A COMPARITIVE STUDY OF NATURE INSPIRED ALGORITHMS FOR AUTISM SPECTRUM DISORDER(ASD) DETECTION

Vinayak Kukreja
SU-ID: 580935977
Syracuse University
Syracuse, USA
vkukreja@syr.edu

Rajas Deshpande
SU-ID: 588939943
Syracuse University
Syracuse, USA
rdeshpan@syr.edu

***Abstract* - It is no exaggeration that optimization has surrounded our everyday life and we often deal with utilizing a commodity in such a way that reaps maximum benefit. Mathematically, we can define optimization as searching of the decision space for decision variables that could produce maximum acceptability considering our goal. Nature inspired algorithms are one way to achieve optimization in various problems. Here in our study we are using five nature inspired algorithms for detection of Autism Spectrum Disorder. The algorithms that we are using are Cuckoo Optimization Algorithm, Grey Wolf Optimization, Firefly Algorithm, Cat Swarm Optimization, and Bat Algorithm. We are doing a comparative study of these algorithms and comparing their accuracy against each other for detection of Autism Spectrum Disorder. This document is intended to study the aforementioned five nature inspired algorithms and discuss their efficiency in detection of Autism Spectrum Disorder.**

I. Introduction

Nature based algorithms are a sub-category of a branch of artificial intelligence known as computational intelligence. Optimization algorithms that imitate any biological, physical or chemical phenomenon can be classified as nature-based optimization algorithm.

Two major components of nature-based algorithms are intensification and diversification. Intensification means to focus on the search in a local region by exploiting the area surrounding the current good solutions. Whereas, diversification means to generate diverse solutions so as to explore the search space. In most cases, the intensification component helps ensure the algorithms convergence to a global optimum solution whereas the diversification or random based component helps to thoroughly investigate the decision space and divert the solutions to be trapped in local optima. The combination of these two components ensures achieving global optimality.

Here we are using the Cuckoo Search Optimization Algorithm, Grey Wolf Optimization, Firefly Algorithm, Cat Swarm Optimization, and Bat Algorithm to identify Autism Spectrum Disorder or ASD. These nature-based algorithms are famous for solving optimization problems and has presented us with great results while detection for the disorder.

II. Problem and Data Description

In order to perform a comparative study of these algorithms for ASD detection, we have taken a labelled dataset. This dataset is open-

source and can be downloaded from this website - http://archive.ics.uci.edu/ml/machine-learning-databases/00426/

This dataset consists of 17 features which are used to distinguish between a person having ASD and a person not having ASD. The data has been pre-processed to remove the unwanted features and converted to numerical data so that the optimization algorithms can be applied to this data. Our primary goal is to model a neural network and calculate the weights and biases of the neural network by using these optimization algorithms and find out the algorithm which gives the best accuracy. The pre-processed data has been stored in the Datasets folder under the name of "Final_Dataset.csv".

A three-layer neural network has been designed which 32 neurons in the first two layers and one neuron in the output layer which will essentially provide the output in a binary form – 0 being a person with no ASD and 1 being a person with ASD. The number of trainable parameters which include the weights and biases of the network has been calculated. In our case, there are a total of 1633 trainable parameters. This means that the dimension parameter of all of our optimization algorithms will be 1633. All of these algorithms will find a best global solution whose size will be 1633 and this best solution will then be fitted into the model and corresponding results will be calculated.

III. Approach

Nature-based algorithms can be of two types, i.e. single point optimization algorithms and population-based optimization algorithms. In our study, we are emphasizing on five nature-based population-based algorithms and compare them with each other as to detect which of these perform the best for detection of Autism Spectrum Disorder.

The first algorithm that we used is Cat Swarm Optimization Algorithm (CSO). This algorithm was developed by Chu and Tsai in 2007 and it is based on the behavior of cats. CSO algorithm and its varieties have been implemented for different optimization problems. Some of the variations of this algorithm were parallel CSO or PCSO, enhanced version of PCSO or EPCSO which were developed by Tsai et al. Another version was binary CSO or BCSO which was developed by Sharafi et al. There are many more versions of CSO algorithm but here we used the basic CSO algorithm.

