ISSN:0975-9646



# Modified Animal Migration Optimization Algorithm for Numerical Function Optimization

AmrataSaraswat[#1], Dushyant Singh[*2]
[#1] *M.Tech Pursing RTU*
**Hod Of Computer Science , Chandravati Group of Institute, Bharatpur, India*

***Abstract*— Animal Migration Optimization Algorithm is a Nature Inspired Algorithm (NIA) that mimics the intelligent behavior of animals during migration from one location to other location in search of better food source and safe shelter. This paper proposed a Modified Animal Migration Optimization (MAMO) Algorithm. The proposed modified Animal Migration Optimization Algorithm improves both migration and position update steps. The MAMO algorithm updates position using difference of two randomly selected position and current position. The MAMO algorithm tested on a set of standard problem and compared with ABC.**

***Keywords*— animal migration optimization algorithm; nature inspired algorithm; swarm intelligence; differential evolution***

## I. INTRODUCTION

Animal Migration Optimization (AMO) algorithm is most recent swam intelligence based optimization technique proposed by X. Li [1]. The AMO algorithm is very popular among researchers and scientists who are working on optimization problems. There are number of trifling multivariable optimization problems with capriciously high dimensionality which cannot be solved by precise search methods in stirred time. So search algorithms capable of searching near-optimal or good solutions within adequate computation time are very realistic in real life. In few years, the technical community has noticed the importance of a large number of nature-inspired metaheuristics and hybrids of these nature-inspired optimization methods. Metaheuristics may be measured a widespread algorithmic skeleton that can be applied to poles apart optimization problems with comparative a small number of modifications to get a feel for them to a specific problem. Metaheuristics are anticipated to make bigger the capabilities of heuristics by hybridizing one or more heuristic strategies using a higher-level methodologies (hence 'meta'). Metaheuristics are strategies that provide guidance to the search process. Hyperheuristics are up till now an additional extension that focuses on heuristics that adapt their parameters in order to get better efficacy or result, or the effectiveness of the computation progression. Hyperheuristics endow with high-level methodologies that possibly will make use of machine learning and get a feel for their search behavior by modifying the application of the sub-procedures or even which procedures are used [2]. Algorithms on or after the meadow of computational intelligence, biologically inspired intelligent computing, and metaheuristics are applied to troublesome problems, to which more classical approaches may not be significant. Michalewicz and Fogel says that these tribulations are difficult [3] as: they has large number of feasible solutions in the search space due to which they not able to exploit the best results; The problem is so intricate, that just to facilitate any reply at all, we have to make use of such beginner's models of the problem that any consequence is in essence a waste of time; the appraisal function that describes the quality of whichever proposed explanation is noisy or varies with time, by this means requiring not just a solitary solution but an entire series of solutions; the promising elucidations are so deliberately constrained that constructing even single feasible answer is incredibly easier said than done, let alone searching for an most advantageous solution; the human being solving the problem is inadequately composed or imagines some psychological fencing that prevents them from discovering exact solution.

Nature inspired algorithms are encouraged by some natural happening, can be categorized as per their source of encouragement. Major classes of NIA are: Evolutionary Algorithms, Immune Algorithms, Neural Algorithms, Physical Algorithms, Probabilistic Algorithms, Stochastic Algorithms and Swarm Algorithms [2].

Evolutionary Algorithms are motivated by advancement of natural selection strategy. Evolutionary Algorithms fit into the Evolutionary Computation field of learning concerned with computational methods encouraged by the itinerary of action and mechanisms of biological progression. Examples of evolutionary algorithm are Differential Evolution (EA), Evolutionary Programming (EP), Evolution Strategies (ES), Gene Expression Programming, Genetic Algorithm (GA), Genetic Programming (GP), Grammatical Evolution, Learning Classifier structure, Non-dominated Sorting Genetic Algorithm, and Strength Pareto Evolutionary Algorithm [2].

