Proceeding of 2007 International Conference on Life System Modeling and Simulation (LSMS2007)

A Fish School Clustering Algorithm: Applied to Student Sectioning Problem

Mahmood Amintoosi[1,2], Mahmoud Fathy[1], Naser Mozayani[1], Adel T. Rahmani[1]

[1] *Faculty of Computer Engineering, Iran University of Science and Technology*
[2] *Mathematics Department, Tarbiat Moallem University of Sabzevar, IRAN, 397*
Email:{mAmintoosi, mahFathy,mozayani,rahmani}@iust.ac.ir

***Abstract*— In this paper a new clustering algorithm based on the fish school behavior is proposed. The algorithm is an extension of the classical flock model of Reynolds with a new characteristic. We have different kinds of fishes: leader fish and follower fish. In addition it is supposed that our artificial fishes live in some predefined caves. The school is a set of groups, which every group has a leader and lie in a cave in every time. Other members of the group are leader followers. Regarding the concepts of joining and splitting of the groups, a clustering algorithm which is almost similar to hierarchical clustering algorithms is proposed. The proposed algorithm is applied to student sectioning problem, a sub problem of timetabling problem. Simulation results show the applicability of the proposed algorithm.**

***Index Terms*— Artificial life, Fish School, Flock, Clustering, Student sectioning, educational timetabling, fuzzy evaluation**

I. INTRODUCTION

Ants' colonies, flocks of birds, termites, swarms of bees etc. are agent-based insect models that exhibit a collective intelligent behavior (swarm intelligence) that may be used to define new distributed clustering algorithms. In these models, the emergent collective behavior is the outcome of a process of self-organization, in which insects are engaged through their repeated actions and interaction with their evolving environment [1].

The flocking algorithm was originally proposed by Reynolds as a method for mimicking the flocking behavior of birds on a computer both for animation and as a way to study emergent behavior. Flocking is an example of emergent collective behavior: there is no leader, i.e., no global control. Flocking behavior emerges from the local interactions. Each agent has direct access to the geometric description of the whole scene, but reacts only to flock mates within a certain small radius. The basic flocking model consists of three simple steering behaviors: separation, cohesion and alignment. Separation gives an agent the ability to maintain a certain distance from others nearby. This prevents agents from crowding too closely together, allowing them to scan a wider area. Cohesion gives an agent the ability to cohere (approach and form a group) with other nearby agents. Steering for cohesion can be computed by finding all agents in the local neighborhood and computing the average position of the nearby agents. The steering force is then applied in the direction of that average position. Alignment gives an agent the ability to align with other nearby characters. Steering for alignment can be computed by finding all agents in the local neighborhood and averaging together the 'heading' vectors of the nearby agents [1].

Some researchers assumed a leader's existence and presumed it directed the movement of the whole flock, and some others don’t agree with this idea [2]. In the proposed method we assume that the whole flock contains several groups. Each group has a leader and other members of the group follow him. The leader considered as the prototype of his group. Every fish is correspondence to one data point. Fishes according to their similarity forms groups. As in real flock and fish schools, which they sometimes split apart to go around an obstacle [3], we shatter a group in some situations. Joining and shattering of the groups is the core of our algorithm. We tested our algorithm with student sectioning problem, which arises in some timetabling problems.

The course timetabling problem essentially involves the assignment of weekly lectures to time periods and lecture room in such a way that a set of constraints satisfy. Many works related to this problem exist [4,5,6]. A particular problem related to timetabling is student sectioning problem (SSP) [7,8,9,10,11,12,13]. This problem is due to courses, which involve a large number of students. For some reasons splitting these students to a few smaller sections is desirable:

1. Room capacity requirements: when number of students in a course is greater than every room capacity.
2. The policies of Institute: some institutes have rules about maximum capacity of courses (e.g. 50 for specialized courses and 60 for public courses).
3. A good student sectioning may reduce the number of edges in the conflict matrix.

In our previous works [7,8], we concentrate on initial student sectioning prior to timetabling and introduced a fuzzy sectioning method for SSP. For demonstrating the applicability of the proposed fish school clustering algorithm, this problem has been chosen. The results are compared with two famous clustering methods: fuzzy c-means and k-means.

The reminder of this paper organized as follows: The proposed method is explained in section II. Section III deals with an application of the proposed method with simulation results. Finally section IV describes the conclusion.

II. The proposed Fish School Clustering Algorithm

The proposed method is based on three assumptions:

1) Existence of leaders in flocks.
2) Joining groups of individuals.
3) Group shattering.

In the following subsections we briefly review the above observations, explain how each item used in our algorithm. The last subsection is about to the proposed algorithm.

