INTERNSHIP REPORT

A report submitted

by

Apurva Sarode

C22018771847

Yukta Saindane

C22018771844

Mrunmayee Pathak

C22018771838

Under the Supervision of Prof.Kalpana Joshi

INDUSTRY NAME

(Duration: July 2021 to Aug 2021)

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

MKSSS CUMMINS COLLEGE OF ENGINEERING FOR WOMEN, PUNE

(An Autonomous Institute)

AY: 2020-21

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

MKSSS CUMMINS COLLEGE OF ENGINEERING FOR WOMEN, PUNE

(An Autonomous Institute)

CERTIFICATE

This is to certify that the "**Internship report**" submitted by (Apurva Sarode, Yukta Saindane, Mrunmayee Pathak) (C.Num.: C22018771847, C22018771844, C22018771838) is work done by her at (Cummins College of Engineering, Pune), and submitted during 2020 – 2021 academic year.

Internship Mentor Internship Coordinator Head of the Department

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all the team members of this project for their valuable guidance and support throughout the internship period.

I would also like to thank our principal Mrs.Madhuri Khambete, HOD DR.Prachi Mukharji, and our mentor Prof.Kalpana Joshi for giving me this great opportunity to do this internship.

T.Y INTERNSHIP WORK PLAN
AY 2020-21

Internship Mentor: Mrs. Kalpana Joshi

Students: 1. Apurva Sarode
2. Yukta Saindane
3. Mrunmayee Pathak

Internship Description: This internship gives a brief overview of Swarm Intelligence and the Cukoo Search Algorithm where we learned diagnostic evaluation of the effects of search population and the number of iterations in solving the benchmark functions and Levy functions using the CS.

Problem statement: Implementation analysis of the benchmark and Levy test functions using the Cuckoo Search with emphasis on the effect of the search population and iterations count in the algorithm's search processes.

Week/Date	Problem/work Description

Week One-	Research on Swarm Intelligence algorithms.
Week Two	Research on Cuckoo Search Algorithm
Week Three	Search for the different benchmark functions and understand the algorithm by searching for the parameters to be initialized in MATLAB code.
Week Four	Implementation of algorithm and start coding it on MATLAB.
Week Five	Testing of code and error correction.
Week six	Tuning of algorithm parameter and their effect on convergence.
Week seventh	Plotting the convergence plot for the unimodal equation
Week Eight	Final Presentation
Expected Outcome	Understanding swarm intelligence and study of Cuckoo Search Algorithm using Benchmark Functions.
Any PO [Program Outcome] Addressed	It has been demonstrated that CS can be far more effective than most existing metaheuristic algorithms, including particle swarm optimization.

Faculty Sign: Dr.Prachi Mukherji

HoD,

ETC

Abstract

This project presented the diagnostic evaluation of the effects of search population and the number of iterations in solving the benchmark functions and Levy functions using the CS. After a number of experimental procedures, it was discovered that in solving the benchmark function as well as the Levy function, the CS obtained better results when a population of 20 nests was deployed to the search space. By using 20 nests, near-optimal solutions were obtainable from as low as 100 to 200 iterations. In fact, at 100 iterations,5 dimensions and +1/-1 lower bound the CS obtained the best result in every run.

Surprisingly, however, this study discovered that increasing the iteration, dimension and lower and upper bound did not improve the solution. It rather worsened the result and increased the processing time. Similarly, for the Levy function, the near-optimal solution was obtainable at 100 iterations and with a search population of 20 nests. The use of more iterations-cum-nests was discovered to only increase the processing overhead without the expected improvement in results

Keyword:- Cuckoo search, iteration, Levy function, population, Benchmark function

Index

Introduction:

The solutions to multitudinous domains-whether in engineering design, operational research, industrial process, or economics inevitably have optimization at heart. However, having a solid grasp for such problems turns out to be painstakingly tough and tedious; and thus, gearing towards an efficient and effective algorithm in the light of solving increasingly complex optimization problems in practice is of paramount significance. Extensive and intensive studies in this aspect fruit in numerous optimization techniques, particularly, the bio-inspired metaheuristic methods which draw inspiration from the means on how humans and living creatures struggle to survive in a challenging environment, for instance, genetic algorithm (GA), particle swarm optimization (PSO), differential evolution (DE), ant colony optimization, artificial bee colony algorithm, and firefly algorithm, form the hot topics in this area.

Cuckoo search is one of many nature-inspired algorithms used extensively to solve optimization problems in different fields of engineering. It is very effective in solving global optimization because it is able to maintain a balance between local and global random walks using switching parameters. The switching parameter for the original Cuckoo search algorithm is fixed at 25% and not enough studies have been done to assess the impact of dynamic switching parameters on the performance of the Cuckoo search algorithm. This project introduces the implementation of ten mathematical unimodal and multimodal benchmark functions and Levy test functions using the Cuckoo Search with emphasis on the effect of the dimensions and iterations count in the algorithm's search processes.

