Volume 3, No. 5, Sept-Oct 2012
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

A Clustering Method Using Simplified Swarm Intelligence Algorithm

Subarna Sinha*
B.E. Final semester student, Department of Computer Science and Engineering,
Tripura Institute of Technology, Narsingarh, Tripura
subarna.sinha9@gmail.com

Asmita Bhaumik
B.E. Final semester student, Department of Computer Science and Engineering,
Tripura Institute of Technology, Narsingarh, Tripura
asmita.6bhaumik@yahoo.com

Aditi Deb
B.E. Final semester student, Department of Computer Science and Engineering,
Tripura Institute of Technology, Narsingarh, Tripura
aditi.deb21@gmail.com

Jhunu Debbarma
Assistant Professor, Department of Computer Science and Engineering,
Tripura Institute of Technology, Narsingarh, Tripura
jhunudb@gmail.com

Abstract: The main aim of clustering is to represent large datasets by a fewer number of segments or partitions. It brings simplicity in modeling data, knowledge discovery and data mining. A family of bio-inspired algorithms, well-known as Swarm Intelligence (SI) has recently emerged that can successfully be applied to a number of real world clustering problems. This paper presents a Bird flocking algorithm that uses the concepts of a flock of agents (birds) moving together in a complex manner with simple local rules. Our algorithm dynamically creates and visualizes groups of data. Each agent/bird representing one data, move with the aim of creating homogeneous groups of data in a 2D environment. This simulation is carried by emulating the natural flocking behavior of birds requiring behavioral rules for cohesion, separation, alignment and avoidance of birds belonging to different community. The birds effectively "think" for themselves and move in a self-organizing manner allowing study of their behavior.

Keywords: Clustering, Swarm Intelligence, Flocking, Collision, Separation, Alignment, Avoidance.

I. INTRODUCTION

The term clustering [1][2] can be defined as dividing a large dataset into a smaller group of set called clusters. The objective of clustering a set of data is to find inherent structure inside the data and to expose this structure as a set of groups. The data objects within each group should exhibit a large degree of similarity while the similarity among different clusters should be [3]. There are a large number of algorithms for discovering natural clusters in a data set, but they are usually implemented in a centralized way [4]. Recent studies have shown that partitioned clustering [5][6] algorithms are more suitable for clustering large datasets. The major problems with these algorithms are that their results are sensitive to the selection of the initial partition and may converge to local optima. To deal with these limitations existing in the traditional partition clustering methods, a number of new document clustering algorithms have been proposed with the inspiration coming from observations of natural processes, which is the area of "Natural Inspired Computing"[7][8] .This area has been further developed into many additional computer science fields, such as swarm intelligence[9][10], evolution computing, Neural Network etc. The main purpose of this work is to show that tools from theoretical computer science might be of benefit to the study of natural algorithms. The flocking cranes, migrating geese, synchronously flashing fireflies are all instances of natural algorithms.[11][12] These algorithms were designed by evolution over millions of years. By and large, their study has been the purview of dynamical systems theory within the fields of zoology, ecology, evolutionary biology, etc.

Autonomous agents are currently being heavily researched across a wide range of fields and it is likely that commercial production of such systems will increase rapidly in the near future. This project provided the opportunity to learn about how autonomous agents work and to gain a greater understanding of their capabilities, functionality and practical applications. There are numerous types of autonomous agents being developed each working in different ways but which are all based on collections of individual units working in unison with a high level of autonomy. We explore the applicability of bio-inspired approach to the development of self-organizing, evolving, adaptive and autonomous clustering techniques, which will meet the requirements of next generation data mining systems, such as diversity, scalability, robustness, and resilience. There are numerous types of autonomous agents being developed each working in different ways but which are all based on collections of individual units working in unison with a high level of autonomy. These include Complex Adaptive Systems, Population-based Adaptive Systems, Swarm Intelligence, Swarm Engineering, Multi-Agent Systems and Self Organizing Systems. We explore the applicability of these bio-inspired approaches to the development of self-organizing, evolving, adaptive and autonomous clustering techniques, which will meet the requirements of next generation data mining systems, such as diversity, scalability, robustness and resilience.

II. RELATED WORKS

Boids is an artificial life program, developed by Craig Reynolds in 1986, which simulates the flocking behaviour of birds. His paper on this topic was published in 1987 in the proceedings of the ACM SIGGRAPH conference [13].