Cats, despite spending most of their time in resting, have high curiosity and alertness for moving objects and their environment. This is an essential attribute for them to seek prey and hunt them down. The time spent by cats resting is much more than that they spend chasing preys as they focus on conserving their energy. This hunting pattern is what inspired the creation of CSO algorithm.

In CSO, a population of cats is created and randomly distributed in M-dimensional solution space where each cat represents the solution. The populations are divided into two subgroups. The cats in one of the groups are resting and keeping an eye on their surroundings. This is known as seeking mode. The cats inf the second subgroup are moving around and chasing the preys. This is known as tracing mode. With the help of these two modes, CSO is able to move towards the global solution in the M-dimensional solution space. As mentioned earlier, the cats spend most of the time resting, therefore number of cats in tracing mode must be less. This is represented by Mixture Ratio (MR) in the algorithm. This MR has a small value in the algorithm.

After the cats are sorted in seeking and tracing group, new positions and fitness functions will be available from which the cat with the best solution will be saved in memory. The aforementioned steps are repeated till a stopping criterion is satisfied. In CSO, number of iterations, the amount of improvement, and the running time are common termination criteria.

The following are the characteristics of the CSO algorithm,

Decision variable	Cat's position in each dimension
Solution	Cat's position
Old solution	Old position of cat
New solution	New position of cat
Best solution	Any cat with the best fitness
Fitness function	Distance between cat and prey
Initial solution	Random position of cats
Selection	-
Process of generating new solution	Seeking and tracing a prey

Following Chu and Tsai (2007), the computational procedures of CSO can be described as follows:

1. Create the initial population of cats and disperse them into the M-dimensional solution space ($X_{i,d}$) and randomly assign each cat a velocity in range of the maximum velocity value ($t_{i,d}$).
2. According to the value of MR, assign each cat a flag to sort them into the seeking or tracing mode process.
3. Evaluate the fitness value of each cat and save the cat with the best fitness function. The position of the best cat (X_{best}) represents the best solution so far.
4. Based on their flags, apply the cats into the seeking or tracing mode process as described below.
5. If the termination criteria are satisfied, terminate the process. Otherwise repeat steps 2 through 5.

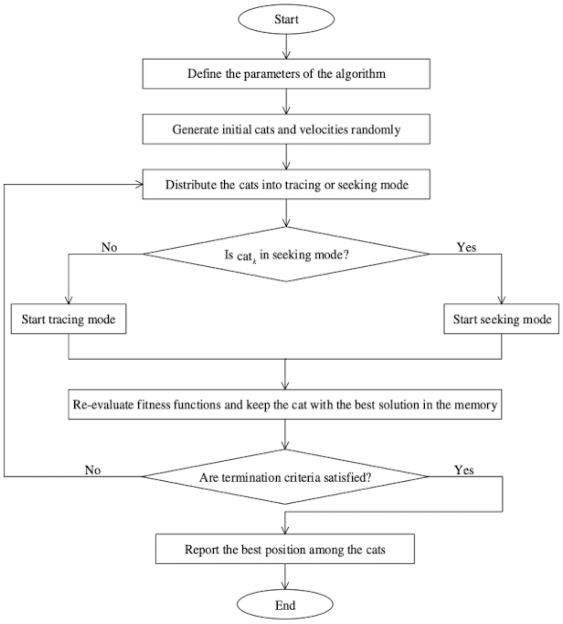


Figure 1 - Flow chart of CSO algorithm

For implementation of this algorithm, we have taken 100 cat agents and the algorithm has been iterated for 100 generations. Here, our length of solution space is 1633 which are the number of trainable parameters. A fitness function is provided which basically calculates the Mean Squared Error (MSE) using the solution generated by the cat agents per iteration. Throughout all the iterations, the MSE is calculated and the solution with the lowest MSE is considered as the best solution at the end of all the iterations. The seeking memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self-position considering (SPC) parameters of the algorithm handle the seeking process of the cat. During the trace mode, a new velocity

vector is calculated for each cat and a new position is assigned to the cat.