Literature Review:

1. An adaptive Cuckoo search algorithm for optimization.

Author: M. Mareli a, B. Twala b

Abstract:

Cuckoo search is one of many nature-inspired algorithms used extensively to solve optimization problems in different fields of engineering. It is very effective in solving global optimization because it is able to maintain a balance between local and global random walks using switching parameters.

The switching parameter for the original Cuckoo search algorithm is fixed at 25% and not enough studies have been done to assess the impact of dynamic switching parameters on the performance of the Cuckoo search algorithm. This paper's contribution is the development of three new Cuckoo search algorithms based on dynamically increasing switching parameters. The three new Cuckoo search algorithms are validated on ten mathematical test functions and their results compared to those of Cuckoo search algorithms with constant and dynamically decreasing switching parameters respectively.

2. Optimization of process parameters in submerged arc welding using a cuckoo algorithm.

Author: P.V. Gopalkrishna, K. Kishore, G. Sravani

Abstract:

Optimization carried one such powerful tool is Cuckoo Optimization Algorithm (COA), which is a computational method that optimizes a problem by iteratively trying to improve a candidate solution regarding a given measure of quality.A combined objective function is also developed which can be used to obtain the common parameter setting which satisfies all the objectives simultaneously. The models developed for various output responses will be very useful for predicting the responses for the given sets of input variables of the submerged arc welding process.

3. Cuckoo Search Algorithm-based Dynamic Parameter Adjustment Mechanism for Solving Global Optimization Problem.

Author: Waleed Alomoush.

Abstract:

Cuckoo search algorithm CSA is a recent optimization algorithm of swarm intelligence, which has demonstrated powerful outcomes on many optimization issues. Nevertheless, it has some limitations such as being stuck in local optima and premature convergence especially when solving complicated problems of optimization. Also, the CSA parameters are static during generations time which leads to stuck in local optima and couldn't find the best solutions.

In this paper, we proposed an improved standard cuckoo search algorithm-based dynamic parameter adjustment mechanism called (CSDPA). The CSDPA presents two equations to update the values of the parameters of step size and discovery probability during the search process. The experiments are tested on ten conventional benchmark functions. Outcomes demonstrate the new CSDPA approach is outperformed by the CSA and other CSA variants.

Specifications:

Animals search for food in a random or quasi-random manner in nature. The foraging path of an animal is effectively a random walk because the next move is based on both the current location/state and the transition probability to the next location. The chosen direction's probability is modeled mathematically. Various studies have shown that the flight behavior of many animals and Insects demonstrates the typical characteristics of Lévy flights. A Lévy flight is a random walk in which the step lengths are calculated according to a heavy-tailed probability distribution. The distance from the origin of the random walk tends to a stable distribution. The cuckoo lays an egg at the random location via Levy flight, ´ which is characterized by,

$$xi = (iter + 1) = xi\ (iter) + \alpha \times \text{levy}'\ (\lambda),$$

$$\text{levy}'\ (\lambda) = \left| \frac{\Gamma\ (1+\lambda) \times \sin\ (\pi\lambda/2)}{\Gamma\ ((1+\lambda)\ /2) \times \lambda \times 2\text{^}((\lambda-1)/2)} \right| \text{^}\ (1/\lambda)$$

where *xi* is the possible solution, iter denotes the current generation number.

The Levy flight process is a random walk that forms a series of instantaneous jumps chosen from a heavy-tailed probability density function.

The step size α, which controls the scale of these random search patterns, helps exploit the search space around the current best solution and if the value for α is too big, the new cuckoo egg might be placed outside the bounds. To balance the effectiveness for both intensification and diversification, Yang and Deb assigned the value of α as **1**.

In the CSA, only the probability of the abandoned nests *pa* is tuned. However, the setting of *pa* = 0.25 is sufficient enough, as they found out that the convergence rate of CSA is insensitive to *pa*. Thus, the fraction of nests to desert *pa* is assigned as 0.25 in this study

Parameter	Description
D	Dimensions

Lb	Lower Bound
Ub	Upper Bound
N	Population Size
Pa	Discovery Rate
max_iter	Maximum Iterations
sigma	Positive step size scaling factor

Methodology:

Cuckoo search algorithm (CSA), another adoption of biomimicry in the optimization technique which reproduces the breeding strategy of the best-known brood parasitic bird, the cuckoos, has been proposed by Yang and Deb recently. Cuckoos, probably one of the most vicious and cunning species of all bird breeds, clandestinely lay their eggs in the nests of other host birds, sparing themselves the parental responsibilities of raising the young. In fact, cuckoos practice the art of deception all the time in their reproductive life.

