IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

Analysis & Visualization of Multidimensional GIS Images Using Multi Objective Algorithm (MOA)

Sanjay Srivas[1*], P. G. Khot[2]

[1]Dept. of Electronics & Information Technology, RTM Nagpur University Nagpur, Maharashtra, India.
[2]Dept. of Statistics, RTM Nagpur University Nagpur, Maharashtra, India.

**Corresponding Author: sanjaysrivas.iit@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 17/Aug/2018, Published: 31/Aug/2018

Abstract— Geographical data related to image processing, environmental monitoring and urban planning are beings collected and stored in various databases. Processing and managing these voluminous multidimensional data have become an important requirement and gained the researchers attention. For any application dealing with the multidimensional data analysis, efficient and effective data processing techniques are required to produce best results from these geographical data sets. The processing of these datasets in timed manner using appropriate techniques is the ultimate requirement while dealing with multidimensional data. There are number of optimization methods available but the Nature-inspired algorithms are among the most powerful algorithms for optimization. We proposed Multi Objective Algorithm (MOA) which is the combination of Dragon Fly (DF) Optimization and Cuckoo Search (CS) Algorithm for Visualization & Data Analysis of Geospatial database. We have compared the various parameters from MOA algorithm with the existing K-nearest neighbors (KNN) algorithm. Results indicate that the MOA algorithm is producing better output in term of classification compare to existing algorithm. Finally, the proposed algorithm provides a framework where image classification and interpretation can be possible for various types of GIS images.

Keywords — Data Analysis, Cuckoo Search, Dragon Fly Optimization, Optimization, Levy flight, Visualization, K-Nearest Neighbour

I. INTRODUCTION

Digital/Satellite image processing is a technique which includes extraction of information from the digital image. These images are always considered as voluminous in size because it contains the large number of pixels in it. Processing of the large GIS images using appropriate techniques are utmost requirement to reduce the execution time. Optimization is a process of adapting a system to make certain features work more efficiently. Process of maximizing the desired parameter and reducing the error contributory parameters to increase the overall performance of system. Maximizing means trying to get noble results without doing alteration in base structure. If the algorithm is optimized, it runs faster or runs with less memory requirements to produce the effective output in minimum processing time. Optimization can be classified in many ways. In this study, we have considered two major Meta heuristic algorithms for solving image analysis and visualization in GIS Images [1].

The Cuckoo Search (CS) optimization is introduced in 2009 by Yang and Deb. Cuckoo Search Algorithm is inspired by the bird cuckoo species in nature. This algorithm is used for the continuous problems and NP-hard problems. This algorithm is tested by the many researchers on some benchmark functions and compared with the other algorithms like PSO (Practical Swarm Optimization) and GA (Genetic Algorithms), and the solutions obtained for cuckoo search algorithm is better than the other algorithms like PSO and GA. These algorithms are applied to Engineering optimization [2],[3],[4].

The dragonfly algorithm (DA) is a new metaheuristic optimization algorithm, based on the simulation of the buzzing behavior of the dragonflies. This algorithm was developed by Mirjalili (2016) and preliminary studies have shown its potential to solve numerous problems of optimization of reference and complex problems of optimization of computational fluid dynamics (CFD).

II. RELATED WORK

A. Cuckoo Search Algorithm Components

Cuckoo search algorithm is a nature-inspired algorithm developed based on reproduction of cuckoo birds. While working with CS algorithms, it is important to associate potential solutions with cuckoo eggs. Cuckoos normally lay their fertilized eggs in other cuckoos' nests with the hope of their off-springs being raised by proxy parents. There are times when the cuckoos discover that the eggs in their nests do not belong to them, in those cases the foreign eggs are either thrown out of the nests or the whole nests are abandoned. The CS optimization algorithm is basically based on the following three rules [5], [6]:

- Each cuckoo selects a nest randomly and lays one egg in it.
- The best nests with high quality of eggs will be carried over to the next generation.
- For a fixed number of nests, a host cuckoo can discover a foreign egg with a probability Pa ϵ [0,1]. In this case, the host cuckoo can either throw the egg away or abandon the nest and build a new one somewhere else.