As with most artificial life simulations, Boids[13] is an example of emergent behavior; that is, the complexity of Boids arises from the interaction of individual agents (the boids, in this case) adhering to a set of simple rules. The rules applied in the simplest Boids world are as follows:

☐ **separation**: steer to avoid crowding local flock mates
☐ **alignment**: steer towards the average heading of local flock mates
☐ **cohesion**: steer to move toward the average position (center of mass) of local flock mates.

III. PROPOSED SYSTEM

The first three rules are those observed by Craig Reynolds which are cohesion, separation and alignment, followed by obstacle avoidance and predator avoidance. The fourth rule" OBSTACLE AVOIDANCE" is an extra feature added to "Boids Algorithm-Craig Reynolds"[14][15] by us. Each rule is weighted according to its level of importance. It was felt that separation (not bumping into each other) for instance was more important than cohesion or alignment so separation was given a much higher weight than the others. Avoidance of both obstacles and predators also has a higher weight than cohesion or alignment as it is assumed that to a real bird not flying into obstacles and potentially becoming injured, or indeed being eaten by a predator, are probably of far higher immediate priority than staying with other birds. Each bird has a set of position co-ordinates (x, y) and a set of destination co-ordinates (x, y).

A. Behaviour Rules In Detail:

a. update bird's position and destination:

The procedure which updates the position and destination of each bird calls all behaviors rules which are currently selected and adds the resulting vectors together, along with the bird's current destination and position. Each rule returns a vector of how much the bird's destination should be changed by that particular rule, so simply adding all of these to the birds' current destination very effectively points the bird to its new corrected destination.

ALGORITHM : UPDATE_POSITION_BIRD

a) Procedure Update_Position_Bird
b) If (bird!= different community bird)
c) Flock (bird)
d) Checkbound() //[check the boundary]
e) Checkspeed() //[check the speed]
f) dx= value //value=value returned from cohesion, separation, alignment or predator avoidance rule.
g) dy= value //value=value returned from cohesion, separation, alignment or predator avoidance rule//.
h) New_value_ x= current_value_x + dx;
i) New_value_ y =current_value _y +dy;
j) New position= (New_value_x, New_value_y);
k) End procedure.

b. cohesion rule:

The Cohesion rule brings the birds together into a Flock or group. The birds try to move towards the centre of their group of immediate neighbours.

ALGORITHM : COHESION

a) procedure Cohesion
b) For each bird in the flock
c) if (distance < sight) //sight = sphere around each bird that shows its perception//
d) update bird.position.x
e) update bird.position.y
f) end if
g) dx+ = (bird.position.x- position.x)*weight
h) dy+ = (bird.position.y- position.y)*weight
i) end for
j) end procedure

c. separation rule:

Birds must avoid bumping into each other as they move maintaining a minimum distance between themselves and their neighbours. (also referred to as "collision avoidance"). The bird's distance from very other bird in the flock is calculated individually (the Euclidean distance) and if they are found to be closer to any other bird than the set minimum collision distance".

ALGORITHM: SEPARATION RULE

a) Procedure Separation
b) For each bird in the flock
c) If (distance < space) //space=maximum desired space between two birds in a flock//
d) dx+ = (position.x- bird.position.x)
e) dy+ = (position.y – bird.position.y)
f) End if
g) End for
h) End procedure.

d. alignment rule:

Birds move towards the average destination of their neighbours keeping the flock in alignment and moving together towards the same general heading. The alignment rule calculates the average destination of all birds within a set neighbourhood distance from the original bird (including the destination of the original bird) and returns the average of those destinations.

ALGORITHM : ALIGNMENT RULE

a) Procedure Alignment
b) For each bird in the flock.
c) If distance < sight // sight= sphere around each bird that shows its perception distance
d) dx+ = bird.dx*weight
e) dy+ = bird.dy*weight
f) End if
g) End for
h) End procedure

e. obstacle avoidance rule:

Birds must avoid hitting predators as they fly while still obeying all the other rules of cohesion, separation and alignment. Each time a bird is updated it checks each obstacle in the obstacle array. When an obstacle/predator is found to be within a minimum set distance to a bird, the bird moves away from the object in order to avoid it. This rule has a high weighting as it would be important for real birds not to fly into obstacles/predators and this is reflected in the simulation.