They mimic the color and pattern of the host eggshell in order to disguise their eggs from being detected by the host birds. To make more space and food for their young chick, cuckoos will steal the host egg while sneaking their own into the nest. However, the relationship between the host species and the cuckoos is often a continuous arms race. The hosts learn to discern the imposters and they either throw out the parasitic eggs or desert the nest; the parasites improve the forgery skill to make their eggs appear more alike with the host eggs.

The CSA, which draws inspiration from cuckoo's adaption to breeding and reproduction, is idealized with the assumptions as follows:

- Each cuckoo lays one egg in a randomly selected host nest at a time, where the egg represents the possible solution for the problem under study.
- The CSA follows the survival of the fittest principle. Only the fittest among all the host nests with high-quality eggs will be passed on to the next generation;
- The number of host nests in the CSA is fixed beforehand. The host bird spots the intruder egg with a probability pa ϵ [0,1].. For such incidents, the host bird will either evict the parasitic egg or abandon the nest totally and seek for a new site to rebuild the nest.

Derived from these assumptions, the steps involved in the computation of the standard CSA are presented in the Algorithm**:**

Objective function f(x), x = (x1, x2 …….xd)^T

Generate initial population of n host nests xi(i = 1, 2...n)

While (t < Max Generation) or (stop criteria)

Get a cuckoo (say i) randomly by Lévy distribution;

Evaluate its quality/fitness Fi;

Choose a nest among n (say j) randomly;

Evaluate its quality/fitness Fj;

If (Fi >Fj)

Replace j with the new solution;

End

A fraction of (Pa) of worse nests are abandoned and new ones are built at new locations via Lévy flights;

Keep the best solutions (or nests with quality solutions);

Rank the solutions and find the current best;

End while

Post-processing

In this algorithm (the CS), the eggs of the host bird in any given nest represent an optimization solution, while the strange eggs of the cuckoo birds represent new solutions. Through careful manipulation of the cuckoo eggs and those of the host birds, the CS is able to arrive at good optimization solutions to complex optimization problems

Detailed design:

Since the focus of the first part of this paper was to determine the effect of the search population-cum-number of iterations required to obtain the best output to the Benchmark Function and the second part was to examine the same in Levy test functions (and by implication, other similar problems). The population of nests was 20. Also, the number of iterations included 100, 200, 100. The CS parameters used for the experiments were

u=rand (size (s)) * sigma; v= rand (size(s));

pa=0.5;

step = u./abs (v). ^ (1/beta);

step size =0.01* step.

Each experiment test case was executed 20 times.

A Lévy flight is a random walk in which the step lengths are calculated according to a heavy-tailed probability distribution.

The distance from the origin of the random walk tends to a stable distribution after a large Levy flight mechanism

$$Xi \,\hat{}\,(t+1) = Xi \,\hat{}\,|(t)| + \alpha \oplus \text{Lévy}(\lambda)(1)$$

Where, $\alpha > 0$ is the step size which should be related to the scales of the problem of interest cases. The above equation is essentially the stochastic equation for a random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the above equation) and the transition probability (the second term) & the product means entry-wise multiplications.

The authors defined the CS algorithm by setting three rules that idealize the behavior of cuckoos in order to become appropriate for implementation as a computer algorithm:

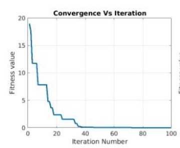
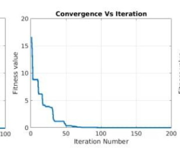
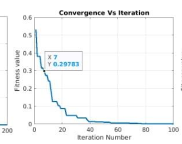
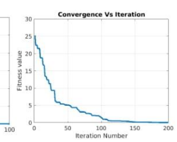
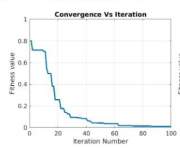
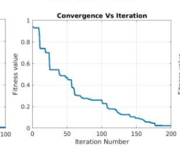
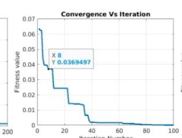
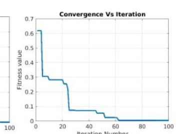
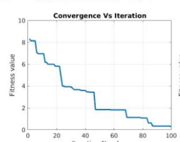
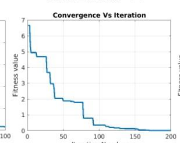
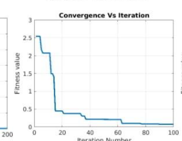
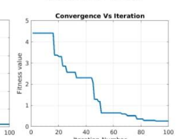
Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest. The best nests with high-quality eggs will be carried over to the next generations.