The next algorithm used for analysis was Cuckoo Optimization Algorithm (COA). The algorithm was proposed by Rajabioun in 2011 and was inspired by cuckoo bird's lifestyle, egg laying features, and breeding technique. This as other evolutionary approaches start off with an initial population and this population consists of mature cuckoos and eggs. Some variations of this algorithm include modified COA which was developed by Kahramanli in 2012, and penalty function COA by Mellal and Williams in 2015.

Cuckoo being a brood parasite i.e. birds that have detached themselves from the challenge of nest making for survival and evading predators, lay their eggs in other species nests and hence uses a cunning way to raise their families. Some of the birds recognize cuckoos' eggs and either they throw the eggs out of the nest or completely leave the nest and build a new one. Hence, cuckoos continuously improve their mimicry from the eggs in the target nests and host birds learn new ways to recognize the strange eggs as well. This struggle for survival among different birds and cuckoos is a constant and continuous process.

The following are the characteristics of COA algorithm,

Decision variable	Cuckoo habitat
Solution	Habitat
Old solution	Old habitat
New solution	New habitat
Best solution	Habitat with best rate of life
Fitness function	Distance between best habitat and recent habitat
Initial solution	Random eggs for all cuckoos
Selection	-
Process of generating new solution	Emigration cuckoos toward best area

COA starts with an initial population (population of cuckoos). These cuckoos have got some eggs that will be laid in other species' nests. Some of these eggs that look like the host's eggs are more probable to be raised and turned into cuckoos. Other eggs are detected by the host and are demised. The rate of the raised eggs shows the suitability of the area. If there are more eggs to be survived in an area, there is more profits to that area. Thus, the situation in which more eggs are survived will be a parameter for the cuckoos to be optimized.

Cuckoos search for the best area to maximize their eggs' life lengths. After hatching and turning into mature cuckoos, they form societies and communities. Each community has its habitat to live. The best habitat of all communities will be the next destination for cuckoos in other groups. All groups immigrate to the best current existing area. Each group will be the resident in an area near the best current existing area. An egg laying radius (ELR) will be calculated regarding the number of eggs each cuckoo lay and its distance from the current optimized area.

Afterwards, cuckoos start laying eggs randomly in the nests within their egg laying radii. This process continues until reaching the best place for egg laying (a zone with the most profit). This optimized zone is the place in which the maximum number of cuckoos gathers together.

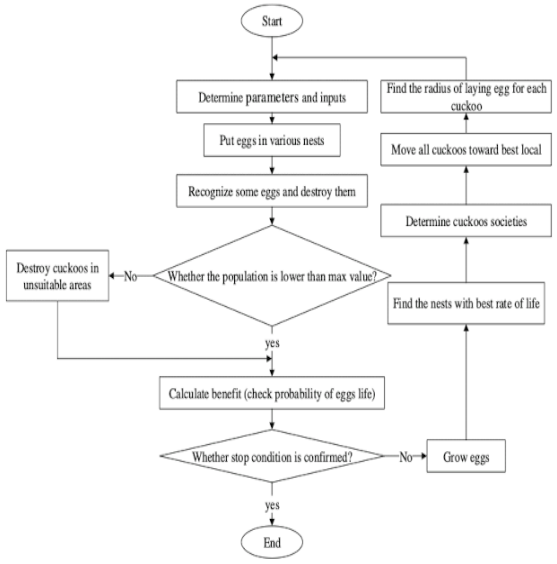


Figure 2- Flowchart of COA algorithm

For implementation of this algorithm, we have taken 100 agents and the algorithm has been iterated for 100 generations. Here, our length of solution space is 1633 which are the number of trainable parameters. A fitness function is provided which basically calculates the Mean Squared Error (MSE) using the solution generated by the agents per iteration. Throughout all the iterations, the MSE is calculated and the solution with the lowest MSE is considered as the best solution at the end of all the iterations. In this algorithm, we have provided the number of nests to be 100. Initially, population of 100 nests has been generated. Each cuckoo agent is pointed to a random nest by performing Levy lights. A nest is chosen randomly, and fitness of that nest is calculated with the previous nest. This is iterated for 100 generations and remove the nests with probability less than a user defined threshold value and generate same number of deleted nests again.