Immune Algorithms are aggravated by the adaptive immune system of vertebrates. A simplified narration of the immune organization is an appendage system anticipated to care for the host organism from the intimidation posed to it from pathogens and noxious substances. Pathogens include an assortment of microorganisms such as bacteria, viruses, parasites and pollen. The conventional viewpoint regarding the responsibility of the immune system is alienated into two most important tasks: the detection and elimination of pathogen. This activity is classically referred to as the delineation of self (molecules and cells that are in the right place to the host organisms) from potentially destructive non-self. Like Clonal Selection Algorithm (CSA), Negative Selection Algorithm, Artificial Immune Recognition System, Immune Network Algorithm and Dendritic Cell Algorithm [2].

Neural Algorithms are encouraged by the flexibility and learning individuality of the human nervous coordination.

Some well known neural algorithms are Perceptron, Back-propagation, Hopfield Network, Learning Vector Quantization and Self-Organizing Map [2].

Physical Algorithms are motivated by corporal and communal systems Physical algorithms are those algorithms motivated by a physical process. Most of the physical algorithm in general belong to the fields of metaheustics and Computational cleverness; even though do not fit neatly into the obtainable categories of the biological motivated techniques. In this vein, they could immediately as by far be referred to as nature inspired algorithms. Like Simulated Annealing, External Optimization, Harmony Search, Cultural Algorithm, and Memetic Algorithm [2].

Probabilistic Algorithms are strategies that concentrate on methods that put together models and guesstimate distributions in search domains. Probabilistic Algorithms are those algorithms that sculpt a dilemma or explore a problem space using a probabilistic model of entrant solutions. Examples of probabilistic algorithms are Population-Based Incremental Learning, Univariate Marginal Distribution Algorithm, Compact Genetic Algorithm, Bayesian Optimization Algorithm and Cross-Entropy Method [2].

Stochastic Algorithms are algorithms that focus on the prologue of unpredictability into heuristic methods. Examples of stochastic algorithms are Random Search, Adaptive Random Search, Stochastic Hill Climbing, Iterated Local Search, Guided Local Search, Variable Neighborhood Search, Greedy Randomized Adaptive Search, Scatter Search, Tabu Search, and Reactive Tabu Search [2].

There are number of new algorithms in class of swarm intelligence algorithms that are inspired by intelligent behavior of living beings like firefly algorithm (FA) [3] , cuckoo search (CS) [4, 5] , bat algorithm (BA) [6] , artificial bee colony (ABC) [] , monkey algorithm (MA) [8] , frog-leaping algorithm (SFLA) [9 ,10] . These algorithm mimics swarming behavior of animals and birds. Each algorithm shows extraordinary behavior while searching for food, mating, migrating. Another class of algorithms that mimic the natural phenomenon that regularly occurs in nature like Optics Inspired Optimization, Black Holes, Gases Brownian Motion, Forest Optimization, Golden Ball, Mine Blasting, Seed based Plant Propagation and Vortex based Searching etc.

## II. ANIMAL MIGRATION OPTIMIZATION ALGORITHM

Animal migration algorithm can be divided into animal migration process and animal updating process. In the migration process the algorithm simulates how the groups of animals move from current position to a new position. During this process, each individual should obey three main rules: (1) move in the same direction as its neighbors; (2) remain close to its neighbors; (3) avoid collisions with its neighbors. During the population updating process, the algorithm simulates how animals update by the probabilistic method.

**Animal Migration Process**

During the animal migration process, an animal should obey three rules: (1) avoid collisions with your neighbors; (2) move in the same direction as your neighbors; and (3) remain close to your neighbors. In order to define concept of the local neighborhood of an individual, we use a topological ring, as has been illustrated in Fig. 1. For the sake of simplicity, we set the length of the neighborhood to be five for each dimension of the individual. Note that in our algorithm, the neighborhood topology is static and is defined on the set of indices of vectors. If the index of animal is i, then its neighborhood consists of animal having indices i-1, i-2,i,i+1,i+2 , if the index of animal is 1, the neighborhood consists of animal having indices NP-1,NP, 1, 2, 3, etc. Once the neighborhood topology has been constructed, we select one neighbor randomly and update the position of the individual according to this neighbor, as can be seen in the following formula:

$X_{i,\ G+1}=X_{i,G}+\$.(X_{neighbouhood,,G} - X_{i,G})$

Where $X_{neighbouhood,,G}$ is the current position of the neighborhood,\$ is produced by using a random number generator controlled by a Gaussian distribution. $X_{i,G}$ is the current position of i th individual, and $X_{i,\ G+1}$ is the new position of i th individual.