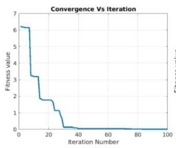
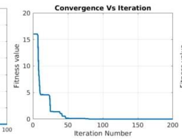
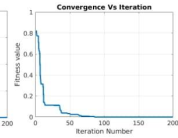
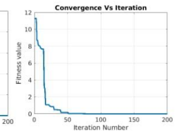
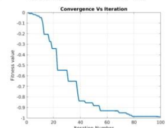
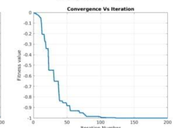
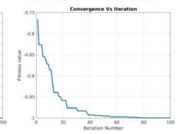
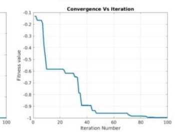
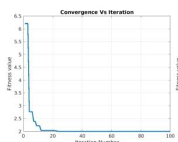
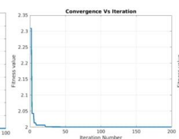
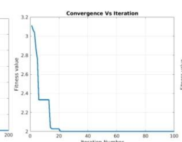
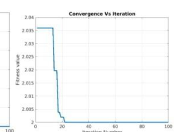
The number of available host nests is fixed and the egg laid by a cuckoo may be discovered by the host bird with a probability pa $\in (0, 1)$.

In this case, the host bird can either get rid of the egg or simply abandon the nest and build a completely new nest. For simplicity, the last assumption can be approximated by a fraction pa of the n nests being replaced by new nests, having new random solutions. For a maximization problem, the quality or fitness of a solution can simply be proportional to the objective function. When generating new solutions x(t+1) for, say, a cuckoo is a Lévy flight is performed

Result:

The proposed algorithm was tested on 10 standard benchmark functions that are commonly used to evaluate the performance of optimization algorithms. The functions selected are carrying different properties of being unimodal or multimodal, separable or non-separable, and having few or many local optima solutions. For each test function, the initial populations of 20 host nests are generated randomly. The simulations are performed for 20 independent runs and then average and standard deviation are calculated to find the best solutions which all are compared with varying dimensions, maximum iteration values, and lower and upper bounds which showed in the following tables.

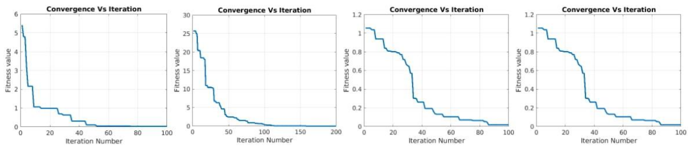
Functions					
			D=5		
Unimodal		**Max_iter=100**	**Max_iter=200**	**lb/ub=-1/+1**	**lb/ub=-3/+3**
Sphere	**Fmin**	0.0004	3.90E-05	0.00094886	0.0266
	Mean	0.0005	5.00E-05	6.12E-04	0.01152584053
	Std Deviation	0.0001	0.000008	0.000238	0.0266
	waveform				
Griewank	**Fmin**	0.011	0.0127	1.02E-04	0.0027
	Mean	0.0209	0.3973	7.47E-05	0.0036
	Std Deviation	0.014000714	0.54390653	0.00003898	0.0012727
	waveform				
Ackely	**Fmin**	0.334	0.0234	0.0777	0.2595
	Mean	0.36385	0.0606	0.1056	0.38105
	Std Deviation	0.04221427484	0.052608744	0.039456558	0.171897658
	waveform				
Brown	**Fmin**	1.77E-02	2.51E-04	9.18E-06	8.03E-05
	Mean	-4.89E-01	1.37E-04	4.91E-06	9.01E-05

	Std deviation	0.00565685424	1.61E-04	6.04E-06	0.000013803431
	Waveform				
Exponential	**Fmin**	-9.88E-01	-1.00E+00	-9.81E-01	-9.95E-01
	Mean	-9.76E+00	-1.00E+00	-9.94E-01	-9.95E-01
	Std deviation	0.01725340546	0.00E+00	0.007566042559	0.00E+00
	waveform				
Multimodal Happycat	**Fmin**	2	2	2	2
	Mean	2	2	2	2
	Std deviation	0	0	0	0
	Waveform				
Alpine	**Fmin**	11.9423	6.207	0.2054	4.7835
	Mean	11.9491	5.5249	0.1564	0.3256
	Std deviation	0.4545	0.6821	0.049	5.1101

