International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 4, Issue 1, January-February 2015 ISSN 2278-6856

# A Review on Nature Inspired Algorithms for Clustering

**Radha A. Pimpale[1], P.K. Butey[2]**

[1] Ph,D Scholar, RTMNU, Nagpur

[2]Kamala Nehru Mahavidyala Nagpur

## Abstract

***In this paper we are proposed most suitable Nature inspired algorithms for clustering. Nature inspired algorithms like cuckoo search, firefly algorithms, Bat algorithms and wolf algorithms are metaheuristic, multiobjective, stochastic algorithms overcome the decencies of traditional clustering algorithms, nature inspired algorithms have get global optima. These algorithms are suitable for the random search, using this can find new solution space for the global optimum. These algorithms are iterative in nature. This paper presents a complete review of given nature inspired algorithms suitable for clustering. Parameter like Convergence, speed, accuracy may be considered. These algorithms help us to adapt environment always and mutual cooperation, which give us many new ideas for solving complex problems.***

**Key words:-** cuckoo, firefly, bat and wolf algorithms.

## 1.INTRODUCTION

Traditional clustering algorithms have some drawback it reach towards local optimum solution, initial point awareness, local point convergence, high time complexity and poor scaling of large number of point to overcome this nature inspired algorithms are introduced. Nature inspired algorithms are design for to explore new approach, new algorithms and to hybrid the algorithms. Here different type of Nature inspired population based algorithms are considered for clustering because it can process in parallel, more no of population can be used , it can deal with large scale problem, clustering can be solved easily, it can optimize intra cluster variance, inter cluster separation, can be get quantization error, so it become natural and appropriate for clustering. Optimization is the finding best solution for the given problem. Optimization helps to maximized performance of the system with minimum runtime and resources [9]. Clustering refers to partitioning the unlabeled data objects into certain number of clusters. The primary objective of clustering is to achieve homogeneity within cluster i.e. objects belonging to the same cluster should be as similar as possible. Secondly to maintain heterogeneity amongst the clusters i.e. objects belonging to different clusters should be as different as possible. Thus the algorithmic task has been stated as an optimization problem [11]. Clustering is an optimization problem, in which all data point should be contained at least one clustered data, all data point should be clustered and different clustered should have no data in common for this we are introducing here suitable, nature inspired algorithms for clustering.

### 1.1Cuckoo Search Algorithms

Cuckoo search was based on the breeding behaviour of some cuckoo species invented by Yang and Deb 2009; Yang and Deb, 2010b; Yang and Deb, 2013; Gandomi et al, 2013b cuckoo search (CS) was based on the brooding behaviour of some cuckoo species which was combined with L´evy flights. Cuckoo Optimization Algorithm (COA) is one of evolutionary techniques is inspired by the lifestyle of a bird called the Cuckoo. This bird did't made nest for itself and it be used the nests of other birds for laying eggs. Ability to create eggs like the bird host is reinforced in cuckoo bird. If the bird's host discover eggs that are not mine, it throw away or leave the nest and it makes a nest in other places. Cuckoo eggs are the bigger size of the host bird until cuckoo brood would hatch soon. Every cuckoos care about his nest. When the host bird's eggs throws out of the nest or demand food so much to other broods die of hungry. When the cuckoo brood grows and becomes a mature bird continues the mother's life instinctively [1]. Pseudocode from cuckoo optimization algorithm [2]

1. Initialize cuckoo habitats with random points
2. Define ELR for each cuckoo
3. Let cuckoo to lay eggs inside their corresponding ELR
4. Kill those eggs that are identified by host birds
5. Eggs hatch and chicks grow
6. Evaluate the habitat of each newly grown cuckoo
7. Limit cuckoos' maximum number in environment and kill those that live in worst habitats
8. Cuckoos find best group and select goal habitat
9. Let new cuckoo population move toward goal habitat
10. If stop condition is satisfied end, if not go to 2.

The basic idea behind this each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest.

**Cuckoo search Algorithms [2]**

**Step 1: Initialization**

1. Generated initial population, Initial habitats and the number of eggs per Cuckoo are randomly initialized, nests represent population
2. Initialize centroid. Select the number of cluster, cuckoo nests, and eggs in nests to start the search. Each nest has multiple eggs representing a set of solutions.

**Step 2: Formation of Clusters**
The clusters are formed, by Cuckoo Search technique. Each egg in a nest is like data point. A group of M nests are chosen with N eggs in it. The probability of choosing the best egg or quality egg is done by random walk. Step size and Levy angle is updated. In turn the nests are updated. The optimal solution *i.e.*; best egg is taken as Cluster Head, The worse nets are abandoned in normal Cuckoo Search.