Algorithm : Obstacle/predator_Avoidance

a) Procedure for predator_avoidance algorithm
b) For each bird in the flock
c) If(bird of _different_community&& distance < sight) // sight= sphere around each bird that shows its perception distance//

d) dx+ = position.x – bird.position.x
e) dy+= position.y – bird.position.y
f) End if
g) End for
h) End procedure

f. ***supporting algorithms:***

EUCLIDEAN DISTANCE:

For all the above steering rules the distance between birds or birds and obstacles needs to be calculated. This is accomplished by calculating the Euclidean distance between them. This is a straight forward calculation using well established mathematical formula [13][14]. To calculate Euclidean distance between two points, **(x1, y1)** and **(x2, y2)** in two dimensional space (used in the 2D simulation) the following formula is used :

Distance= $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. Styles named "Paper Title", "Heading 1", "Heading 2", "Heading 3", and "Heading 4" are prescribed in this template.

IV. DESIGN AND DEVELOPMENT

A. implementation platform and language:

Microsoft Visual Studio and the .Net framework was chosen as a development environment implementing C# [15][16]as the chosen language due to some degree of existing familiarity with both and the ready availability of support documentation and tutorials[17]. Visual Studio 2008 also has fully integrated support for test suites and is reputedly the "best of breed" in this type of application being widely used in industry **C#** is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure. C# is intended to be a simple, modern, general-purpose, object-oriented programming language.

V. RESULTS

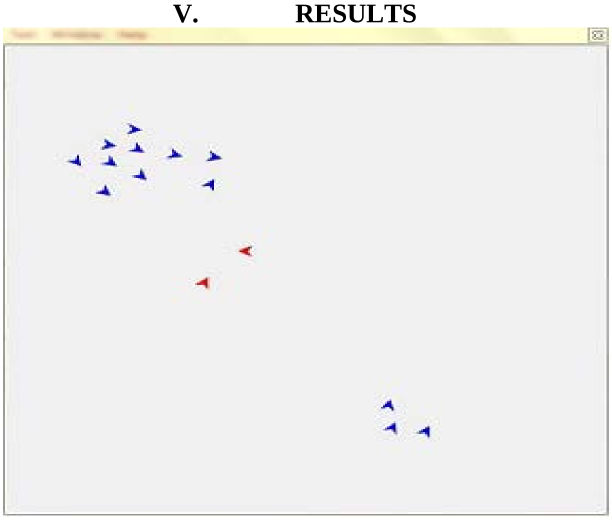

Figure 1 depicts the bird's behavior without cohesion.

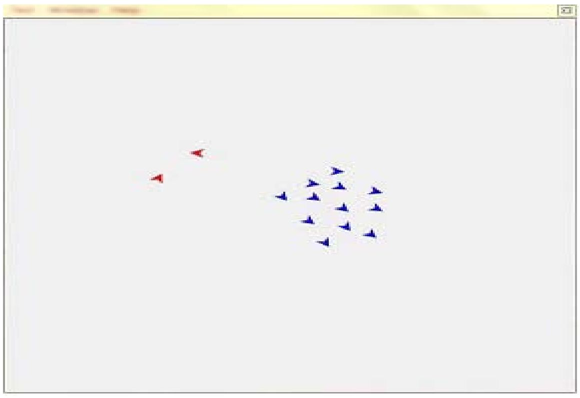

Figure 2 depicts the bird's behavior with cohesion

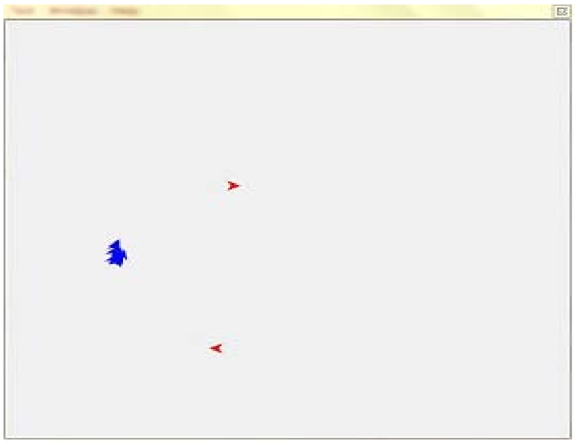

Figure 3 depicts the bird's behavior without separation.