1) Existence of leaders in flocks

Some researchers assumed a leader's existence and presumed it directed the movement of the whole flock [2, 14], and some others don't agree with this idea [2, 3]. Bajec [2] assumed two types of flock: leader flock and leaderless flock. Leader flock considered as a flock with at least one leader (i.e. animate that is not influenced by any of its flockmates, but influences at least one), whereas a leaderless flock is a flock with no leader (i.e. each animate is influenced by at least one of its flockmates). [2].

In the proposed algorithm we considered two types of fishes: leaders and followers. We define a leader fish, a fish that is not influenced by any of its flockmates. It is something like Bajec's leader animate. A follower fish is a fish, which is influenced by at least one of its flockmates.

2) Joining groups of flockmates.

Base on the Peterson's article [15] when a number of individuals are initially placed at random in the environment, they quickly aggregate into several groups, generally consisting of four to five individuals. Over time these groups will themselves aggregate, if they are confined within a bounded area. The other individuals in the aggregation follow each other and the leader. Over time leadership of the group will switch from one individual to another. A common leadership change occurs when a group leader sees another group, and begins to follow one of that group's members, bringing the two groups together. Over time, as aggregations move and encounter one another they will often change their forms. Figure 1 shows some forms of the groups.

In the proposed method we used such aggregation to join the groups. Because the leader fishes have a major role in our algorithm, and carousel structure doesn't have any leader, we avoid such groups. Thus our groups will have tree shapes. This implies that we can benefit usual data representation and search methods for tree structure as those introduced in data structure courses.

3) Group shattering

The jostling between members will occasionally become intense enough to split an aggregate into two or more parts. When two groups meet, they will either join, shatter into smaller groups, or most commonly, avoid one another and continue intact [15]. Also Reynolds in his pioneer paper [3] mentioned that real flocks sometimes split apart to go around an obstacle. Sparrows might flock around a group of obstacles that is in fact a herd of elephants. Similarly, behavioral obstacles might not merely be in the way; they might be objects of fear such as predators. It has been noted that natural flocking instincts seem to be sharpened by predators.

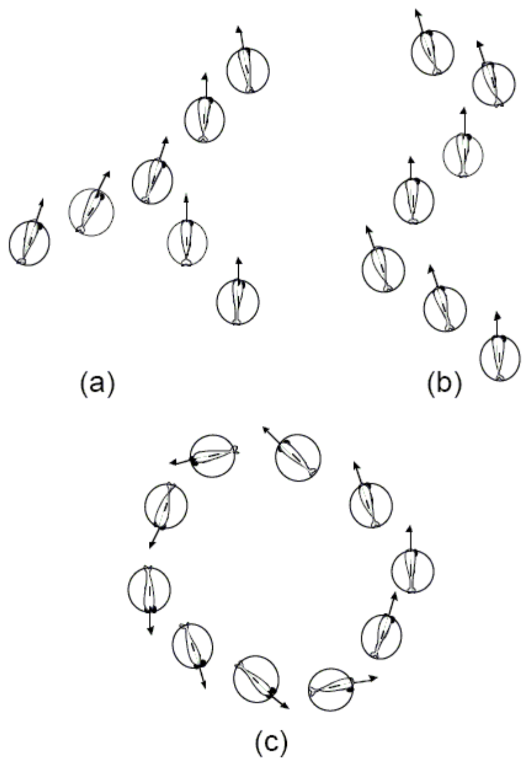

Fig. 1. a) An example of a 'Y' formation. b) An example of a 'Z' formation. c) An example of a carousel structure [15].

In the proposed method, each cluster considered as a cave with a predefined capacity, which each group can goes into them. When a leader fish enters to a cave, all of his followers follow him and go into the mentioned cave, unless the cave reaches to its maximum capacity. In this case, the group will be breaks to small pieces and the remaining followers, as sub-trees form some other smaller groups with their roots as their leaders. They remain on their previous cave. Scanning a group as a tree for finding the next fish, which can go into the new class has been done with a breadth first search method.

4) The proposed fish school clustering

In the previous subsections we saw the subparts of our algorithm. The main idea of the proposed algorithm is borrowed from Erik Rasmussen's friendly agent [16], which he demonstrated a java applet containing some agents which act together with some simple rules. Each friendly agent looks at all the other agents he can see and randomly selects one to be his friend. However, he can only make another agent his friend if the other agent does not already have the agent selected as his friend. The friendly agent then tries to move toward his friend by accelerating towards him. The agent remembers who his friend is and continues to accelerate towards him as long as the friend remains visible. If the agent ever loses sight of his friend he simply selects another one at random given the selection criteria stated above. If ever a friend can-not be found the agent accelerates in a random direction to search for one.

