Far East Journal of Mathematical Sciences (FJMS)
Volume 77, Number 1, 2013, Pages 1-15
Published Online: June 2013
Available online at http://pphmj.com/journals/fjms.htm
Published by Pushpa Publishing House, Allahabad, INDIA

HETEROGENEOUS GROUPING BY GENETIC ALGORITHM FOR STUDENT COALITION: A CASE STUDY

Anon Sukstrienwong

Information Technology Department
School of Science and Technology
Bangkok University
40/4 Rama IV Rd., Klong-Toey
Bangkok, Thailand
e-mail: anon.su@bu.ac.th

Abstract

Forming student groups has long been considered as an effective approach for collaboration work in the university. It involves balancing several students with different levels of knowledge in the appropriate groups to learn from others. In terms of fairness of student's educational skills among formed groups, however, forming groups of students is becoming difficult when the number of student is big and the heterogeneity of students is more complex. In this paper, we present an approach, called heterogeneous grouping by genetic algorithm for (HGGA), to generate student formation based on heterogeneous grouping. Our algorithm aims to achieve fairness among generated groups and compares the result with a random self-selecting method made by students. A case study is also exemplified in

2010 Mathematics Subject Classification: 20.

Keywords and phrases: corporative learning, genetic algorithm, heterogeneous grouping, student group formation.

Submitted by K. K. Azad

Received October 19, 2012

this study which was performed with a group of 85 students to demonstrate the scalability and ability of the approach. Based on our case study and experiment, the results are illustrated that the performance of HGGA is efficient at distributing and mixing heterogeneous students in the established groups.

1. Introduction

In the cooperative learning, several approaches of learning and teaching rely on a concept that allows students to work in groups, which are called *group coalition*. It involves assigning individual member to the right group in order to enhance student interaction, social experiences and relationship. However, gaining individual student knowledge and achievement in groups is much more important [1]. Researches in several areas especially in the education have presented that learning with other members in a group improves the student's learning knowledge, as it is enable individual student to learn from others. It is observed by several researchers that students gain well benefits of cooperative working in group. Therefore, at present, there exist several theories and methods of collaborative learning allowing the students to obtain the knowledge in groups, such as group discussion, peer learning and coaching, and team leading. Additionally, there are several papers studying the way of grouping students with abilities and knowledge in school as illustrated in [2] and [3]. As shown in [6], Swing and Peterson found that students of mixed ability gain better knowledge than homogeneous groupings. Based on Martin and Paredes [7], they have affirmed that heterogeneous groups are better in a broader range of tasks. Educators firmly advise that a teacher must combine students of a variety of skills in one group; it is a way to help most students in the group to learn from heterogeneous friends. When each individual student succeeds, the whole group also succeeds.

Nevertheless, a result derived from a paper in [5] presents that in general most students were in the favor of working in a homogeneous group. Therefore, in most classes, if teachers allow students to form groups by themselves, all formed groups will more likely to be homogeneous. This method may cause unequal opportunities for all students to learn. Consequently, most teachers prefer assigning the students randomly to the group because it is a quick and efficient way in order to ensure some heterogeneity in groups. Notwithstanding, this typical technique remains a problem for most teachers because the task can be time-consuming if the list of students in a class is large [8]. It can be more complex than it seems to be if the teachers set up the student groups based on heterogeneous grouping. This is because the groups are constructed with the assumption that a formed group comprises students whose prior educational skills and knowledge should be unequal. On the contrary, fairness and equity of the whole students are concerned; the formed groups are similar in all attributes. This measuring takes time, but it is always worth the effort. In real classes, there are several attributes that can be involved to build the heterogeneous groups of students based on the course objectives, such as previous academic performance, the grade point average (GPA), student attitude, and preferences. Thus, searching for an optimized group of the whole students is an exhaustive search.

In this paper, the algorithm for student coalition formation is presented based on fairness of heterogeneous groups. The heuristic search algorithm called *genetic algorithm* (GA) is used in order to search for optimized student coalition. The paper is separated into five parts including this introduction section. Section 2 provides the conceptual framework of the proposed algorithm. Then genetic algorithm for HGGA is described in Section 3. The experimental results of our case study are demonstrated in Section 4. The conclusions and upcoming works are illustrated in Section 5.