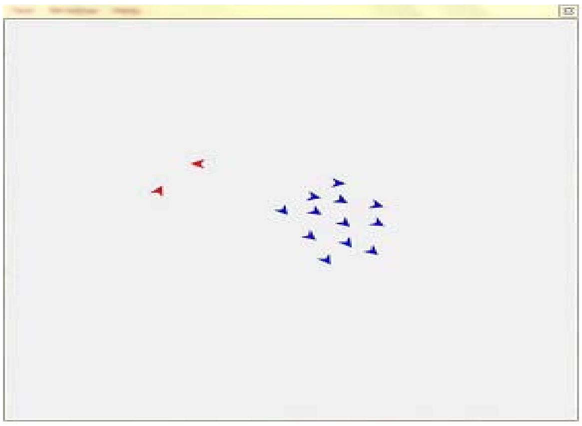

Figure 4 depicts the bird's behavior with separation.

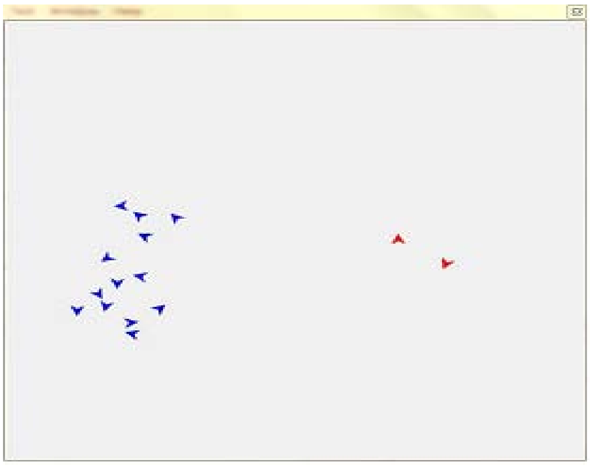

Figure 5 depicts the bird's behavior without alignment.

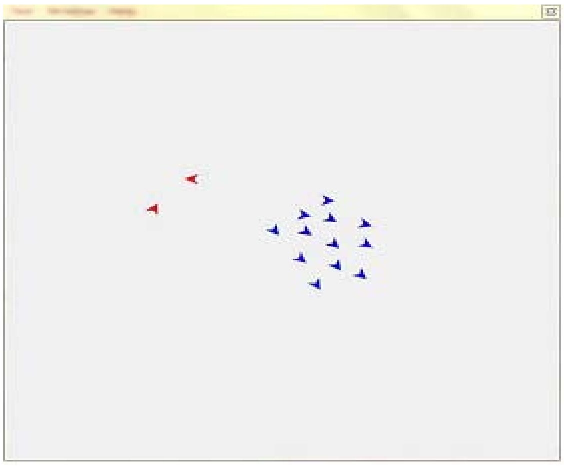

Figure 6 depicts the bird's behavior with alignment.

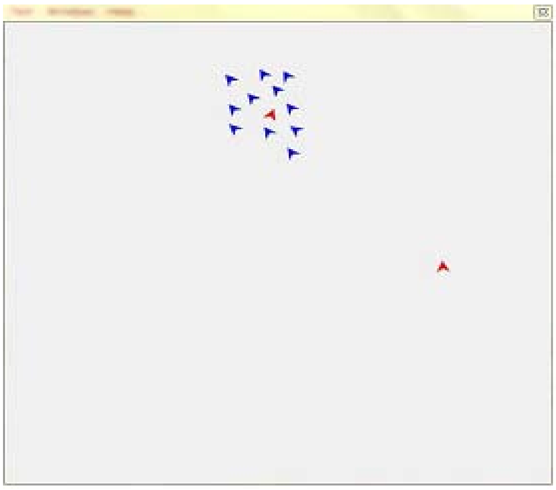

Figure 7 depicts the bird's behavior without avoidance.