Functions

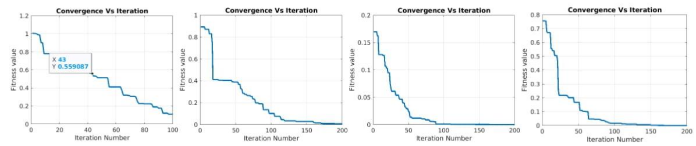
		D=10			
Unimodal		**Max_iter=100**	**Max_iter=200**	**lb/ub=-1/+1**	**lb/ub=-3/+3**
Sphere	**Fmin**	0.4232	0.0039	0.0181	0.0987
	Mean	1.1337	0.0137	0.0184	0.1936
	Std Deviation	0.502399368	0.0006	0.0002121	0.067104433

waveform



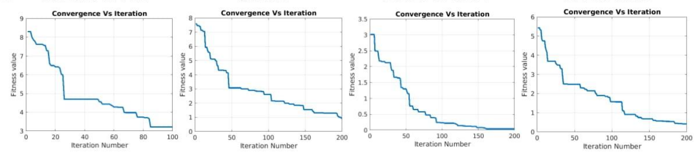
Griewank	**Fmin**	0.1621	0.0073	4.44E-05	3.02E-04
	Mean	0.1368	0.01075	2.73E-05	1.95E-03
	Std Deviation	0.03577960313	0.004879036	0.000024214	0.002331861

waveform

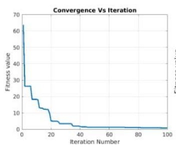
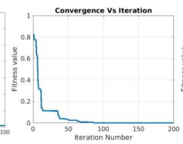
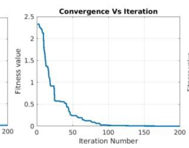
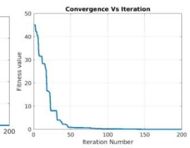
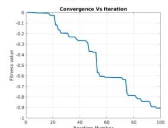
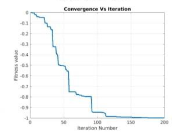
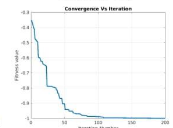
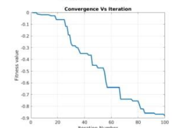
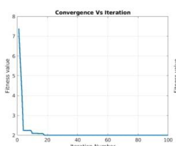
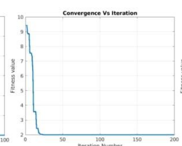
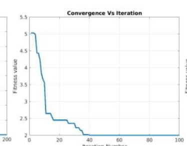
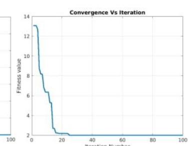


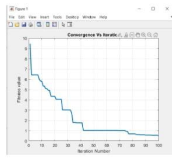
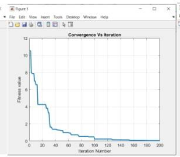
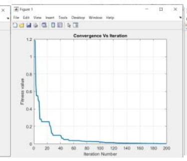
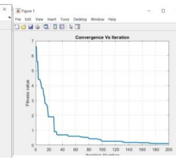
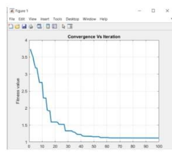
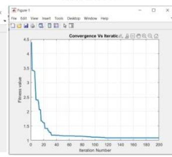
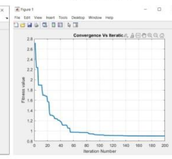
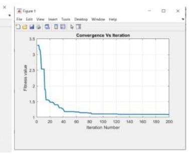
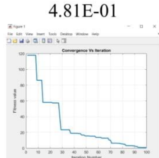
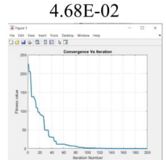
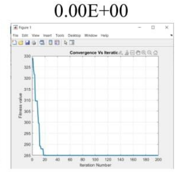
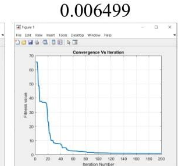
Ackely	**Fmin**	3.2114	0.9348	0.0406	0.4128
	Mean	3.1262	0.036062445	0.0427	0.57395
	Std Deviation	0.1204909955	0.9093	0.0029698484	0.22790051

waveform

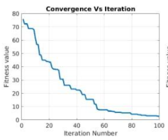
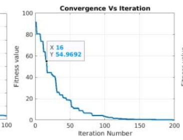
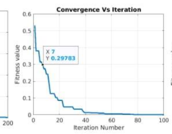
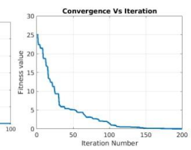
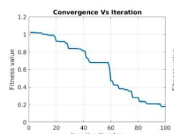
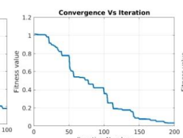
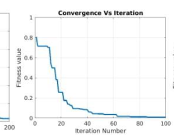
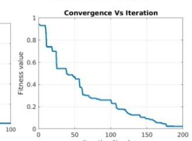
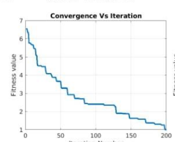
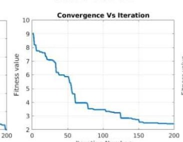
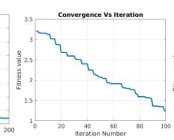
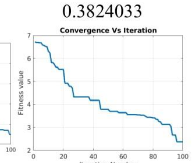