2. Conceptual Framework of HGGA

Every student has different knowledge background because everyone has individual information and experiences [10]. However, we can classify students into three simple types based on their prior knowledge, which are good students, moderate students, and low students. To allocate these heterogeneities of students to appropriate groups, we define the mathematical term and its definition as follows. A class contains n students, denoted by $S = \{S_1, S_2, ..., S_n\}$. Each student contains exactly m attributes, which are prior grades, student preferences, and pretest or exercise scores. It can be represented in an m-dimensional vector, denoted by $A_i = (a_{i1}, a_{i2}, ..., a_{im})$, where a_{im} is the value of attribute m of student i. For example, student S_i, the 2-dimensional attribute vector is represented as $A_i = (a_{i1}, a_{i2})$. The values of attributes have various kinds in values. For instance, if the attribute represents the grade received of a prerequisite course, its value ranges from 0-4.0. In this research, every individual student belongs exactly to one group. No other groups contain the same student. The algorithm tries to divide students into smaller groups with the same size as possible. In certain cases, we cannot construct groups with the same size. If n students are divided into p smaller groups, then the size of the group is $\left\lfloor \frac{n}{p} \right\rfloor \le size \le \left\lceil \frac{n}{p} \right\rceil$. Let G denote the whole groups of students, $G = \{G_1, G_2, ..., G_p\}$. Then

$$| G | = \sum_{i=1}^{p} | G_i | = | S | = n.$$

The algorithm generates groups of heterogeneous students. Then, a group i contains k students, denoted by $G_i = \{g_{i1}, g_{i2}, ..., g_{ik}\}$, where each member is mapped to one element of S. Let a mapping function of a group's member g_{ik} be denoted by $f(g_{ik}) = S_m$, where $S_m \in S$ and $1 \le m \le n$. When the group completely generated, each attribute of the formed group is the average value of group's members. Therefore, the property of G_i is associated with the vector of $(V_{i1}, V_{i2}, ..., V_{im})$, where m is the number of student's attributes. The attribute value of V_{im} can be calculated by the average value of all members belonged in the group, which is shown in (1):

$$V_{im} = \frac{a_{f(g_{i1})} + a_{f(g_{i2})} + \cdots + a_{f(g_{ik})}}{k}, \tag{1}$$

where $a_{f(g_{ik})}$ is the attribute of $f(g_{ik}) = S_m$, $1 \le m \le n$. To achieve the mechanism for fairness and equity, it is necessary to compute followings values.

1. If two students called s_i, s_j belong to the same group, then the distance between students is evaluated by the Euclidean distance (ED_1) as seen in (2):

$$ED_1 = | s_i - s_j | = \sqrt{\sum_{t=1}^{m} (a_{it} - a_{jt})^2}, \tag{2}$$

where a_{it} is the value of attribute t of student i.

And the graph in Figure 1 shows the distance of two students called s_i and s_j, where two-attribute vector is applied

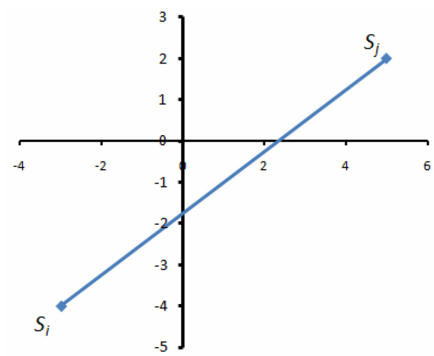

Figure 1. The distance of students s_i and s_j, where attribute vector of S_i is $A_i = (a_{i1}, a_{i2})$ and attribute vector of S_i is $A_j = (a_{j1}, a_{j2})$.

Adapted from (2), the Euclidean distance of the group G_q is calculated by summing up all values of ED_1 as the following equation:

$$\begin{aligned} ED_1(G_q) &= \sum_{d=1}^{k-1} \sum_{e=d+1}^{k} | g_{qd} - g_{qe} | \\ &= \sum_{d=1}^{k-1} \sum_{e=d+1}^{k} \sqrt{\sum_{t=1}^{m} (a_{f(g_{qd})t} - a_{f(g_{qe})t})^2}, \end{aligned} \tag{3}$$

where k is the size of group G_q.