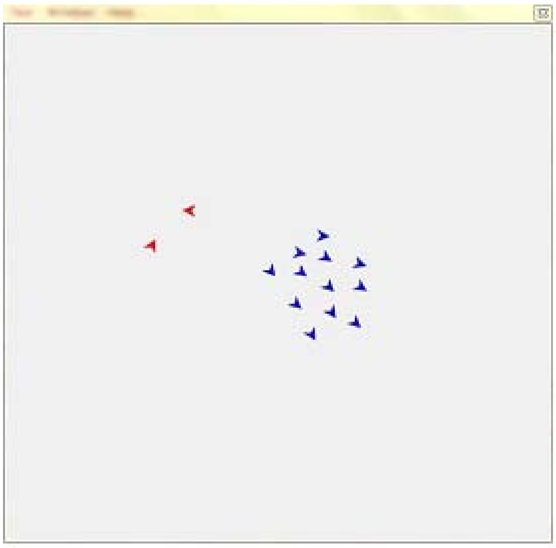

Figure 8 depicts the bird's behavior with avoidance

VI. CONCLUSION

Based on the flocking behavior of birds, we have developed and simulated our algorithm successfully using C#. It aims at partitioning different objects of different groups (in our case, "BIRDS"). In our simulation the basic Bird flocking algorithm is used. From the simulation clustering of object is obtained which is the act of partitioning an unlabeled dataset into groups of similar objects. In the simulation, by implementing avoidance algorithm the two different flocks of birds are getting partitioned and by separation & cohesion algorithm the birds are maintaining a particular distance and a same way of flocking. Thus the sense of clustering is obtained in the proposed algorithm which uses Swarm intelligence. Swarm Intelligence provides us with a powerful new paradigm for building fully distributed de-centralized systems in which overall system functionality emerges from the interaction of individual agents with each other and with their environment. Such systems are intrinsically highly parallel and can exhibit high levels of robustness and scalability; qualities desirable in high-integrity distributed systems. And our project provides a greater edge by using this concept to obtain clustering which provides a better way for software evolution, image segmentation, market research etc.

VII. REFERENCES

[1]. Youssef, S. M., Rizk, M., El-Sherif, M.: Dynamically Adaptive Data Clustering Using Intelligent Swarm-like Agents. International Journal of Mathematics and Computers in Simulation, 1(2), pp. 108-118 (2007).

[2]. Kogan, J.: Introduction to Clustering Large and High-Dimensional Data. Cambridge University Press, University of Maryland, Baltimore (2007).

[3]. Jiawei Han andMicheline Kamber. Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann, September 2000.

[4]. "Kaufman, L., & Rousseeuw, P. J. (1990). Finding groups in data: "An introduction to cluster analysis". New York: John Wiley & Sons.

[5]. Andr´e L. Vizine, Leandro Nunes de Castro, Eduardo R. Hruschka, and Ricardo R. Gudwin. Towards improving clustering ants:An adaptive ant clustering algorithm. Informatica (Slovenia), 29(2):143–154, 2005.

[6]. International Journal of Computer Applications (0975 – 8887) Volume 14– No.1, January 2011, Particle Swarm Optimization: Technique, System and Challenges.

[7]. "Hitchhiker's Guide to Biomorphic Software". The natural world may be the inspiration we need for solving our computer problems. Kenneth N. Lodding, Nasa by Kenneth N Lodding | June 1, 2004 Topic: Bioscience

[8]. E. D. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. In From Animals to Animats: Proc. of the Third Int. Conf.on Simulation of Adaptive Behaviour, pages 501–508. MIT Press, 1994.

[9]. Eric Bonabeau,Marco Dorigo, and Guy Theraulaz. Swarm intelligence: from natural to artificial systems. Oxford University Press, New York, NY, USA,1999.

[10]. Handl, J., Meyer, B.: Ant-based and swarm-based clustering. Swarm Intelligence 1(1), pp.95--113. (2007)

[11]. Grosan C, Abraham A and Monica C, Swarm Intelligence in Data Mining, in Swarm Intelligence in Data Mining, Abraham A, (2006), Grosan C and Ramos V (Eds), Springer, pp. 1-16.

[12]. E. D. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. In From Animals to Animats: Proc. of the Third Int. Conf. on Simulation of Adaptive Behaviour, pages 501–508. MIT Press, 1994.

[13]. Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pages 25–34, New York, NY, USA, 1987.ACM Press.

[14]. A New Clustering Boids Algorithm for Data Mining Marcio Frayze David and Leandro Nunes de Castro Mackenzie University.

[15]. C# 3.0:A Beginner's Guide byHerbert Schildt,Edision 2009, The McGraw-Hill Companies, Page 431.

[16]. Practical C# charts Newand Graphics by Jack Xu, UniCAD Publishing, 30-Mar-2007,Page 383.

[17]. Microsoft C# 2008 by Wrox Box, Publisher John Wiley & Sons, 2009.