The next algorithm used was Grey Wolf Optimization algorithm (GWO). This algorithm was introduced by Mirjalili et al. in 2014. This algorithm is inspired by the social hierarchy and the hunting method of grey wolves. These wolves usually live in a group of five to twelve individuals and follow a strict social hierarchy. The leaders of the pack are male and female wolves that are often responsible for making decision for their pack. The next level in hierarchy is beta wolves and their role are to help alpha's make decision and disciplining the pack. The weakest level in hierarchy is omega and these play roles of a scapegoat. The remaining wolves are delta. These obey alpha and beta wolves and dominate omega wolves.

According to Muro et al. (2011) grey wolves' hunting includes the following three main parts:

(1) Tracking, chasing, and approaching the prey.
(2) Pursuing, encircling, and harassing the prey till it stops moving.
(3) Attacking the prey.

These two social behaviors of grey wolves' pack (social hierarchy and hunting technique) are modeled in the GWO algorithm.

In the GWO algorithm, the best solution is considered as alpha. Therefore, the second and third best solutions are respectively considered as beta and delta, and other solution is assumed to be omega. In the GWO algorithm, hunting (optimization) is guided by alpha, beta, and delta wolves, and omega wolves follow them.

The following are the characteristics of GWO algorithm,

Decision variable	Grey wolf
Solution	Position of grey wolf
Old solution	Old position of grey wolf

New solution	New position of grey wolf
Best solution	Position of prey
Fitness function	Distance between grey wolf and prey
Initial solution	Initial random position of grey wolf
Selection	-
Process of generating new solution	Hunting operators i.e. encircling and attacking prey

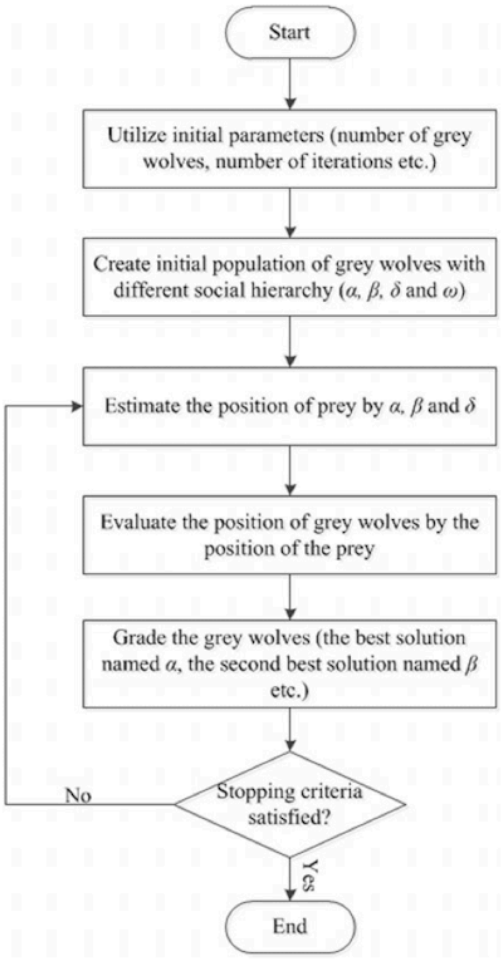


Figure 3 - Flowchart of GWO algorithm

For implementation of this algorithm, we have taken 100 agents and the algorithm has been iterated for 100 generations. Here, our length of solution space is 1633 which are the number of trainable parameters. A fitness function is provided which basically calculates the Mean Squared Error (MSE) using the solution generated by the agents per iteration. Throughout all the iterations, the MSE is calculated and the solution with the lowest MSE is considered as the best solution at the end of all the iterations. The flowchart depicts the entire flow of the algorithm once these initial parameters are set.

The next algorithm used was Firefly algorithm. This algorithm was introduced by Yang in 2008 and is inspired by the flashing power of fireflies. Some variants of firefly algorithm are, levy flight FA was developed by Yang in 2010 and Yan et al. developed adaptive FA in 2012. Firefly algorithm has been applied to various problems due to its high convergence speed.