**Step 3:** After the clusters are formed, The inter cluster and intra cluster distance is calculated. Intra cluster refers to communication between cluster head and non cluster head nodes within the cluster. Inter cluster communication refers to communication between the clusters. Add the value of Object with the minimum distance

(Object = Object+ min (|Center, m|)
i=1,…,K (1)

**Step 4: laying eggs in host birds' nests**
The cuckoo egg radius is computed based on equation 1. Egg laying is done randomly within a circle-shape area with determined radius. Then, the objective function of each egg is calculated; 10 % of the egg's population with improper cost function will be identified and replaced by the host birds.

**Step 5: Cuckoo immigration Cuckoos move depend on current location.**
After eggs grow up and turn into adult cuckoo, the best cuckoo Gbest is identified. Other cuckoos will start migrating toward this cuckoo according to the explanations presented in section 4. In the case that $\alpha$ is greater than 0.01, the parameter amount should be reduced.

**Step 6: Elimination of the cuckoos in worst habitats**
If the total of all available cuckoos exceed the maximum number of them, the cuckoos in worst habitats with undesirable cost function will be eliminated.

**Step 7:** if the stop condition is maintained, the algorithm will stop. Otherwise, the determined egg laying radius will be determined according to Equation(1) and algorithm will be performed from the 3rd step.

## 2. Firefly algorithm

Firefly algorithm is a nature-inspired metaheuristic optimization algorithm, inspired by the flashing behaviour of fireflies. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. Xin-She Yang formulated this firefly algorithm by assuming [5].

1. All fireflies are unisexual, so that one firefly will be attracted to all other fireflies;

2. Attractiveness is proportional to their brightness, and for any two fireflies, the less bright one will be attracted by (and thus move to) the brighter one; however, the brightness can decrease as their distance increases;
3. If there are no fireflies brighter than a given firefly, it will move randomly [6].

We can idealize some of the flashing characteristics of fireflies so as to develop firefly -inspired algorithms. Flashing characteristics of fireflies is used to develop firefly inspired algorithm. Firefly Algorithm (FA or FFA) developed by Xin-She Yang at Cambridge University in 2007, use the following three idealized rules:

- All the fireflies are unisex so it means that one firefly is attracted to other fireflies irrespective of their sex.
- Attractiveness and brightness are proportional to each other, so for any two flashing fireflies, the less bright one will move towards the one which is brighter. Attractiveness and brightness both decrease as their distance increases. If there is no one brighter than other firefly, it will move randomly.
- The brightness of a firefly is determined by the view of the objective function. For a maximization problem, the brightness is simply proportional to the value of the objective function. Other forms of the brightness could be defined in an identical way to the fitness function in genetic algorithms.
- Light Intensity And attractiveness In the firefly algorithm, there are two important points:

The variation in the light intensity and formulation of the attractiveness. For simplicity, we can assume that the attractiveness of a firefly is determined by its brightness which in turn is connected with the encoded objective function. In the simplest case for maximum optimization problems, the brightness I of a firefly for a particular location x could be chosen as I(x) f(x). Even so, the attractiveness $\beta$ is relative, it should be judged by the other fireflies. Thus, it will differ with the distance rij between firefly i and firefly j. In addition, light intensity decreases with the distance from its source, and light is also absorbed by the media, so we should allow the attractiveness to vary with the varying degree of absorption [7].

**Firefly algorithm [10]**
**Step I:** Initialize algorithm parameters:
MaxGen: the maximal number of generations
$\gamma$: the light absorption coefficient
r: the particular distance from the light source
d: the domain space
Define the objective function of f(x), where x=(x1,........,xd)T
**Step II:** Generate the initial population of fireflies or x i (i=1, 2 ,..., n)
**Step III:** Determine the light intensity of Ii at xi via f(xi)
While (t<MaxGen)
For i = 1 to n (all n fireflies);
For j=1 to n (n fireflies)
If (Ij> Ii),
move firefly i towards j by using 11
equation;
end if
Attractiveness varies with distance r via Exp

[$\gamma r^2$];
Evaluate new solutions and update
light intensity;
End for j;
End for i;
Rank the fireflies and find the current best;
End while;
Post
**Step IV:** process results and visualization
**Step V:** End procedure

