Fuzzy Student Sectioning
(Extended Abstract)

Mahmood Amintoosi[1], Hadi Sadooghi Yazdi[2], Javad Haddadnia[3]

[1] Mathematics Department,
Sabzevar Teacher Training University, IRAN, 397
amintoosi@sttu.ac.ir,
[2,3] Engineering Department,
Sabzevar Teacher Training University, IRAN, 397
haddadnia@sttu.ac.ir

In this paper a new student sectioning method based on fuzzy clustering is presented. One of the less studied sub-problems of timetabling is student sectioning. This problem is due to courses, which involve a large number of students. We concentrated on initial student sectioning to create a section conflict graph prior to timetabling. Our aim is to allocate students to course sections so as to satisfy the following criteria:

1. Student course selections must be respected.
2. Section enrollments should be balanced, i.e. all sections of the same course should have roughly the same number of students;
3. Student schedules in each section would be the same as each other, as much as possible.

In the proposed method, at first, with a Fuzzy C-means algorithm, students in large classes have been classified; then this clustering is evaluated with a fuzzy function, according to some criteria: clusters centers distance, clusters density, and size of clusters. Each student has a feature vector in fuzzy classifier. Taken curses of each student are its features. Based on the proposed fuzzy function and with an exhaustive search, the best features are selected. The best classification of students is corresponds with these selected features. The simulation results show that this method has fewer conflicts with respect to an entrance year based sectioning.

Most previous works related to the course scheduling problem has concentrated on timetable construction with little regard to student sectioning. Lewandowski [1] present a method of solving the problem that cycles between building student schedules and a timetable for courses.. Selim [2] introduced the idea of split vertices and start made to determine those vertices, which should be split in order that the chromatic number may be reduced. (S)he decreases the chromatic number of the conflict matrix, from 8 to 3. Thus the total number of periods needed reduced. Aubin and Ferland [3] first generate an initial timetable with an assignment of the students to the course sections; then an iterative procedure is used which adjusts the timetable and the grouping successively until no more improvement of the objective function can be obtained. Hertz [4] used a tabu search technique for both timetabling and sectioning problems. He assumed that the numbers of students in each section are fixed. The neighborhood N(s) of a solution s consists of all those grouping which can be obtained from s by exchanging the two students of two different section of a course. Laporte and Desroches [5] also take into account the student sectioning.

In the following paragraphs the proposed method has been explained briefly.

Without loss of generality, the number of clusters assumed to be 2. For data representation a bit array has been used. Each student has a feature vector in fuzzy classifier. Taken lessons list of each student are its features and is represented with a bit array.

Feature selection has an important role in classification problems. It is summarized as follows: Given a number of features, how can one select the most important of them so as to reduce their number and at the same time retain as mush as possible of their class discriminatory information? [6]. In addition to dimensionality reduction, a good feature selection may increase the quality of clustering.

In each class at least one lesson is common between all of its students. Removing it should not influence the clustering results. In our problem it seems that removing the following lessons is a good idea:

1. Lessons that *the most* students have taken.
2. Lessons that *the least* students have taken.

Removing those lessons that none of students have taken, is done in a preprocessing stage. One problem that arises is specifying the appropriate thresholds for *the most* and *the least* values. We found them with an exhaustive search procedure. The pseudo-code illustrated in Fig.1. shows the overall procedure and feature selection method.

```
AllLessons = Set of all lessons that students of this lesson have
     taken; //(All Features)
List1 = Percentage of students that take each lesson, sorted in
     non increasing order;
Thresholds1 = Distinct elements of List1;
for i:=1 to length(Thresholds1) do
begin
   MaxSet= Set of Lessons that percentage of students that take
        them > Thresholds1[i];
   List2 = Percentage of students that don't take each lesson,
        sorted in non increasing order;
   Thresholds2 = Distinct elements of List2;
   for j:=1 to length(Thresholds2) do
   begin
      MinSet= Set of Lessons that percentage of students that
          don't take them > Thresholds2[j];
      SelectedFeatures = AllLessons – (MaxSet ∪ MinSet);
      Clusters=FuzzyC-Means(SelectedFeatures);
      if ClusteringEvaluation(Clusters) is better than previous
         clusters then
      begin
         BestClusters = Clusters;
         BestThreshold1 = Thresholds1[i];
         BestThreshold2 = Thresholds2[j];
      end;  // end if
   end;  //end for j
end;  // end for i
```

Fig. 1. Overall procedure and feature selection method. FuzzyC-Means() is a fuzzy classifier and ClusteringEvaluation() is a fuzzy inference engine.