Levy flights are random walks whose directions are random and their step lengths are derived from the Levy distribution. These Levy flights are performed by animals and insects and it is characterized by series of straight flights followed by sudden turns. Compared to normal random walks, Levy flights are more efficient in exploring large–scale search areas.

<u>Pseudo Code of the CSA</u>

Begin

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 criterion)*
Get a cuckoo randomly by Levy Flights
Evaluate its fitness Fi
Choose a nest among n (say j) randomly
***If** (Fi > Fj)*
Replace j by the new solution;
End If
A fraction (pa) of worse nests is abandoned and new ones are built;
Keep the best solutions (or nest with quality solutions)
Rank the solution and find the solution and find the current best
End while
Post process results and visualization

End Begin

B. Dragonfly Algorithm Components

The main objective of any swarm is survival, so all of the individuals should be attracted towards food sources and distracted outward enemies. It is also shown in Figure 1. Considering these two behaviours, there are five main factors in position updating of individuals in swarms. According to Reynolds, the behaviour of swarms follows three primitive principles [7],[8],[9]:

- **Separation**, which refers to the static collision avoidance of the individuals from other individuals in the neighbourhood.
- **Alignment**, which indicates velocity matching of individuals to that of other individuals in neighbourhood.
- **Cohesion**, which refers to the tendency of individuals towards the centre of the mass of the neighbourhood.

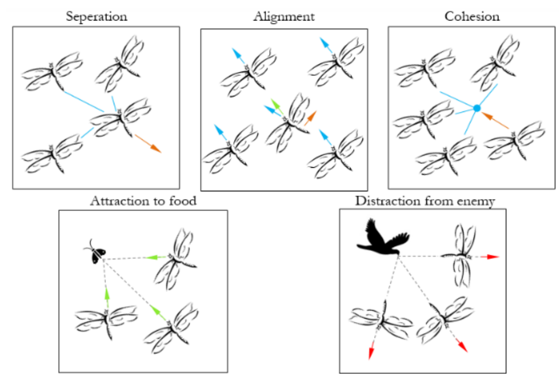


Figure 1: Primitive priniciples of DFA (Source: https://en.wikiversity.org/wiki/Dragonfly_algorithm)

<u>Mathematical Model for these behaviours as follows:</u>

- The separation is calculated as follows:

$$S_i = \sum_{j=1}^{N} X - X_i \quad \text{Eqn 1)}$$

where, X is the position of the current individual, Xj shows the position jth neighbouring individual, and N is the number of neighbouring individuals.

- Alignment is calculated as follows:

$$A = \frac{\sum_{j=1}^{N} V_j}{N} \quad \text{Eqn 2)}$$

where Xj shows the velocity of jth neighbouring individual.

- The cohesion is calculated as follows:

$$A = \frac{\sum_{j=1}^{N} V_j}{N} \quad \text{Eqn 3)}$$

where X is the position of the current individual, N is the number of neighbourhoods, and Xj shows the position jth neighbouring individual.

- Attraction towards a food source is calculated as follows:

$$F_i = X^+ - X \quad \text{Eqn 4)}$$

where X is the position of the current individual, and X^+ shows the position of the food source.

- Distraction outwards an enemy is calculated as follows:

$$E_i = X^- + X \quad \text{Eqn 5)}$$

where X is the position of the current individual, and X^- shows the position of the enemy.

- The update velocity vector as follows:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad \text{Eqn 6)}$$

where s shows the separation weight, S_i indicates the separation of the i-th individual, a is the alignment weight, A_i is the alignment of i-th individual, c indicates the cohesion weight, C_i is the cohesion of the i-th individual, f is the food factor, F_i is the food source of the i-th individual, e is the enemy factor, E_i is the position of enemy of the i-th individual, w is the inertia weight, and t is the iteration counter.