During the population updating process, the algorithm simulates how some animals leave the group and some join in the new population. Individuals will be replaced by some new animals with a probability *Pa*. The probability is used according to the quality of the fitness. We sort fitness in descending order, so the probability of the individual with best fitness is 1/*NP*, the individual with worst fitness, by contrast, the probability is 1, and this process can be shown in Algorithm 1.

```
For i=1 to NP
For j=1 to D
    If rand>Pa
  X_{i,g+1}=X_{r1,G}+rand.(X_{best,G}-X_{i,G})+rand.(X_{r2,G},-X_{i,G})
    End If
End For
End For
```

Algorithm 1: Animal Migration Optimization Algorithm[1]

## III. MODIFIED ANIMAL MIGRATION OPTIMIZATION (MAMO) ALGORITHM

Animal Migration Optimization Algorithm is a Nature Inspired Algorithm (NIA) that mimics the intelligent behavior of animals during migration from one location to other location in search of better food source and safe shelter. The basic AMO algorithm mainly has two steps. First is process of migration from one location to other location or current position replaced by new position. Second process is population update. In this step some new animals join the group in place of abandoned population.

This paper proposed a Modified Animal Migration Optimization (MAMO) Algorithm. The MAMO algorithm suggests two changes in original AMO algorithm. The proposed modified Animal Migration Optimization Algorithm improves both migration and position update steps. The MAMO algorithm updates position using difference of two randomly selected position and current position. This algorithm also changes neighborhood policy of basic AMO algorithm. The proposed MAMO algorithm replace Gaussian distribution based random number generator by simple function.

**Algorithm 2: Modified AMO Algorithm**

1. Initialize the population of *NP* evenly distributed animals $x_i$. Also set counter and other parameters.

$$x_{i,j,0} = x_{j,\min} + rand_{i,j}[0,1] \times (x_{j,\max} - x_{j,\min})$$

2. Evaluate the fitness of each individual using

```
if(fun_value >= 0)
  {
    Fitness = (1/(2* fun_value +1))
  }
 else
  {
    Fitness = (1+fabs(1/fun_value))
  }
```

Repeat Step 3 to while stoping criteria meets

3. Modified Migration Phase

*for i = 1 to NP do*

*for i = 1 to D do*

$$x_{i;G+1} = x_{i;G} + rand[0,1] \times \left(x_{neighbourhood;G} - x_{i;G}\right)$$

Here G is counter.

Value of neighbourhood decided by randomly using

$$neighbourhood = (\mathrm{int})(i + NP + rand[0,1])$$

4. Select Improved Phase

*for i = 1 to NP do*

Evaluate the offspring $x_{i;G+1}$ and apply greedy selection among existing and new population.

$$p_i = 0.9 \times \frac{fitness_i}{fitness_{\max}} + 0.1$$

5. Modified Population Update Phase

*for i = 1 to NP do*

*for i = 1 to D do*

$$x_{i;G+1} = x_{r1;G} + rand[0,1] \times \left(x_{r1;G} - x_{i;G}\right)$$
$$+ rand[0,1] \times \left(x_{r2;G} - x_{i;G}\right)$$

$r_1$ and $r_2$ are randomly generated within population.

6. Select Improved Phase

*for i = 1 to NP do*

Evaluate the offspring $x_{i;G+1}$ and apply greedy selection among existing and new population.

$$p_i = 0.9 \times \frac{fitness_i}{fitness_{\max}} + 0.1$$

7. Memorize the best solution found so far.

In the proposed MAMO algorithm, the perturbation in the solution depends on the fitness of the solution and probability of selection for next iteration. It is clear from Algorithm 2 that the fitness depends on quality of solution as fitness is function of function value. Probability of selection depends on fitness of individual. Highly fitted solutions are most feasible for next iteration. It is also assumed that the global optima should be near about to the better fit solutions. Therefore, in the proposed MAMO strategy, the better solutions speedup rate of convergence.