## 3. Bat Search optimization

Bat Algorithm for continuous constrained optimization problems was introduced by Yang in 2010 [3]. It simulates the echolocation behaviour of microbats as microbats can generate high echolocation. The Bat produces a very high sound to detect its prey which echoes back with some frequency. The capability of echolocation of microbats is fascinating as these bats can find their prey and discriminate different types of insects even in complete darkness. By observing the bounced frequency of sound, bats are able to distinguish between the prey and obstacle and can sense the distance between them in their nearby surroundings. They fly randomly with some velocity, frequency and sound (loudness) to search for food. Solution of objective function is to find prey at minimum distance. The frequency and zooming parameters maintain the balance between exploration and exploitation processes. The algorithm continued till convergence criteria are satisfied. bats` echolocation characteristics should be idealized:

a. Objects` distances are always perfectly sensed by the echolocation system on bats. This considers the ability to differentiate between different objects even in darkness.
b. Bats are flying randomly with velocity vi, fixed frequency fmin at position xi, and fluctuating wavelength $\lambda$, and loudness from A0 to Amin to search for its prey. Wavelength or frequency can be changed spontaneously by adjusting the pulse emission rate r $\in$ [0, 1], based on the closeness of the bat`s objective.
c. Variation of the loudness parameter takes vales between large loudness (A0) and minimum loudness (Amin).

**Bat Algorithm**
If we idealize some of the echolocation characteristics of microbats, we can develop various bat-inspired algorithms or bat algorithms. For simplicity, we now use the following approximate or idealized rules:

**Step I-** All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way;

**Step II-** Bats fly randomly with velocity vi at position xi with a fixed frequency fmin, varying wavelength _ and loudness A0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission r 2 [0, 1], depending on the proximity of their target;

**Step III-** Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A0 to a minimum constant value Amin. Another obvious simplification is that no ray tracing is used in estimating the time delay and three dimensional topography. Though this might be a good feature for the application in computational geometry, however, we will not use this as it is more computationally extensive in multidimensional cases. In addition to these simplified assumptions, we also use the following approximations, for simplicity. In general the frequency f in a range [fmin, fmax] corresponds to a range of wavelengths [_min, _max].

Pseudocode for Bat Algorithm
1. Objective function f(x), x = (x1, ..., xd)T
2. Initialize the bat population xi (i = 1, 2, ..., n) and vi
3. Define pulse frequency fi at xi
4. Initialize pulse rates ri and the loudness Ai
5. while (t <Max number of iterations)
6. Generate new solutions by adjusting frequency,
7. and updating velocities and locations/solutions [equations (2) to (4)]
8. if (rand > ri)
9. Select a solution among the best solutions
10. Generate a local solution around the selected best solution
11. end if
12. Generate a new solution by flying randomly
13. if (rand < Ai & f(xi) < f(x_))
14. Accept the new solutions
15. Increase ri and reduce Ai
16. end if
17. Rank the bats and find the current best x_
18. end while

## 4. The wolf colony algorithm

The wolf colony algorithm (WCA) is proposed by C. G Yang et al .in 2007. The algorithm is a swarm intelligence algorithm to simulate the intelligent predatory behaviors of the wolf colony. The wolf is a very intelligent animal. They are not alone when they catch and feed on food but by teams composed of several wolves The wolf colony sends a few wolves to search quarry by smell. When the searching wolves discover the quarry, they notify the position of the quarry to the other wolves by howl The other wolves get close to the quarry and besiege it. After they get the quarry, they distribute the food according to the strength of the wolf. At last, the weak wolves will be eliminating [4]. The wolf pack unites and cooperates closely to hunt for the prey in the Tibetan Plateau, which shows wonderful skills and amazing strategies. Inspired by their prey hunting behaviours and distribution mode, we abstracted three intelligent behaviours, scouting, calling, and besieging, and two intelligent rules, winner-take-all generation rule of lead wolf and stronger-survive renewing rule of wolf pack. Then we proposed a new heuristic swarm intelligent method, named wolf pack algorithm (WPA). The wolf pack is marvellous. Harsh living environment and constant evolution for centuries have created their rigorous organization system and subtle hunting behaviour. Wolves tactics of Mongolia cavalry in Genghis Khan period, submarine tactics of Nazi Admiral Doenitz in World War II and U.S. military wolves attack system for electronic countermeasures all highlight great charm of their swarm intelligence. Proposes a wolf colony algorithm (WCA) to solve the optimization problem. But the accuracy and efficiency of WCA are not good enough and easily fall into local optima, especially for high-dimensional functions. So, in this paper, we reanalyzed collaborative predation behavior and prey distribution mode of wolves and proposed a new swarm intelligence algorithm, called wolf pack algorithm (WPA); Moreover, the efficiency and robustness of the new algorithm were tested by compared experiments. Wolves are gregarious animals and have clearly social work division. There is a lead wolf; some elite wolves act as scouts and some ferocious wolves in a wolf pack. They cooperate well with each other and take their respective responsibility for the survival and thriving of wolf pack. Firstly, the lead wolf, as a leader under the law of the jungle, is always the smartest and most ferocious one. It is responsible for commanding the wolves and constantly making decision by evaluating surrounding situation and perceiving information from other wolves. These can avoid the wolves in danger and command the wolves to smoothly capture prey as soon as possible. Secondly, the lead wolf sends some elite wolves to hunt around and look for prey in the probable scope. Those elite wolves are scouts. They walk around and independently make decision according to the concentration of smell left by prey; and higher concentration means the prey is closer to the wolves. So they always move towards the direction of getting stronger smell. Thirdly, once a scout wolf finds the trace of prey, it will howl and report that to lead wolf. Then the lead wolf will evaluate this situation and make a decision whether to summon the ferocious wolves to round up the prey or not. If they are summoned, the ferocious wolves will move fast towards the direction of the scout wolf. Fourthly, after capturing the prey, the prey is not distributed equitably, but in an order from the strong to the weak. That is to say that, the stronger the wolf is, the more the food it will get is. Although this distribution rule will make some weak wolf dead for lack of food, it makes sure that the wolves that have the ability to capture prey get more food so as to keep being strong and can capture more prey successfully in the next time. The rule avoids that the whole pack starves to death and ensures its continuance and proliferating. In what follows, the author made detailed description and realization for the above intelligent behaviours and rules. Having discussed all the components of WPA, the important computation steps are detailed below [5].