<u>Pseudo Code of the DA</u>

Begin

Define population size (N)
Set the iteration counter t=1
Initialize the population by generating Xi for i = 1, 2, 3…, N
Calculate the objective function values of all dragonflies
Update the food and the predator's location
***While** (the stop criterion is not satisfied)*
***For** i=1: N*
Update neighborhood radii (or update w, s, a, c, f, and e)
If a dragonfly has at least one neighborhood dragonfly
Calculate
Separation motion
Alignment motion
Cohesion motion
Food attraction motion
Predator distraction motion
Update position vector
Else
Update position vector using the Levy flight function
End if
***End for** i*
Sort the population/dragonflies from best to worst and find the current best
End while
Post-process and visualize the results

End

III. PROPOSED ALGORITHM

A. Multi Objective Algorithm

We now outline our proposed algorithm and parameters which have been considered while forming new algoithm. We have integrated the objective functions of Dragonfly and Cuckoo Search Algorithms to arrive at hybrid apporach called *Multi Objective Algorithm* that will improve the optimization properties. Below is the flow daigram of MOA shown in Figure 2.

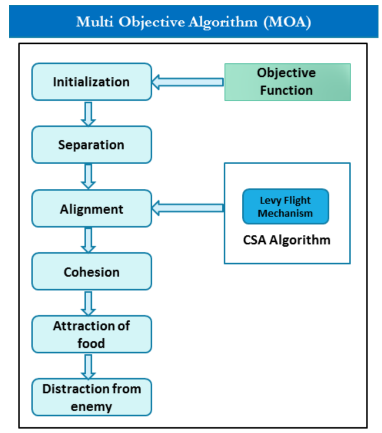


Figure 2: Flow diagram of MOA

Block daigram of the proposed method is shown in Figure 3. The entire processesing is divided into following steps: Pre-processing, Feature extraction using KNN & KNN+MOA Classifcation. Initially we had applied pre-processing on input image like subsetting, filtering and echancement. Post that Frue Color Composite (FCC) applied for feature extraction using principle component analysis.

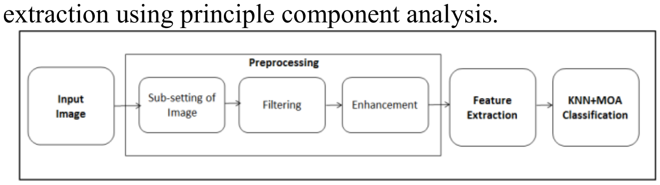


Figure 3: Block diagram of proposed method

B. Data Used:

To perform this experiment, two satellite imageries have been taken over the 25 years of time period (1991-2016). Both the images have been downloaded from the website: https://remotesensing.usgs.gov/ which is open access. Additional information of the images like: Latitude, Longitude, Acquisition date, Band and Sensor are described in Table-1.

Table 1: Details of data

Acquisition Date	**14 March, 1991**	**2 March, 2016**
Band	753	764
Sensor	Landsat 5	Landsat 8
Latitude	28.5067	28.5067
Longitude	77.4829	77.4829

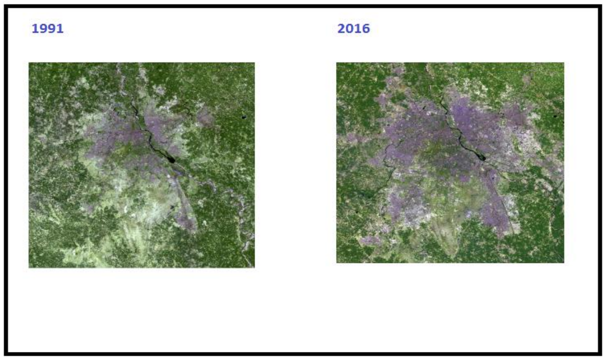


Figure 4: Input datasets

C. Pre-processing
First the input image has to go through the pre-processing stage. In the pre-processing stage, we will perform following steps to make our input ready for further steps of processing.

- Sub-setting
- Filtering
- Enhancement

D. Sub-setting
Sub-setting is a process where we have truncated the additional vegetation area from the image and kept only the relevant part of the image which can be further used to identify the built-up, water bodies, open land and vegetation area.