In the feature selection procedure, `BestThreshold1` and `BestThreshold2` are the appropriate values for *the most* and *the least* thresholds, respectively. In fuzzy clustering, those lessons that the percentage of students taken them is greater than `BestThreshold1` or less than `BestThreshold2`, would be remove. Simulation results for a class with total 35 lessons, showed that sometimes even 4 lessons are sufficient for clustering.

Fuzzy C-means (FCM) is a data clustering algorithm that each data point is associated with a cluster through a membership degree. Most analytical fuzzy clustering approaches are derived from Bezdeck's FCM [7,8]. Fuzzy C-Mean algorithm has been used for student sectioning. After clustering with a pair of thresholds, the produced clusters are evaluated and those clusters correspond with BestThresholds are selected as the best clustering form and the problem solution.

Evaluation of each produced clustering, is done with a fuzzy function. As mentioned earlier, we would like to create a section conflict graph that leads to a timetable with fewest possible conflicts, but we have no timetable at all when we start. For this reason, in the evaluation function the following in hand quantities are used to satisfy our mentioned criteria:

1. Size of cluster1 / size of cluster2: for satisfying criteria no. 2 (Section enrollments should be balanced). The number of sections can be computed with dividing the number of students with the section bound.
2. Between-class distance.
3. Cluster density: this item satisfies the third criteria (Student schedules in each section would be the same as each other).

Fuzzy classification satisfies criteria no.1. The above parameters have been used in a fuzzy inference engine as linguistic variables for clustering evaluation. We named them: N1PerN2, and Density. Generally the number of students in each section of a course should be as equal as possible. Thus it is suitable to N1PerN2 be as near as possible to 1.

Sum of two cluster's density is considered as Cluster Density. Density of each cluster is sum of common taken lessons by its students.

The output of the fuzzy inference engine is named **Performance** and has the following values: Bad, NotBad, Medium, Good and Excellent. Some of total 12 used rules are as follows:

If (Density is High) *and* (N1PerN2 is Suitable) **then** (Performance is Excellent)
If (IDChanging is High) *and* (N1PerN2 is Middle) **then** (Performance is Good)
If (Density is Low) *and* (N1PerN2 is UnSuitable) **then** (Performance is Bad)

The information used for simulation, are taken from more than 200 students schedules at Mathematics department of our university. For comparison purposes, sectioning is done in two ways: the proposed method and a method based on the entrance year of students that is like with the manual manner. Table 1. shows the number of students in 3 lessons before and after sectioning with two methods.

Table 1. Number of students in 3 lessons before and after sectioning with two methods.

Lesson Name	Number of students before sectioning	Density and Number of students in each cluster after sectioning						
		Sectioning based on entrance year			Proposed method			
		First Section	Second Section	D	First Section	Second Section	D	SF/TF

Teaching Mathematics1	74	38	36	3320	38	36	3622	24/35
Algebra 1	67	53	14	3902	33	34	3379	20/32
Numerical Analyses 1	52	32	20	1845	26	26	1871	27/32

In Table 1, D indicates Density of clusters and SF shows the number of selected features and TF is the number of total features. As can be seen in Table 1 balancing of the sections is very nice and clusters densities in the proposed method are better than the year based method, except for Algebra1. But with the year based sectioning, the sections sizes of this lesson are not suitable.

Conclusion - In this paper a new student sectioning method based on fuzzy clustering is presented. Our aim is to allocate students to course sections so as to satisfy some criteria. The simulation results showed that the proposed method minimizes the density of the section conflict graph by clustering students with similar schedules such that section balancing is also satisfies. One can use the proposed algorithm in cycles between building student schedules and a timetable for courses.

Acknowledgment – The authors wish to thank the anonymous referees for their valuable suggestions and comments.

References

[1] G.Lewandowski, "Course Scheduling: Metrics, Models, and Methods", lewan@xavier.xu.edu, 1996.

[2]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.

[3] J. Aubin, , J.A. Ferland, “A large scale timetabling problem”, *Computers and Operations Research* 16, pp. 67–77, 1989.

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

[5] G. Laporte, 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.

[6] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.

[7] J. C. Bezdek , *Pattern Recognition with Fuzzy Objective Function Algorithms*, plenum, New York, 1981.

[8] L. Bob Rowski, J. C. Bezdek, “C-means Clustering with the L_1 and L_∞ Norms”, *IEEE Trans. on Syst.Man. and Cybern*, Vol. 21, No.3, pp. 545-554, 1991.