**Algorithms**

**Step 1. (initialization)-** Initialize the following parameters, the initial position of artificial wolf , the number of the wolves , the maximum number of iterations , the step coefficient , the distance determinant coefficient , the maximum number of repetitions in scouting behaviour , and the population renewing proportional coefficient.

**Step 2.** The wolf with best function value is considered as lead wolf. In practical computation, , which means that wolves except for lead wolf act with different behaviour as different status. So, here, except for lead wolf, according to formula (2), the rest of the wolves firstly act as the artificial scout wolves to take scouting behaviour until or the maximum numbers of repetition is reached and then go to Step 3.

**Step 3-** Except for the lead wolf, the rest of the wolves secondly act as the artificial ferocious wolves and gather towards the lead wolf according to (3); is the smell concentration of prey perceived by wolf ; if , go to Step 2; otherwise the wolf continues running until ; then go to Step 4.

**Step 4-** The position of artificial wolves who take besieging behaviour is updated according to (4).

**Step 5-**Update the position of lead wolf under the winner-take-all generating rule and update the wolf pack under the population renewing rule according to (6).

**Step 6-** If the program reaches the precision requirement or the maximum number of iterations, the position and function value of lead wolf, the problem optimal solution, will be outputted; otherwise go to Step 2.

**Pseudocode for Wolf Pack Clustering Algorithm:**

1. Input: Objects (population size) to be clustered (N), the number of clusters (K)
2. Initialize the control parameters of the algorithm;
3. Randomly assign the k clusters with each of the N wolves initialized in step 1;
4. For each wolves select K objects from S data objects as initial centroids;
5. While (exit criteria not met) do
6. Calculate the fitness of the centroid in each wolf and find the best solution;
7. The wolves prey randomly, looking for a nearby companion that has the best fitness within the visual dist ance;
8. If (companion fitness> self fitness)
9. Approach to the companion;
10. Else repeat step 5;
11. While (visual distance < random number generated [0 -1]) do
12. Escape to the new position beyond the visual range chosen at random;
13. Update the centroids according to the new position of the wolves;
14. Reassign the clusters;
15. Output the best cluster configuration represented by the wolf with greatest fitness;
16. End while;
17. End while

