Media Informatika, Vol. 5, No. 2, Desember 2007, 79-86
ISSN: 0854-4743

GENETIC ALGORITHM WITH CENTER BASED CHROMOSOMAL REPRESENTATION TO SOLVE NEW STUDENT ALLOCATION PROBLEM

Zainudin Zukhri[1], Khairuddin Omar[2]
[1]*Jurusan Teknik Informatika, Fakultas Teknologi Industri,Universitas Islam Indonesia*
Jl. Kaliurang Km. 14 Yogyakarta 55584
E-mail: zainudin@fti.uii.ac.id
[2]*Jabatan Sains dan Pengurusan Sistem, Fakulti Teknologi dan Sains Maklumat,*
Universiti Kebangsaan Malaysia
43600 UKM Bangi Malaysia
E-mail: ko@ftsm.ukm.my

ABSTRACT

Genetic Algorithm (GA) is one of the most effective approaches for solving optimization problem. We have a problem difficulty for GA in clustering problem. It can be viewed as optimization problem, that is maximization of object similarity in each cluster. The objects must be clustered in this paper are new students. They must be allocated into a few of classes, so that each class contains students with low gap of intelligence and they must not exceed the class capacity. The intelligence gap of each class should be low, because it is very difficult to give good education service for the students in the class whose high diversity of achievements or high variation of skills. We call this problem as New Student Allocation Problem (NSAP). Initially, we apply GA with Partition Based Chromosomal Representation (PBCR). But experiments only provide a small scale case (200 students and 5 classes with same capacities). Then we try to apply GA with Center Based Chromosomal Representation (CBCR) and we evaluate it with the same data. We have successfully improved the performance with this approach. This result indicates that chromosomal representation design is the important step in GA implementation. CBCR is better than PBCR in all aspects. All classes generated by CBCR approach have largest gap of intelligence in each class less than generated by PBCR. CBCR approach can reduce these values almost a half of the values with PBCR approach.

Keywords: center based, partition based, Genetic Algorithm, similar student

1. INTRODUCTION

Genetic Algorithm (GA) is one of the most effective approaches for solving optimization problem, but we should be carefull to handled it for solving clustering problem. Clustering problem can be viewed as optimization problem, that is maximization of object similarity in each cluster. The real clutering problem is very complicated, because the larger the number of objects, the harder to find the optimal solution and furthermore, the longer to reach a reasonable results. Therefore, it is a NP-hard problem.

The objects must be clustered in this paper are new students. They must be allocated into a few of classes, so that each class should contain students with intelligence level as similar as possible and the number of students in each class must not exceed the capacity. In other words, the classes should contain students with low gap of intelligence. Clustering of students is an important matter, because it very difficult to give good education service for students in large number whose high diversity of achievements (Vanderhart 2006) or high variation of skills (Wiedemann 2006). With the students allocated to the groups, discriminating policies to these groups can be implemented easily (Ma et al. 2000).

Universities or schools, usually ignore this problem and they distribute new students into their classes at randomly. It is can make an educational problem. To avoid this problem, new students must be clustered with a suitable method. For a while, there is sorting-score method which allocates the new students into a few of classes based on their achievements. Actually, in a normal distribution of students scores, this method is an improper method because it only produces a smartest class and a weakest class that have a good student similarity and the other classes have high student dissimilarities. According to Statistics, the sorting method is not one of clustering methods (Jain et al. 1999). It is reasonable if this method should not be utilized to cluster new students.

For solving this problem we cannot use any clustering methods directly, and we should modify them. It is caused by some differences between clustering new students and general clustering problem. In clustering new students, the number of objects (students) in each cluster (class) cannot be determined based on the result of clustering process, but it is determined before clustering process. In addition, dissimilarity between each class can be ignored in clustering new students. Hence implementation of clustering methods needs modification.

2. RELATED WORK

Wright (2001) takes the view that students allocation problem is a type of constrained multi-dimensional bin packing problem, with students being ″items″ to be packed and the classes being ″bins″. His view can be applied, if objective of the problem is to minimize the number of classes. Since the objective is to minimize the gap of intelligence in each class, student allocation problem should be viewed as clustering problem rather than bin packing problem. It is viewed by Susanto et al. (2002) who have used Fuzzy C-Means algorithm (FCM) for solving this problem. In their experiments, they cluster students of certain subject based on their scores of prerequisite subjects. It is a good work, but it has not shown the advantage of FCM yet, because it only involves 20 students.