## IV. Selected Problems For Experiment

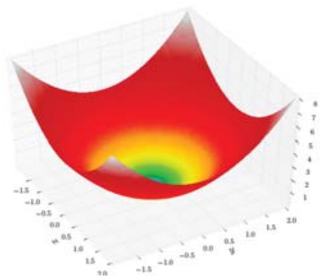
Animal Migration Optimization algorithm with modifications in position update equation and an additional step applied to some benchmark functions for whether it gives better result or not. Benchmark functions taken in this paper are of different characteristics like uni-model or multi-model and separable or non-separable and of different dimensions. In order to analyze the performance of MAMO it is applied to these global optimization problems.

**Uni-modal functions**

Function $f_1$ to $f_4$ are uni-modal functions.

1. The Sphere function defined as

$$f_1(x) = \sum_{i=1}^{n} x_i^2$$



Where D = 30, optimum value *f(0) = 0* in search range [-5.12, 5.12] with acceptable error 1.0E-05.

2. The Schewel function described as

$$f_2(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|$$

Where D = 30, optimum value *f(0) = 0* in search range [-10, 10] with acceptable error 1.0E-05.

3. The Rosenbrock function defined as

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

Where D = 30, optimum value *f(0) = 0* in search range [-30, 30] with acceptable error 1.0E-01.

4. The Step Function described as follow

$$f_4(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$$

Where D = 30, optimum value $f(-0.5 \le x \le 0.5) = 0$ in search range [-100, 100] with acceptable error 1.0E-05.

**Multimodal High dimensional functions**

Function $f_5$ to $f_8$ are Multimodal high dimensional modal functions.

5. The Rastrigin function described by

$$f_5(x) = \sum_{i=1}^{D}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$$

Where D=30, optimum value *f(0) = 0* in search range [-5.12, 5.12] with acceptable error 1.0*E*-05.

6. The Shifted Ackley function defined as follow

$$f_6(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i))$$
$$+20 + e + f_{bias}, z = (x - o), x = [x_1, x_2, ...x_D], o = [o_1, o_2, ...o_D]$$

Where D=10, optimum value *f(o)=fbias=-140* in search range [-32, 32] with acceptable error 1.0*E*-05.

7. The Griewank function described by

$$f_7(x) = \frac{1}{4000}\left(\sum_{i=1}^{D}(x_i^2)\right) - \left(\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)\right) + 1$$

Where D=30, optimum value *f(0) = 0* in search range [-600, 600] with acceptable error 1.0*E*-05.

8. The Levy Montalvo -1 problem defined as

$$f_8(x) = \frac{\pi}{D}(10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2(1 + 10\sin^2(\pi y_{i+1}))$$
$$+(y_D - 1)^2), Where\ \ y_i = 1 + \frac{1}{4}(x_i + 1)$$

Where D=30, optimum value *f(-1) = 0* in search range [-10, 10] with acceptable error 1.0*E*-05.

**Multimodal Low dimensional functions**

Function $f_9$ to $f_{11}$ are Multimodal low dimensional modal functions.

9. The Kowalik function described by

$$f_9(x) = \sum_{i=1}^{11}(a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4})^2$$

Where D=4, optimum value *f(0.1928, 0.1908, 0.1231, 0.1357) = 3.07E-04* in search range [-5, 5] with acceptable error 1.0*E*-05.

10. The Six-hump camel back problem explained as

$$f_{10}(x) = (4 - 2.1x_1^2 + \frac{1}{3}x_1^4)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

Where D=2, optimum value *f(-0.0898, 0.7126) = -1.0316* in search range [-5, 5] with acceptable error 1.0E-05.

11. The Goldstein-Price problem defined as follow

$$f_{11}(x) = (1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2$$
$$-14x_2 + 6x_1x_2 + 3x_2^2)) \times (30 + (2x_1 - 3x_2)^2$$
$$\times(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$$

Where D=2, optimum value *f(0, -1)=3* in search range [-2, 2] with acceptable error 1.0E-14.

## V. Experimental Results and Discussion

Newly proposed MAMO algorithm coded in C language, and experiments are performed on a Pentium 3.0 GHz Processor with 4.0 GB of memory. In the test conducted in this section, success of MAMO algorithm introduces in this paper has been compared with the success of some other well known swarm intelligence algorithms.