## 5.Conclusion

Above Algorithms overcome the decencies of traditional algorithms like is optimal solution has not been achieved, speed and accuracy is not perfect. In traditional algorithms, there is need to defined clustering before it is applied, but in given iterative algorithms overcome these decencies. The search of the solution space is started from a more proper area through Cuckoo algorithms improves convergence speed ,due to cuckoos' egg laying radius are made by the repetition of the reduced algorithm results the less random changes in solution space. So proper level of the clustering accuracy. Cuckoo search algorithms has very good convergence behaviour, very effective, Convergence and accuracy is more, Speed is fasted, algorithms avoid local optima Cuckoos algorithms is simple to implement. Firefly algorithms ability to do automatic subdivision which allow the fireflies to be able to find all optima simultaneously, it also have ability to deal with multimodelling, it have high ergodicity and diversity in solution so Firefly algorithms is unique and efficient. Firefly algorithms have scalability, it can deal with different types of attributes, it can discover clusters with arbitrary shape, minimal requirements for domain knowledge to determine input parameters, ability to deal with noise and outliers, insensitivity to order of input records, high dimensionality, interpretability and usability. Bat algorithms is accurate and efficient because of Frequency tuning, Automatic zooming, Parameter control Wolf Algorithms can be handling to get global objective function value, speed up the process finding centroid in the cluster All above algorithms have features such as self -organization, no central control, derivative free, easy to implement, worked with both globular and non -globular data and independent of Initial solutions.

## References

[1] Samane Asadi, Vahid Rafe, “Presenting a method for Clustering Using Cuckoo Optimization Algorithm”, International Advances in Engineering and Technology (IAET) ISSN: 2305-8285 Vol.22 2014 www.scholarism.net International Scientific Researchers (ISR).

[2] R. Rajabioun, “Cuckoo Optimization Algorithm, Applied Soft Computing” (2011) 5508–5518.

[3] X. S. Yang, “Bat algorithm: literature review and applications”, International Journal of Bio-Inspired Computation, Vol. 5, No. 3, 2013, pp. 141–149.

[4] Qiang Zhou ,Yongquan Zhou and Xin Chen, “A Wolf Colony Search Algorithm Based on the Complex Method for Uninhabited Combat Air Vehicle Path Planning”, International Journal of Hybrid Information Technology Vol.7, No.1 (2014), pp.1, 83,ISSN:1738-9968 IJHIT.

[5] Hu-Sheng Wu and Feng-Ming Zhang, “Wolf Pack Algorithm for Unconstrained Global Optimization”, Materiel Management and Safety Engineering Institute, Air Force Engineering University, Xi’an 710051, China Materiel Engineering Institute, Armed Police Force Engineering University, Xi’an 710086, China, Received 28 June 2013; Revised 13 January 2014; Accepted 27 January 2014; Published 9 March 2014.

[6] Yang, X. S, “Nature-Inspired Metaheuristic Algorithms”, Frome: Luniver Press. ISBN 1-905986-10-6, 2008.

[7] J. Senthilnath, S. N. Omkar and V. Mani, “Clustering using firefly algorithm: Performance study”, Swarm and Evolutionary Computation, June (2011), doi:10.1016/j.swevo.2011.06.003.

[8] Y. Liu and K. M. Passino, “Swarm Intelligence: A Survey”. International Conference of Swarn Intelligence, 2005.

[9] Saibal K. Pal, Dr.C.S Rai, Dr. Prof. Amrit Pal Singh, “Comparative Study of Firefly Algorithm and Particle Swarm Optimization for Noisy Non-Linear Optimization Problems”, Conference of Swarm Intelligence. International Journal of Intelligent Systems and Applications, 2012, 10, 50-57, published Online September 2012 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijisa.2012.

[10] Ehsan Valian, Shahram Mohanna and Saeed Tavakoli,” Improved Cuckoo Search Algorithm for Global Optimization”, IJCIT-2011-Vol.1-No.1 Dec. 2011.

[11] Brucker, P., “On the complexity of clustering problems. In: M. Beckmann and H. K¨unzi (Eds.): Optimization and Operations Research”. Lecture Notes in Economics and Mathematical Systems, 157, Springer-Verlag, Heidelberg, 45-54, 1978.

**AUTHOR**

**Dr . Pradeep Butey** is presently working as Head of the Department of Computer Science, Kamla Nehru Mahavidyalaya, Nagpur, He has done his doctorate in computer Science from RTM Nagpur University, Nagpur, in 2011 His areas of interest include Data mining, fuzzy logic, optimization, RDBMS. He has 18 Research papers published in reputed national and International Journals. He is Chairman Board of Studies in Computer Science of the University, He has vast experience of 25 years in the field of teaching and research in Computer Science. He has worked as a member of selection committee at many instances. He is a Member of Academic Council RTM Nagpur University.

**Prof. Radha A. Pimpale** is presently working as Head of Department of MCA at Priyadarshini Institute of Engineering and Technology, Nagpur. She has completed her MCA degree from SGBAU Amravati. She has done M. Phil from Algappa University Karaikudi. She has 3 research papers published in reputed international journals. She is persuing Ph. D, in Computer Science. She has total experience of 12 years in teaching and research in the field of Computer Science