Here we summarize our algorithm based on the previous subsections and the Rasmussen's friendly agent's rules:

```
//Fish School Clustering algorithm
Initializing:
  - maxClustSize: maximum allowed cluster
    size
  - Label each fish's class randomly.
  - Mark each fish as a leader fish.
Repeat
1-Each leader fish A looks at all fishes
  he can see and select the nearest fish
  (B) as his leader. He can only make
  another fish his leader, if B does not
  already followed A, directly or indi-
  rectly (for preventing carousel
  structure).
2-If a leader found, A follows the leader
  by setting its class label and his
  followers' class labels to this new
  leader's class label, in a breadth first
  search manner. Mark A as a follower.
3-Before changing a follower's class label,
  it is checked whether the size of the new
  class exceeds maxClustSize or not. If it
  exceeds, the labels of the remaining
  followers of A, will remain unchanged,
  and the headings fishes mark as leaders
  (shattering).
4-The fish remembers who his leader is.
Until the labels remain unchanged in two
consecutive iterations or reaching to
maximum Iteration.
```

As each cluster considered as a cave with predefined maximum capacity, it is obvious that the above algorithm is suitable for those clustering applications, which, the number of clusters and maximum capacity allowed for each cluster are known in advance. In the next section we will show the applicability of the proposed algorithm with an example.

III. Experimental Results: Application of the Proposed Method to Student Sectioning Problem

In this section at first we briefly review the SSP problem and then show how this problem can be solved with the proposed algorithm.

1. SSP definition

The aim of the Student Sectioning problem is to allocate students of a course to smaller sections for satisfying the following criteria:

a) Student course selections must be respected.
b) Section enrollments should be balanced, i.e. all sections of the same course should have roughly the same number of students;
c) Section capacities and policies of institute should not be exceeded.
d) Student schedules in each section would be the same as each other (as much as possible).

Each student has a feature vector. The courses taken by each student are its feature elements, which is represented with a bit array. Suppose that P is the number of all courses and *Vi* is the list of taken courses by student *i*. As shown in equation 1, *Vi* is the feature vector of the i^{th} object (student).

$$V_i = \left(V_{i_1}, \ldots, V_{i_P}\right)$$
$$V_{i_j} = \begin{cases} 1; \text{ if student } i \text{ has taken lesson } j \\ 0; \text{ otherwise} \end{cases} \quad \textbf{(1)}$$

2. Fish School Clustering applied to SSP

In fact students in our universities have behaviors like our artificial fishes. It is usual in our universities that a group of students have had same course schedules; and often the members of each group are influenced by a specified person. Hence we suppose that each student is a fish in our algorithm. As we see in previous section, we need a meter for measuring similarity or distance of two students. The total number of common courses of two students is defined as their similarity measure.

The maximum capacity of each room is dictated by the college and hence the number of sections (clusters or coves) will be in hand by dividing the total number of students taken course X by the maximum number of students allowed in each section.

3. Simulation Results

The information used for simulation, are taken from Mathematics department at Sabzevar University. Students and courses are randomly selected with the following characterization:

- Total number of students: 210.
- Number of courses: 38.

The simulator program has been written with MATLAB on a Windows platform.

For comparison purposes we need a clustering evaluation function. We do this by a fuzzy evaluation function. Here our used courses for running the algorithm have to split into 2 sections. Rates of section balancing and similarity of students' schedules in each section, and similarity between clusters (as a negative parameter) are its inputs and its output is the clustering performance. Three lingual variables 'N1PerN2' (N1/N2), 'Density' and 'ClustersDistance' are defined as inputs of a fuzzy inference engine. 'N1PerN2' represents the section's balancing rate. It is supposed that N1 is the size of the smaller section and hence, the range of this variable is between 0 and 1. Since the number of students in each section should be equal as possible, the suitable values of N1PerN2 are close to 1. Density of each cluster is the sum of the common courses of all its containing students' pairs, divided by total common courses between all students' pairs. ClustersDistance is defined in a manner like that Density, but for each student pair (x,y), x and y have to be belonging to different sections. Figure 1 shows the membership functions of the mentioned variables.

The output of the fuzzy inference engine is named 'Performance' and has the following values: Bad, NotBad, Medium, Good and Excellent. A subset of our total 27 fuzzy rules is as follows:

Fuzzy Rules:

Rule1: if (Density is High) and (N1PerN2 is Suitable) and (ClustersDistance is Good) then (Performance is Excellent)

Rule2: if (Density is High) and (N1PerN2 is Middle) and (ClustersDistance is Good) then (Performance is Good)

…

Rule27: if (Density is Low) and (N1PerN2 is UnSuitable) and (ClustersDistance is bad) then (Performance is Bad)

Figure 2 shows the simulation results for 5 selected courses. Each clustering method was run 50 times for each course. As can be seen the proposed method is always better than K-means, and except for one course, its performance is equal or better than fuzzy C-means.