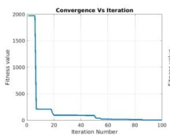
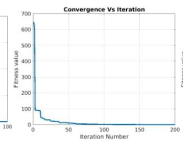
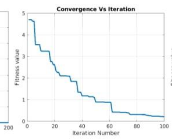
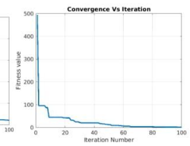
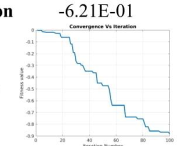
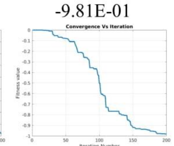
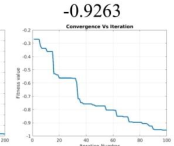
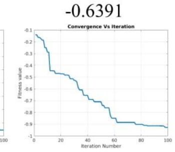
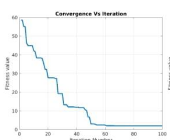
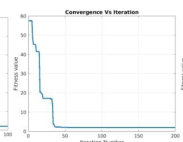
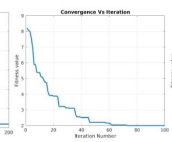
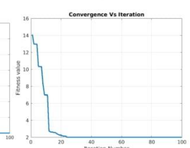
Brown	**Fmin**	1.93E+00	1.17E-02	4.70E-04	2.20E-03
	Mean	1.93E+00	1.17E-02	4.70E-04	6.79E-02

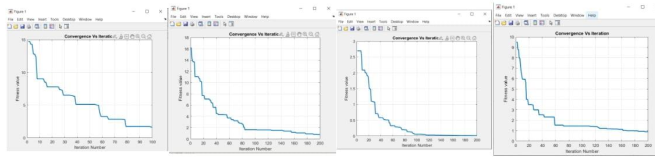
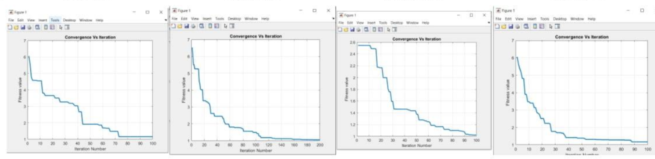
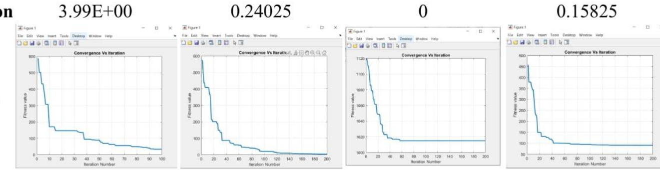
	Std deviation	1.43E+00	7.27E-02	8.37E-04	7.80E-03
	Waveform				
Exponential	**Fmin**	-9.06E-01	-9.99E-01	-0.9996	-8.67E-02
	Mean	-9.06E-01	-9.99E-01	-0.9996	-8.67E-01
	Std deviation	-8.94E-01	-9.97E-01	-1.00E+00	-8.93E-01
	waveform				
Multimodal Happycat	**Fmin**	2	2	2	2
	Mean	2	2	2	2
	Std deviation	0	0	0	0
	Waveform				
Alpine	**Fmin**	0.3556	0.0547	0.0024	0.1127
	Mean	0.44645	0.0587	0.002	0.0858
	Std deviation	0.09085	0.004	0.0004	0.0269

	Waveform				
Periodic	**Fmin**	1.1178	1.0884	0.9005	1.0902
	Mean	1.1188	1.0895	0.90055	1.0956
	Std deviation	0.001	0.00064	4.99E-05	0.0054
	Waveform				
Qing	**Fmin**	7.64E-01	1.29E-01	2.85E+02	1.0175
	Mean	1.24E+00	1.75E-01	2.85E+02	1.01815
	Std deviation	4.81E-01	4.68E-02	0.00E+00	0.006499
	Waveform				

Functions

			D=15		
Unimodal		**Max_iter=100**	**Max_iter=200**	**lb/ub=-1/+1**	**lb/ub=-3/+3**
Sphere	**Fmin**	3.4786	3.4786	0.095	0.0382
	Mean	3.14015	0.185	0.1085	0.061
	Std Deviation	0.47864	0.020506096	0.019091883	0.032314779
	waveform				
Griewank	**Fmin**	0.1779	0.0338	4.17E-04	0.9009
	Mean	0.26775	0.05405	4.03E-04	0.923
	Std Deviation	0.12706708	0.0286378	0.000019014	0.031254119
	waveform				
Ackely	**Fmin**	2.8292	2.4308	1.2455	2.9146
	Mean	2.607	2.317	1.1337	2.6442
	Std Deviation	0.31423825	0.1609375	0.158109	0.3824033
	waveform				
Brown	**Fmin**	4.14E+00	6.81E-01	0.2186	0.8974
	Mean	4.41E+00	6.81E-01	0.2186	0.8974