E. Filtration
In GIS images, filtering is the primary process which is used to enhance the image quality to perform further process on it. Anisotropic Filtering (AF) is a method of enhancing the image quality of textures on surfaces of computer graphics that are at oblique viewing angles with respect to the camera where the projection of the texture appears to be non-orthogonal. Anisotropic filter concentrates on the preservation of important surface features like sharp edges and corners by applying direction dependent smoothing. In this step, we have used Anisotropic filters to remove the noise, and ing, edge detection, correlation, compression, deconvolution, simulation purpose [10], [11].

F. Enhancement
This is the last step of pre-processing process where quality of images would be enhanced to perform further operations like feature extraction and images classification using enhanced image. Enhancement process will increase the image quality which can further used for classification.

Figure 5 depicts end-to-end processes for GIS image processing which comprises of various steps like: Input image, pre-processing, false color composite representation and image classification.

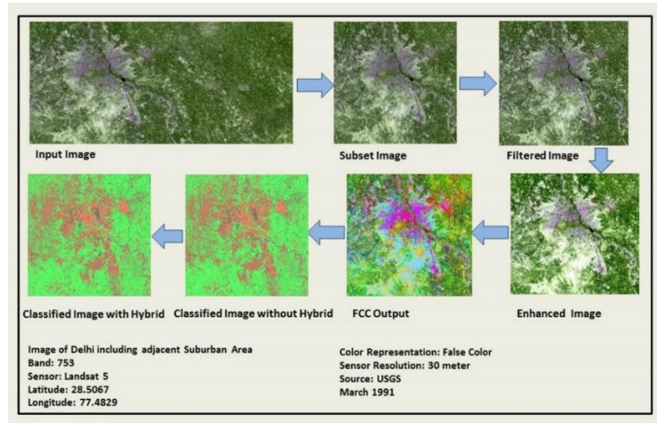


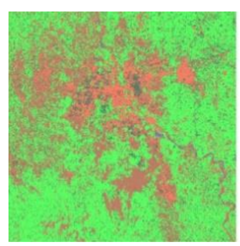
Figure 5: End-to-end GIS image processing in MATLAB

IV. EXPERIMENT AND RESULT

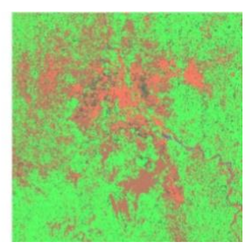
A. Simulation Setup
Performance of proposed algorithm tabulated in Table 2 was investigated in MATLAB version 7.10.0.499 (R201a). The test environment was DELL laptop with the following specifications; RAM of 8.0 GB, CPU is Intel inside CORE i5 and 32-bit windows 7 home Basic operating system.

B. Classification Results
After performing the pre-processing process, we had applied the K-Nearest Neighbour Classifier with Multi Objective Algorithm to classify the land cover such as Built-up, Water bodies, Vegetation and Open land. Results of the classified images of 1991 and 2016 years are as follows:

Delhi 1991:

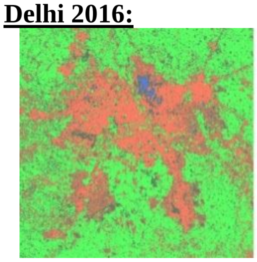


Classified Image of 1991 without MOA

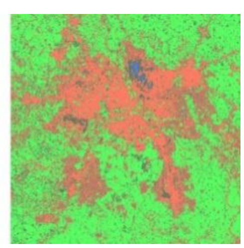


Classified Image of 1991 with MOA

Delhi 2016:



Classified Image of 2016 without MOA



Classified Image of 2016 with MOA

C. Observations
It is observed from the classified images that the built-up areas have increased to the maximum during the study period

1991 to 2016 which is due to urbanization and growth of population density. Built-up area of Delhi shows an overall increment from 520.64 km² to 752.48 km² or 30.81% of the total city area (1484 km²) over the period of 25 years (from 1991 to 2016). Table 2 shows the value of Built-up Area and Built-up density that can observe in Figure 5. The table comprised of value corresponding to the proposed Algorithm. In this case, Built-density has increased after combining the MOA algorithm with the existing classification mechanism.