Different species of fireflies exhibits unique flashing patterns. The flashing pattern is generated by bioluminescence and it is believed that these flashes are used to attract mating partners or to attract potential preys or to evade predators. The factors that varies flashes is the rate of flashing, rhythmic flashes and the duration of flashing. They produce attractiveness by shining light.

The FA assumes that the flashing light can be formulated in such a way that it is associated with the objective function of the optimization problem. The FA is based on three idealized rules:

1) All fireflies are unisex, so their attractiveness depends on the amount of light flashed by them regardless of their sex.
2) The attractiveness of fireflies is proportional to their brightness. Thus, for any two flashing fireflies, the firefly that flashes less will move toward the firefly that flashes more. The attractiveness and the brightness of fireflies decrease as the distance between fireflies increases. Thus, the movement of

fireflies continues in this manner until there is no brighter firefly in a group. Once this happens the fireflies move randomly.
3) The brightness of a firefly is determined by a fitness function.

The FA designates a firefly as a solution whose location in any N-dimensional
is a decision variable. In nature each firefly moves toward other fireflies according to their attractiveness. For simplicity, it is assumed that the attractiveness of a firefly is determined by its brightness, which in turn is associated with the fitness function. The FA dictates that if the fitness value of a firefly is larger than that of another firefly, the firefly with less brightness (fitness value) moves toward the firefly with more brightness. The movement of the firefly is based on the light intensity of the other firefly, which is influenced by the distance between the fireflies. New positions occupied by the fireflies are new solutions.

Decision variable	Position of firefly in each dimension
Solution	Firefly(position)
Old solution	Old position of firefly
New solution	New position of firefly
Best solution	
Fitness function	Brightness
Initial solution	Random firefly
Selection	-
Process of generating new solution	Movement of firefly

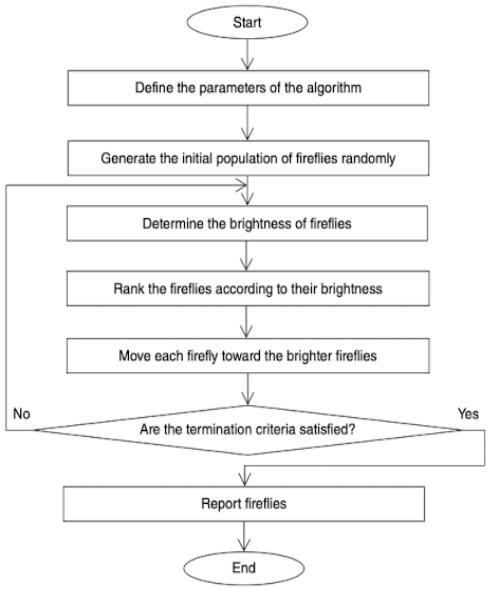


Figure 4 - Flowchart of FA algorithm

For implementation of this algorithm, we have taken 100 agents and the algorithm has been iterated for 100 generations. Here, our length of solution space is 1633 which are the number of trainable parameters. A fitness function is provided which basically calculates the Mean Squared Error (MSE) using the solution generated by the agents per iteration. Throughout all the iterations, the MSE is calculated and the solution with the lowest MSE is considered as the best solution at the end of all the iterations. The flowchart depicts the entire flow of the algorithm once these initial parameters are set.

The last algorithm was the Bat Algorithm. This algorithm was developed by Yang in 2010 and is based on the echolocation feature of microbats. Echolocation is a technique used by bats with the help of which they are able to determine their location while flying by sound emissions and reception. This enables bats to detect prey, avoid obstacles and locate their roosting crevices in the dark. Bats emit sound pulses and each pulse has a constant frequency and lasts a few thousandths of a second. About 10–20 sounds are emitted every second with the rate of

emission up to about 200pps (pulses per second) when they fly near their prey while hunting. The wavelengths emitted are of the same order of magnitude as the prey sizes. The loudness of the pulse that a bat emits from the loudest when searching for prey to a quieter base when homing towards the prey. The travelling range of such short pulses is typically a few meters.