	Std deviation	7.32E+00	6.11E-01	0.323	2.3196
	Waveform				
Exponential	**Fmin**	-4.36E-01	-9.68E-01	-0.9553	-0.9623
	Mean	-4.36E-01	-9.68E-01	-0.9553	-0.9263
	Std deviation	-6.21E-01	-9.81E-01	-0.9263	-0.6391
	waveform				
Multimodal Happycat	**Fmin**	2.0001	2	2	2
	Mean	#REF!	2	2	2
	Std deviation	#REF!	0	0	0
	Waveform				
Alpine	**Fmin**	1.638	0.7754	0.0112	0.8134
	Mean	213.655	0.5497	0.0118	0.04215
	Std deviation	300.057	0.2257	0.0006	0.85555

	Waveform				
Periodic	**Fmin**	1.1552	1.0517	1.0217	1.1637
	Mean	1.218	1.0376	1.02605	1.196
	Std deviation	0.0628	0.0141	0.00435	0.0323
	Waveform				
Qing	**Fmin**	3.36E+01	3.49E+00	1015	91.163
	Mean	2.96E+01	3.25E+00	1015	91.32125
	Std deviation	3.99E+00	0.24025	0	0.15825
	Waveform				

Here, we have tested those algorithms with 100, 200, independent iterations, on each of the test functions. The Dimension for each function for each algorithm is set for D=5, 10, 15 with 5,3 and 1 upper and lower bounds. Along with this, we use Pa, α, λ to improve the result for CSA globally and locally. Here in CSA, we used the Discovery rate of alien eggs or solutions, Pa=0.25. We have found the Fmin, mean, and Standard deviation values for the algorithm.

Conclusion:

This project presented the diagnostic evaluation of the effects of search population and the number of iterations in solving the benchmark functions and Levy functions using the CS. After a number of experimental procedures, it was discovered that in solving the benchmark function as well as the benchmark Levy function, the CS obtained better results when a population of 20 nests was deployed to the search space. By using 20 nests, near-optimal solutions were obtainable from as low as 100 to 200 iterations. In fact, at 100 iterations,5 dimensions and +1/-1 lower bound the CS obtained the best result in every run.

Surprisingly, however, this study discovered that increasing the iteration, dimension and lower and upper bound did not improve the solution. It rather worsened the result and increased the processing time. Similarly, for the Levy function, the near-optimal solution was obtainable at 100 iterations and with a search population of 20 nests. The use of more iterations-cum-nests was discovered to only increase the processing overhead without the expected improvement in results

In the light of the findings of this project, it is recommended that when solving any benchmark function and Levy functions or any similar problems, using a population of 20 nests will give an optimum or near-optimum solution. In terms of the required number of iterations, this study recommends from 100 to 200 depending on the primary considerations. Conversely, for the Levy function, the use of 100 iterations is sufficient to give near-optimal solutions. Before this study, it was believed by some researchers that the more populations deployed to the search space, the more likely it was to obtain optimal results. This study proved otherwise. Based on this finding, it is recommended that further implementation evaluations be carried out on the different benchmark and Levy functions using different platforms and parameter-set to validate the findings of this study.

Application:

CS is successfully used to solve scheduling problems and used to solve design optimization problems in structural engineering. In many applications like speech reorganization, job scheduling, global optimization. Cuckoo search idealized such breeding behavior and can be applied to various optimization problems.CS algorithm is in various domains including Industry, Image processing, wireless sensor networks, flood forecasting, document clustering, speaker recognition, shortest path in a distributed system, in the health sector.

One rank Cuckoo Search algorithm for solving Economic Load Dispatch problems: Nguyen Thang Trung, Ngoc Dieu proposed a paper on power generations to minimize the cost of fuels. The problem is considered a non-convex and piecewise quadratic fuel cost function of thermal units in the objective of the problem with complicated constraints such as prohibited operating zones (POZ) and power loss. CSA is used mainly for searching optimal solutions based on random walks. One Rank Cuckoo Search algorithm is used i.e. Two modifications are done to the basic CSA method to enhance its searchability for an optimal solution within minimum time. The first modification is the exploitation and exploration phase corresponding to the new solution via levy flight of new eggs is the replacement of egg. The second modification is the technique for handling inequality constraints. i.e. The ORCSA was proposed by Ahmed et.al 2013. In this, the input is in power with probability to generate in different values. This method obtains better cost with time when compared to other methods. It is a more efficient method for solving ELD.