There are some statistical approaches available to solve clustering problem, those are Agglomerative Methods (AM), the most popular statistical approach for clustering problem (Cole 1998). But we cannot apply them to solve this problem directly and we should modify them. Experimental study shows that the performance of this modification is depend on the data distribution. In normal distribution, it generates classes with the largest intelligence gap is growing proportional with the clustering sequence (Zukhri & Omar 2006). Therefore the first class has the lowest gap, and the last class has the highest gap.

Cole (1998) has used GA for solving general clustering problem. He applied GA to cluster any objects so that each cluster has high dissimilarities with other clusters and each cluster contains similar objects. His idea about chromosomal representation and GA operators is very good to be applied. But we cannot apply all of his works to solve NSAP, because the most important matter in NSAP is the student similarity in each class and capacity of class. Hence the students dissimilarities between classes can be ignored in NSAP. Beside of that, chromosomal representation in his works does not enough to represent the capacity of each classroom. However, it inspired our previous research to modify his work for solving NSAP (Zukhri & Omar 2007).

Initially, we supposed that NSAP can be solved by GA with permutation chromosomal representation, that is one of Cole's chromosomal representation. This representation is very simple and the chromosome represents the distribution of new students in each class directly. We call it as Partition Based Chromosomal Representation (PBCR), because this representation separates a chromosom into some sub-chromosomes based on the capacity of class. But comparison of the experimental results with AM in our previous research shows that the average of the largest intelligence gap by GA is greater than by AM. It means that in all classes, the performance of PBCR is not better than AM. It gives us a motivation to find other approach for improving the GA performance. In this paper, we propose an alternative approach to handle it.

3. RESEARCH METHODOLOGY

We assume that attributes of new students to be used as the criteria on clustering them are their admission test scores. These scores are represented as integer numbers between 0 and 100. We develop the proposed approach, that is GA with Center Based Chromosomal Representation (CBCR) as software that can cluster n students with m attributes (dimensions) into n_c classes. To evaluate the effect of chromosomal representation substitution, we use the same 2-dimensional data used in evaluation of PBCR approach (Zukhri & Omar 2007) and finally we compare the results between PBCR and CBCR.

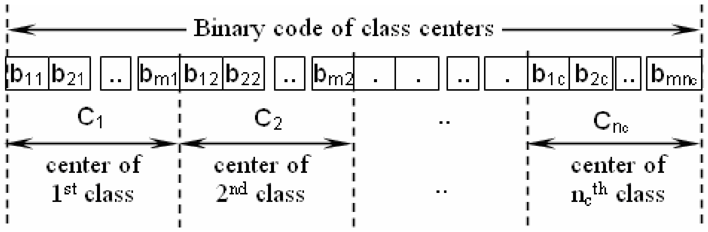
4. GENETIC ALGORITHM WITH CENTER BASED CHROMOSOMAL REPRESENTATION

4.1 Chromosomal Representation

The center based chromosomal representation is binary representation. To cluster n new students into n_c classes, chromosomal representation is designed as follows:

- A chromosome consists of n_c sub-chromosome. The i^{th} sub-chromosome is representation of a student as the center of i^{th} class. It consists of m genes or bits.
- A student should be a member of a class, but he/she probably becomes the center of two or more classes.

The chromosomal representation is shown in Figure 1.



Binary code of class centers
b11 b21 .. bm1 b12 b22 .. bm2 . . .. . b1c b2c .. bmnc
C1
C2
..
Cnc
center of 1st class
center of 2nd class
..
center of nc th class

Figure 1. Center Based Chromosomal Representation

Chromosomes which consist of n_c sub-chromosomes will be generated randomly with binary representation in the initial generation step. To generate its, we should determine width of sub-chromosome. Based on this value, the mathematic formulation of the class centers can be determined too. It is the rules to determine them:

- If there is an integer m so that number of new students equals to 2^m, then the width of sub-chromosome is m. The index of student as the center of the ith class is

$$C_i = \sum_{k=1}^{m} b_k 2^{m-k} + 1. \tag{1}$$

- If there is no integer m so that number of new students equals to 2^m, then the width of sub-chromosome is the lowest m which $2^m > n$. The index of student as the center of the ith class is

$$C_i = int\left((n \sum_{k=1}^{m} b_k 2^{m-k}) / 2^m \right) \tag{2}$$

The centers of classes should be converted to distribution of new students with algorithms as follows:

```
1. j ← 1
2.   member[j] ← 0
3.   j ← j + 1
4.   if j ≤ $n_c$ then goto 2
5. i ← 1
6. j ← 1
7.    find the nearest student g from center[j] who has not
          clustered
8.    if member [j] ≤ Kuota[j] then
9.         cluster the student g into class[j]
10.        i ← i + 1
11.        member[j] ← member[j] + 1
12.    j ← j + 1
13.    if (j ≤ $n_c$) and (i ≤ $n_c$) then goto 7
```

After converting the centers to the distribution, the value of objective function can be calculated for evaluation.

4.2 Initialization

In CBCR, initialization is an easy process because it only requires binary generator. There are three parameters will handle this step: population size (*popsize*), number of classes (n_c) and number of new students (n). Clustering n new students into n_c classes requires *popsize* chromosomes.

Each chromosome in the initial population can be generated with an algorithm as follows:

```
1. j ← 1
2.   gene[i] ← random(0,1)
3.   i ← i + 1
4.   if i ≤ $n_c \cdot m$ then goto 2
```

4.3 Objective Function and Fitness Function

In our previous research, the objective function is minimization of total of the largest intelligence gap in each class. Now it is changed as minimization of the largest gap of intelligence in all classes as follows:

$$h(x) = min\left(max\left(d_i(g_a, g_b) \mid i = 1..n_c\right)\right), \tag{3}$$

where $1 \leq a \leq q_i$, $1 \leq b \leq q_i$, $a \neq b$ and d is the Euclidean distance between the students g_a and g_b. We change objective function with this function in order to make the largest gap of intelligence in each class is as same as possible. For fitness function, we use the same fitness function with PBCR approach (see Zukhri & Omar 2007 for detail).

4.4 Crossover

We use Uniform Crossover (UX). In UX, each gene of the children is created by copying the corresponding gene from one or the other parent, chosen according to a random generated binary crossover mask of the same length as the chromosomes. Where there is a 1 in the crossover mask, the gene is copied from the first parent, and where there is a 0 in the mask the gene is copied from the second parent. A new crossover mask is randomly generated for each pair of parents. Children, therefore contain a mixture of genes from each parent (Syswerda et al. 1989). An illustration of UX is shown in Figure 2.

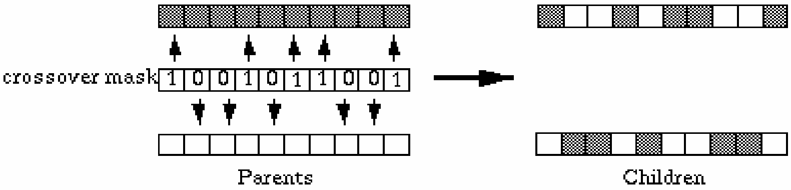


Figure 2. Uniform Crossover (UX)

4.5 Mutation

For binary representation, mutation is a simple operator. It is only inverting the value of gene randomly (Gen & Cheng 2000). It can be illustrated as shown in Figure 3.

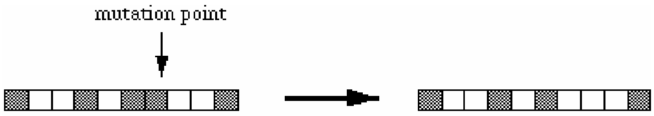


Figure 3. Mutation

5. RESULT AND DISCUSSION

We evaluate the proposed approaches with the same 2-dimensional data used in evaluation of PBCR approach. In this data, there is a pair of students whose the largest intelligence gap. This gap equals to 60.14. Among the students, there are 9 pairs of students have same scores.

Experiments have been performed for each combination of the following parameters:

- Crossover probability: 0.25 and 0.50.
- Mutation probability: 0.01, 0.05 and 0,1.

We follow the suggested parameters of GA by Holland (1975).

The PBCR approach trapped to local optima. We evaluated it with all combination of GA parameter and we expect that it can improve the best solution. But the changing still make GA trapped to local optima. The bad performance of GA with PBCR approach is shown by the best known solution is not so better than the objective function of initial population. It is shown in Table 1.