Such echolocation behavior of microbats has been formulated to create the bat-inspired optimization algorithm applying the following idealized rules (Yang, 2010):

1) All bats use echolocation to sense distance, and they can discern the difference between food/prey and background barriers.

2) Bats fly randomly with velocity v_i at position x_i with a fixed frequency λ_{min}, varying wavelength W, and loudness A_0 to search prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the pulsation rate, depending on the proximity of their target.

3) The loudness can vary from a large (positive) A_0 to a minimum constant value A_{min}.

In general the frequency (λ) is in the range of [λ_{min}, λ_{max}] and corresponds to a range of wavelengths [W_{min}, W_{max}]. For simplicity, λ is assumed to be in the range of [0, λ_{max}]. The pulsation rate (δ) is in the range of [0,1], where 0 means no pulses at all and 1 means the maximum pulsation rate.

Decision variable	Position of bat in any dimension
Solution	Position of bat
Old solution	Old position of bat
New solution	New position of bat
Best solution	Best bat
Fitness function	Distance with food
Initial solution	Random bat
Selection	Loudness criteria
Process of generating new solution	Fly bats

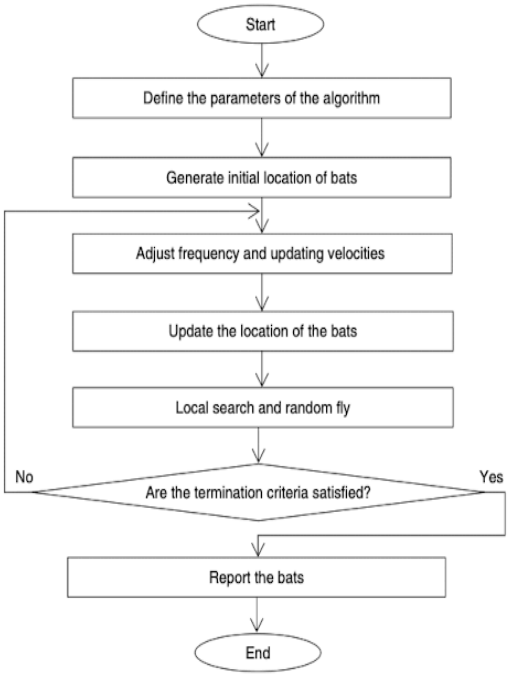


Figure 5 - Flowchart of BA

The initilization details are similar to all the previous algorithms. The flowchart above depicts the entire flow of the algorithm once these initial parameters are set.

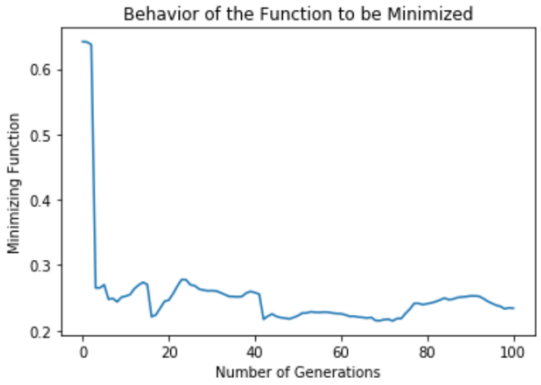
IV. Results

The following results were obtained when all the algorithms were executed

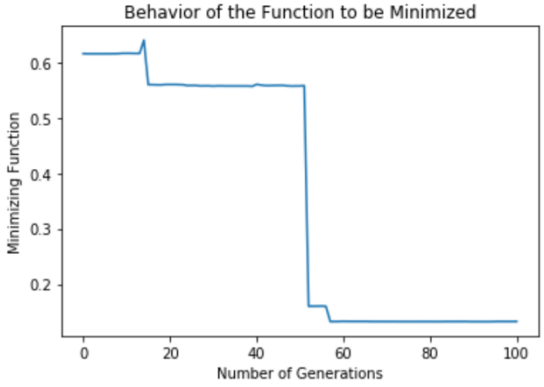
Algorithm	**Accuracy**	**MSE**
Bat Algorithm	80.4%	0.19
Cuckoo Search Optimization	88.6%	0.11
Cat Algorithm	86.4%	0.13
Gray Wolf Optimization	93.2%	0.06
Firefly Algorithm	80.5%	0.19
Particle Swarm Optimization	95%	0.05
Backpropagation	99%	0.005