The cuckoo search algorithm for feature selection: A Feature or variable or attribute refers to elements of data. Feature selection is a data pre-processing technique used in the classification of IDS. It is used to remove the unnecessary, redundant attributes from the given datasets. It improves accuracy and decreases training time. The main aim is to maximize the classification performance and minimizing the number of features. The feature selection can be done with BCS (Binary Cuckoo Search Algorithm), the problem is which features to select, or which features are not to select in a given problem, to solve this the binary vector is used i.e. where 1 is either to select the feature for a given data and 0 is or to select.

Advantages:

An important advantage of this algorithm is its simplicity. In fact, comparing with other population or agent-based-metaheuristic algorithms such as particle swarm optimization and harmony search, there is essentially only a single parameter Pa in CS (apart from population size n). Therefore it is easy to implement.

Cuckoo search is a new meta-heuristic search algorithm enjoying extensive research prospects. By measuring it with the above performance metrics, it can be concluded that the cuckoo search has advantages in terms of great robustness, on the whole, readable logic, portability, and platform independence. The algorithm also demonstrates advantages in strong global search ability, few selected parameters, excellent search path, and strong ability to solve multi-objective problems. Cuckoo search solves the continuance problems which cannot be handled by ACO. The cuckoo search demonstrates obvious advantages, including its excellent global searchability. Due to the characteristics of Levi flight, the cuckoo search can better avoid falling into the local optimum. Also, cuckoo search shows good applicability when dealing with multi-objective problems. What's more, the Levi flight demonstrated in cuckoo search is an efficient global random search algorithm and has been proven to perform better than other algorithms in optimal problems. The rise of the cuckoo algorithm over the past few years has confirmed that based on the advantages mentioned above, it can be said to be a new star among swarm intelligence algorithm sewer parameters in the formula of, it is less affected by other factors and more robust.

Disadvantages:

The original cuckoo search can only be used to solve continuous problems. As for discrete problems and multi-objective problems, however, it fails to perform well. Therefore, the algorithm has certain limitations when processing some other problems, which is exactly what we need to further study and improve. As for the continuous problems that the algorithm can deal with, there has been much advancement in terms of step size, parameter, intercoupling with other algorithms, and other factors used to improve the performance of related indicators. Meanwhile, this algorithm also has problems in adaptability and obtaining ideal search results, and its ability to solve complex problems is limited. Future research should be directed towards studying how to explore new methods and strategies to improve the performance of high coupling functions between variables. The cuckoo search algorithm also has slower convergence.

Future work:

CS has been applied to many areas such as engineering optimization and diverse applications. In addition, many industries also used CS for their optimization tasks and the improvement of their industrial processes. However, there is still plenty of room for improvement. From our survey and observations, we expect that further research in the following areas using CS will be very useful:

- Large-scale global optimization. combinatorial optimization, scheduling, and traveling salesmen problem.
- Hybridization with other algorithms.
- Systematical theoretical analysis self-adaptation, like in the firefly algorithm by Fister working with ensemble strategies various representations parameter tuning and control more industrial applications

References:

[1] X.-S. Yang and S. Deb, "Cuckoo search: recent advances and applications", Neural Computing and Applications, vol. 24, no. 1, pp. 169-174, 2014.

[2] L. Fredriksson, "A brief survey of lévy walks: with applications to probe diffusion", 2010.

[3] X.-S. Yang, Cuckoo search and firefly algorithm: Theory and applications, Springer, vol. 516, 2013.

[4] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights", Proc. World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), pp. 210-214, 2009.

[5] X.S. Yang and S. Deb, "Cuckoo search: recent advances and applications", Neural Computing and Applications, vol. 24, no. 1, pp. 169-174, 2013.

[6] H. Soneji and R. C. Sanghvi, "Towards the improvement of Cuckoo search algorithm", Information and Communication Technologies (WICT), pp. 878-883, 2012.

[7] X. S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[8] Goldberg, D.E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional.

[9] Kennedy, J. and Eberhart, R. 1995. Particle swarm optimization. Neural Networks, 1995. Proceedings., IEEE International Conference on (1995), 1942–1948.

[10] Tan, Y. et al. 2011. Advances in Swarm Intelligence, Part I: Second International Conference, ICSI 2011, Chongqing, China, June 12-15, 2011, Proceedings. Springer. [11] Baghel, M. et al. 2012. “Survey of Metaheuristic Algorithms for Combinatorial Optimization.” International Journal of Computer Applications. 58, 19 (Nov. 2012), 21–31.

[12] Blum, C. and Roli, A. 2003. “Metaheuristics in combinatorial optimization: Overview and conceptual comparison.” ACM Computing Surveys (CSUR). 35, 3 (2003), 268–308.