*. Springer, 315–324.
[2] Marco Dorigo. 2007. Ant colony optimization. *Scholarpedia* 2, 3 (2007), 1461.
[3] Ágoston E Eiben, Robert Hinterding, and Zbigniew Michalewicz. 1999. Parameter control in evolutionary algorithms. *IEEE Transactions on evolutionary computation* 3, 2 (1999), 124–141.
[4] DE Golberg. 1989. Genetic algorithms in search, optimization and machine learning reading. *MA: Addisonn-Wisley, USA* (1989).
[5] John J Grefenstette. 1986. Optimization of control parameters for genetic algorithms. *IEEE Transactions on systems, man, and cybernetics* 16, 1 (1986), 122–128.
[6] M Gunasekaran and KS Ramaswami. 2011. A fusion model integrating anfis and artificial immune algorithm for forecasting indian stock market. (2011).
[7] Dervis Karaboga. 2010. Artificial bee colony algorithm. *scholarpedia* 5, 3 (2010), 6915.
[8] Dervis Karaboga and Bahriye Akay. 2009. A comparative study of artificial bee colony algorithm. *Applied mathematics and computation* 214, 1 (2009), 108–132.
[9] Dervis Karaboga and Ebubekir Kaya. 2016. An adaptive and hybrid artificial bee colony algorithm (aABC) for ANFIS training. *Applied Soft Computing* 49 (2016), 423–436.
[10] Dervis Karaboga and Ebubekir Kaya. 2016. An adaptive and hybrid artificial bee colony algorithm (aABC) for ANFIS training. *Applied Soft Computing* 49 (2016), 423–436.
[11] Dervis Karaboga and Celal Ozturk. 2011. A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Applied soft computing* 11, 1 (2011), 652–657.
[12] Nurhan Karaboga. 2009. A new design method based on artificial bee colony algorithm for digital IIR filters. *Journal of the Franklin Institute* 346, 4 (2009), 328–348.
[13] James Kennedy. 2011. Particle swarm optimization. In *Encyclopedia of machine learning*. Springer, 760–766.
[14] D Jeya Mala, V Mohan, and M Kamalapriya. 2010. Automated software test optimisation framework-an artificial bee colony optimisation-based approach. *IET software* 4, 5 (2010), 334–348.
[15] Zbigniew Michalewicz and Stephen J Hartley. 1996. Genetic Algorithms+ Data Structures= Evolution Programs. *Mathematical Intelligencer* 18, 3 (1996), 71.
[16] Mudita Sharma and Satish Chandra. 2015. Application of Artificial Bee Colony Algorithm for numerical optimization technique. In *Advance Computing Conference (IACC), 2015 IEEE International*. IEEE, 1267–1272.
[17] Jim Smith and Terence C Fogarty. 1996. Self adaptation of mutation rates in a steady state genetic algorithm. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. IEEE, 318–323.
[18] Reza Teimouri and Hamid Baseri. 2014. Optimization of magnetic field assisted EDM using the continuous ACO algorithm. *Applied Soft Computing* 14 (2014), 381–389.
[19] Guopu Zhu and Sam Kwong. 2010. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl. Math. Comput.* 217, 7 (2010), 3166–3173.