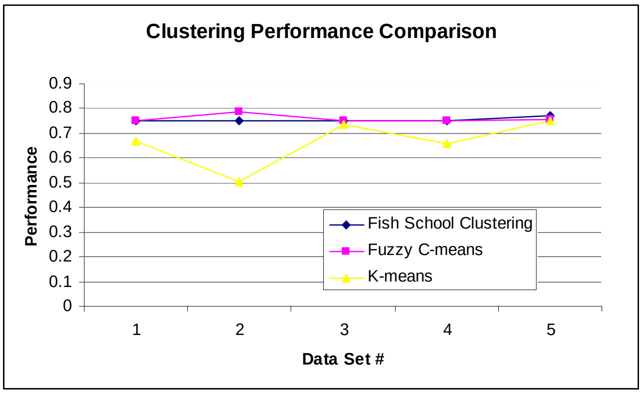

Fig. 2. Comparison of the proposed fish school clustering with fuzzy c-means and k-means clustering methods.

IV. Concluding Remarks

In this paper, we present a prototype using a new algorithm based on the concepts of a school of fishes that move together in a complex manner with simple local rules. Each data point considered as a fish. Several fishes, according to their similarity, which have a leader, form a group. The whole school contains several groups. The leader acts as the prototype of his group. Joining of groups can be done when leader of one group closes enough to a member of another group and follows him. It is sup-posed that our artificial fishes live in some caves. Each cave is correspondence to one cluster and has a predefined capacity. Shattering of a group occurs when a leader tries to go into another cave, but the cave doesn't have enough capacity for all of his followers. The group will be shattered and the remaining followers will stay on their previous caves. Some of them will be newer leaders. Joining and shattering the groups in an iterative manner partitions the input data points into homogenous clusters. The simulation results of the proposed algorithm applied to student sectioning problem confirms its applicability.

References

[1] A. Augimeri, G. Folino, A. Forestiero and G. Spezzano, "A multidimensional flocking algorithm for clustering spatial data", Proceedings of the 7th WOA 2006 Workshop From Objects to Agents, Catania, Italy, 2006.

[2] I.L. Bajec, "Fuzzy Model for a Computer Simulation of Bird Flocking", PhD Thesis, University of Ljubljana,Faculty of Computer and Information Science, 2005.

[3] C.W. Reynolds, "Flocks, herds and schools: A distributed behavioral model", SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, New York, NY, USA pp. 25-34, 1987.

[4] E.K. Burke and P. Causmaecker (eds.), The 4th International Conference on the Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science 2740. Springer-Verlag, 2002.

[5] E.K. Burke and M. Trick (eds.), The 5th International Conference on the Practice and Theory of Automated Timetabling. Lecture Notes in Computer Science 3616, Springer-Verlag, 2005.

[6] E.K. Burke and H. Rudová (eds.), Proceedings of The 6th International Confer-ence on the Practice and Theory of Automated Timetabling, Faculty of Informatics, Masaryk University, Brno, The Czech Republic, 2006.

[7] M. Amintoosi, H. Sadooghi Yazdi and J. Haddadnia, "Fuzzy Student Sectioning", Burke, E.K., Trick, M. (eds.): The Practice and Theory of Automated Timetabling V, pp. 421-425, 2004.

[8] M. Amintoosi and J. Haddadina, "Feature Selection in a Fuzzy Student Sectioning Algorithm", Burke EK and Trick M (eds.) The 5th International Conference on the Practice and Theory of Automated Timetabling. Lecture Notes in Computer Science 3616, Springer-Verlag, pp. 147-160, 2005.

[9] S.M. Selim, "Split Vertices in Vertex Colouring and Their Application in Developing a Solution to the Faculty Timetable Problem", The Computer Journal, Vol. 31, No. 1, pp. 76-82, 1988.

[10] A. Hertz, "Tabu search for large scale timetabling problems", European Journal of Operational Research 54, pp. 39–47, 1991.

[11] J. Aubin and J.A. Ferland, "A large scale timetabling problem", Computers and Operations Research 16, pp. 67–77, 1989.

[12] G. Laporte and S. Desroches, "The Problem of Assigning Students to Course Sections in a Large Engineering School", Computers and Operational Research, Vol. 13, No. 4, pp. 387-394, 1986.

[13] C. Beyrouthy, E.K. Burke, J.D. Landa-Silva, B. McCollum, P. McMullan and A.J. Parkes, "The Teaching Space Allocation Problem with Splitting", Proceedings of The 6th International Conference on the Practice and Theory of Automated Timetabling, pp. 103-122, 2006.

[14] I.D. Couzin, J. Krause, N.R. Franks and S.A. Levin, "Effective leadership and decision-making in animal groups on the move", Nature.433 pp. 513-516, 2005.

[15] G.D. Peterson, "Animal aggregation: experimental simulation using vision-based behavioral rules", Lectures in complex systems, MA, Addison-Wesley, pp. 623-630, 1993.

[16] http://www.erik-rasmussen.com/