The behavior of the minimization function with respect to the generations is given below –

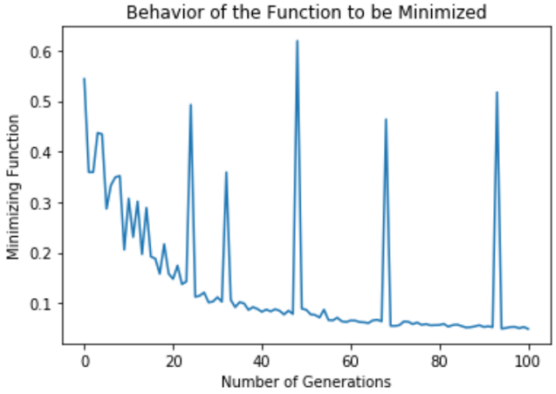
1. Bat Algorithm



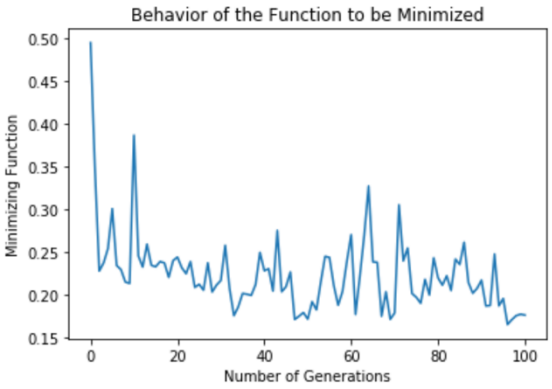
2. Cat Swarm Optimization



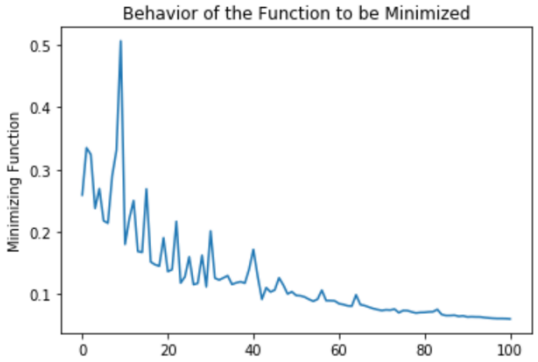
3. Cuckoo Search Optimization



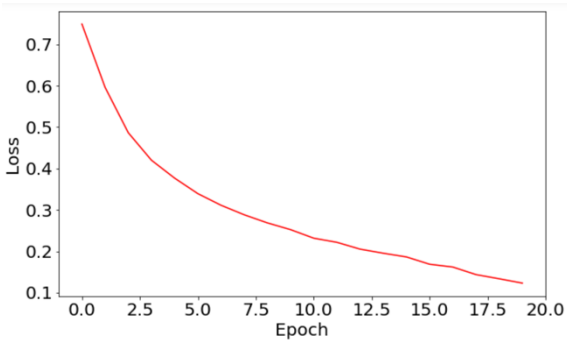
4. Firefly Algorithm



5. Gray Wolf Optimization



6. Back Propagation



V. References

[1] (Wiley series in operations research and management science) Bozorg-Haddad, Omid_ Loaiciga, Hugo A._ Solgi, Mohammad - Meta-heuristic and evolutionary algorithms for engineering optimization-John Wile

[2] [Elsevier Insights] Xin-She Yang (Auth.) - Nature-Inspired Optimization Algorithms (2014, Elsevier)

[3] Omid Bozorg-Haddad - Advanced Optimization by Nature-inspired Algorithms-Springer (2017)

[4] Xin-She Yang (ed.) - Nature-inspired Algorithms and Applied Optimization-Springer (2018)

[5]https://github.com/SISDevelop/SwarmPackagePy