Table 2: Comparison of Algorithms

Parameters	**Input Images from the Landsat**	
	Delhi 1991	**Delhi 2016**
Total Area (Sq. Km.)	1484	
Built-up Area	520.64	752.48
% Change in built-up due to urbanization	30.81 %	
Built-up density using K-NN	27.72	44.8
Built-up density using K-NN with MOA	35.08	50.71

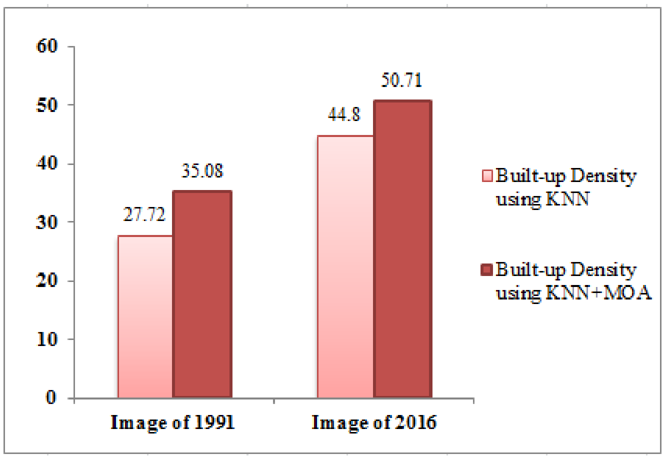


Figure 5: Comparison analysis of Built-up density

V. CONCLUSION

In this paper, we have proposed Multi Objective Algorithm by integrating two Meta heuristic algorithms (i.e., CSA and DA). The objective of proposed algorithm is to classify GIS images. The alignment process in hybrid is modified using levy flight mechanism which enhanced the efficiency of the algorithm. The proposed algorithm was presented to evaluate the parameters like: accuracy, built-up area and built-up density etc. We compared the proposed method with the existing KNN classification method. The algorithm was analyzed for a number of experimental problems and compared with KNN classification method. The results indicate that MOA is outperforming than the traditional data classification method considering the accuracy of built-up area in GIS Image.

REFERENCES

[1] F.S. Gharehchopogh, I. Maleki, S.R. Khaze, "A New Optimization Method for Dynamic Travelling Salesman Problem with Hybrid Ant Colony Optimization Algorithm and Particle Swarm Optimization", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 2, Issue 2, pp 352-358, February 2013.

[2] Xin-She Yang, Suash Deb, "Cuckoo Search via L´evy Flights", IEEE: pp 210-214, 2009.

[3] Ghodrati, A. and Lotfi, S. "A hybrid CS/PSO algorithm for global optimization", in Intelligent Information and Database Systems, Lecture Notes in Computer Science, Vol. 7198, pp 89–98, Springer 2012.

[4] Pinar Civicioglu and Erkan Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms", © Springer Science+Business Media B.V.: pp 315-346, 2011.

[5] Ahmed, H. and Abdelhafid, H. "Cuckoo search optimization for linear antenna arrays synthesis", Serbian Journal of Electrical Engineering, October, Vol. 10, No. 3, pp 371–380, 2013.

[6] Bhargava, V., Fateen, S. and Bonilla-Petriciolet, A. "Cuckoo search: a new nature-inspired optimization method for phase equilibrium calculations", Fluid Phase Equilibria, Vol. 337, pp191–200, 2013.

[7] Cui Z, Alignment particle swarm optimization. In: Cognitive informatics, 2009. ICCI'09. 8th IEEE international conference on, pp 497–501, 2009.

[8] Reynolds CW, Flocks, herds and schools: a distributed behavioral model. ACM SIGGRAPH Comput Gr 21: pp25–34, 1987.

[9] Russell RW, May ML, Soltesz KL, Fitzpatrick JW Massive swarm migrations of dragonflies (Odonata) in eastern North America. Am Midl Nat 140: pp 325–342, 1998.

[10] J. Monteil, A. Beghdadi, New interpretation and improvement of the nonlinear anisotropic diffusion for image enhancement, IEEE Transactions on Pattern Analysis and Machine Intelligence 21(9) pp 940–946, 1999.

[11] K.Krissian, S. Aja-Fernandez, Noise-driven anisotropic diffusion filtering of MRI, IEEE Transactions on Image Processing, pp 2265–2274, 2009.