Table 1. Solution found by PBCR approach

Class index, i	*Objective function of initial population*	*Best known solution*
1	52.43	46.04
2	51.09	47.89
3	50.91	52.20
4	52.33	46.62
5	52.62	48.92
Average	51.88	48.33

The bad performance of PBCR approach indicates that it is not suitable for the problem. It is very sensible because the chromosomal representation makes the searching space more wide than the real problem. The searching space of GA in this approach depends on the number of students (width of chromosome), and does not depend on the number of classes at all. For 200 students, the searching space is factorial of 200, it is much greater than the total way to cluster 200 students into five classes with same capacities (200! >> 200! / 40!5).

The experimental study shows that performance of CBCR approach is better than the PBCR approach. The performance comparison between PBCR and CBCR results is shown in Table 2. PBCR approach reaches best known solution with population size equals to 300, cross over probability 75% and mutation probability 1%. Meanwhile CBCR approach reaches best known solution with population size equals to 40, cross over probability 30% and mutation probability 1%. We run the two approaches until 200 generations.

Table 2. Comparison performance between PBCR approach and CBCR approach

Class index, i	*Objective function of initial population*	*Best known solution*
1	46.04	28.16
2	47.89	25.24
3	52.20	24.76
4	46.62	23.41
5	48.92	27.86
Average	48.33	25.89

Table 2 shows that CBCR approach is better than the PBCR approach in all aspects. All classes generated by CBCR approach have largest gap of intelligence less than generated by the PBCR approach. CBCR approach can reduce this values almost a half of it generated by PBCR approach. It is shown by comparison of its average: 25.89 and 48.33. CBCR approach is also can reduce the searching space. If PBCR approach with population size equals to 300, then CBCR approach only with population size equals to 40. It is mean that PBCR generates 60,000 chromosomes, but CBCR approach generates only 8,000 chromosomes.

6. CONCLUSION

Chromosomal representation is an important part of Genetic Algorithm implementation. It should be defined carefully. PBCR in permutation representation is not suitable for solving New Student Allocation Problem, because this approach was trapped to the local optima. This representation makes GA has no ability to solve the problem. In other hand, it is easy to solve New Student Allocation Problem by CBCR in binary representation. Experimental study shows that PBCR to local optima, because the genetic operators are not effective to produce a population in the next generation whose better fitness functions. It is caused by total number of chromosome can be generated by GA is much larger than the real problem. Meanwhile CBCR can reduce largest gap of intelligence in each class and it can also reduce the searching space.

REFERENCES

Cole, R.M. (1998). Clustering with Genetic Algorithms. *Master Thesis*. University of Western Australia.

Gen, M. and Cheng, R. (1997). *Genetic Algorithms and Engineering Design*. Canada: John Wiley & Sons, Inc.

Holland, J.H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor.

Jain, A.K., Murty, M.N. and Flynn, P.J. (1999). Data clustering: a review. *ACM computing surveys, 31*(3): 264–323.

Ma, Y., Liu, B., Wong, C. K., Yu, P.S. and Lee, S. M. (2000). Targeting the right students using data mining. *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Datamining KDD*, 457-464.

Susanto, S., Suharto, I. and Sukapto, P. (2002). Using Fuzzy Clustering Algorithm for Allocation of Students." *Transaction on Engineering and Technology Education, 1*(2), 245-248.

Syswerda, G., (1989). Uniform crossover in genetic algorithms. In Glover, F., and Kochenberger, G.A., *Handbook of Metaheuristics. Kluwer Academic Publishers*, New York.

Vanderhart, P. G. (2006). Why do some schools group by ability? *American Journal of Economics and Sociology, 65*(2), 435.

Wiedemann, T. (2000). A Virtual Textbook for Modeling and Simulation. *Proceedings of the 2000 Winter Simulation Conference*, 1660-1665.

Wright, M. (2001). Experiments with a Plateau-Rich Solution Space. *Proceedings of the 4th Metaheuristics International Conference*, 317-320.

Zukhri, Z. and Omar, K. (2006). Modification of Agglomerative Methods to Cluster New Students into Their Classes. *Proceedings of the 1st International Conference on Mathematics and Statistics*, 493-498.

Zukhri, Z., and Omar, K. (2007). Comparative Evaluation of Genetic Algorithm and Modification of Agglomerative Method in New Students Allocation Problem. *Proceedings of Application of Information Technology National Seminar*, B9-B12.