Comparison of the Results of Test Problems for MAMO

| Test Problem | Algorithm | SD | ME | AFE | SR |
|---|---|---|---|---|---|
| $f_1$ | MAMO | 2.75E-06 | 6.49E-06 | 15006.58 | 100 |
| | ABC | 2.09E-06 | 7.86E-06 | 23218 | 100 |
| $f_2$ | MAMO | 1.66E-06 | 8.49E-06 | 24873.66 | 100 |
| | ABC | 5.86E-07 | 9.47E-06 | 64664.5 | 100 |
| $f_3$ | MAMO | 7.18E-01 | 2.43E+01 | 100016 | 0 |
| | ABC | 1.25E+01 | 9.33E+00 | 98135.5 | 11 |
| $f_4$ | MAMO | 0.00E+00 | 0.00E+00 | 11816.1 | 100 |
| | ABC | 0.00E+00 | 0.00E+00 | 18030.84 | 100 |
| $f_5$ | MAMO | 3.16E-06 | 4.94E-06 | 27225.86 | 100 |
| | ABC | 1.58E+00 | 3.41E+00 | 99490 | 2 |
| $f_6$ | MAMO | 7.47E-01 | 2.03E+01 | 100016 | 0 |
| | ABC | 5.26E-01 | 2.04E+01 | 100000.8 | 0 |
| $f_7$ | MAMO | 2.90E-06 | 5.41E-06 | 22657.12 | 100 |
| | ABC | 7.51E-03 | 4.36E-03 | 76412 | 68 |
| $f_8$ | MAMO | 1.23E-02 | 3.77E-02 | 100016 | 0 |
| | ABC | 2.24E-06 | 7.55E-06 | 30044 | 100 |
| $f_9$ | MAMO | 1.90E-04 | 1.83E-04 | 67082.16 | 63 |
| | ABC | 7.40E-05 | 1.71E-04 | 90212.85 | 19 |
| $f_{10}$ | MAMO | 5.47E-02 | 5.71E-02 | 100016 | 0 |
| | ABC | 3.92E-03 | 3.80E-03 | 99759.02 | 1 |
| $f_{11}$ | MAMO | 4.36E-15 | 5.06E-15 | 11651.94 | 100 |
| | ABC | 5.92E-06 | 2.12E-06 | 82426.71 | 26 |

## References


[1] Li, Xiangtao, Jie Zhang, and Minghao Yin. "Animal migration optimization: an optimization algorithm inspired by animal migration behavior." Neural Computing and Applications 24.7-8 (2014): 1867-1877.

[2] Brownlee, Clever algorithms: nature-inspired programming recipes. Jason Brownlee, 2011.

[3] X-S Yang. Nature-Inspired Metaheuristic Algorithms, Luniver Press, USA.(2008)

[4] Selvi,"Improved Cuckoo Search Algorithm for Job Scheduling Problem", International Journal of Advanced Research in Computer Science and Software Engineering 5(10),October-2015, pp. 230-234

[5] 2. X-S Yang, Deb S. Cuckoo search via Levy flights, in: world congress on nature & biologically inspired computing (NaBIC 2009). IEEE Publication, USA, 210-214.(2009)

[6] 3. X-S Yang. A New Metaheuristic Bat-Inspired Algorithm. Nature Inspired Cooperative Strategies for Optimization. Berlin, Germany: Springer Ver-lag, 65-74.(2010)

[7] 4. Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. J Global Optim, Vol.39,No.3,459-471.(2007)

[8] 5. Zhao R, Tang W. Monkey algorithm for global numerical optimization. Journal of Uncertain Systems. Vol.2,No. 3,164-175.(2008)

[9] 6. Eusuff M, Lansey K. Optimization of water distribution network design using the shuffled frog leaping algorithm. Journal of Water Resources Planning and Management, Vol.129,No.3,210-225.(2003)

[10] 7. Eusuff M, Lansey K, Pasha F. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, Engineering Optimization, Vol.38, No.2, 129-154,(2006)

[11] Z. Michalewicz and DB Fogel. How to Solve It: Modern Heuristics. Springer, 2004.