If a group G_q has higher $ED_1(G_q)$ than other groups, then it implies that the group G_q has mixed with different skills of students more than others. For example, a group called G_q contains three students named g_{q1}, g_{q2}, and g_{q3}, where $f(g_{q1}) = S_1$, $f(g_{q2}) = S_2$, and $f(g_{q3}) = S_3$. The calculation of $ED_1(G_q)$ is presented below:

$$
\begin{aligned}
ED_1(G_q) &= \sum_{d=1}^{2} \sum_{e=d+1}^{3} |\, g_{qd} - g_{qe} \,| \\
&= |\, g_{q1} - g_{q2} \,| + |\, g_{q1} - g_{q3} \,| + |\, g_{q2} - g_{q3} \,| \\
&= \sqrt{\textstyle\sum_{t=1}^{m} (a_{f(g_{q1})t} - a_{f(g_{q2})t})^2} \\
&\quad + \sqrt{\textstyle\sum_{t=1}^{m} (a_{f(g_{q1})t} - a_{f(g_{q3})t})^2} \\
&\quad + \sqrt{\textstyle\sum_{t=1}^{m} (a_{f(g_{q2})t} - a_{f(g_{q3})t})^2} \\
&= \sqrt{\textstyle\sum_{t=1}^{m} (a_{S_1t} - a_{S_2t})^2} + \sqrt{\textstyle\sum_{t=1}^{m} (a_{S_1t} - a_{S_3t})^2} \\
&\quad + \sqrt{\textstyle\sum_{t=1}^{m} (a_{S_2t} - a_{S_3t})^2}.
\end{aligned}
$$

2. If two groups called G_i, G_j are formed, the distance between groups is evaluated by the following equation:

$$ED_2 = |\, G_i - G_j \,| = \sqrt{\sum_{k=1}^{m} (V_{ik} - V_{jk})^2}, \tag{4}$$

where V_{ik} is the attribute k of group i, and m is the total number of attributes.

If two-attribute vector is applied, the two-attribute vector of G_i is denoted by (V_{i1}, V_{i2}). Each element of the vector is derived by the average value of all members in the group which is previously presented in (1). For instance, the graph of group G_i and G_j is exhibited in Figure 2, and the Euclidean distance between of two groups is illustrated below:

$$ED_2 = | G_i - G_j | = \sqrt{(V_{i1} - V_{j1})^2 + (V_{i2} - V_{j2})^2}.$$

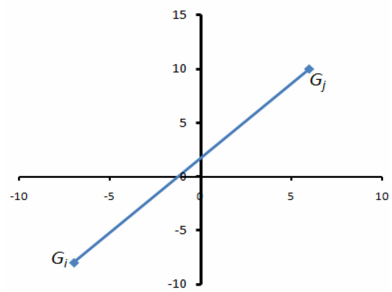

Figure 2. The distance of group G_i and G_j, attribute vector of $G_i = (V_{i1}, V_{i2})$ and attribute vector of $G_j = (V_{j1}, V_{j2})$.

3. Developing the Algorithm for Heterogeneous Grouping

3.1. Problem encapsulation and fitness functions

To generate the genetic algorithm, we start by encrypting the problem into a fixed-length character string of chromosome. In our problem, a set of n student denoted by $S = \{S_1, S_2, ..., S_n\}$, therefore the length of a chromosome equals n. Each element of the chromosome represents a group that the student belongs to. The chromosome structure of our problem is expressed in Figure 3. An element of the chromosome can be assigned to any of $\{G_1, G_2, ..., G_p\}$. For instance, if a set of students are $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$ separated to form three smaller groups, the smallest group that comprises $\left\lfloor \frac{7}{3} \right\rfloor = 2$ students, and the biggest group can make up of $\left\lceil \frac{7}{3} \right\rceil = 3$ students. Suppose nine students are divided to establish 3 groups as $G_1 = \{S_2, S_3\}$, $G_2 = \{S_1, S_5, S_7\}$, and $G_3 = \{S_4, S_6\}$. Then the chromosome of this formation can be encoded as shown in Figure 4.

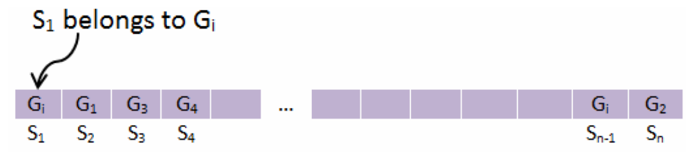

Figure 3. The chromosome's structure for HGGA.

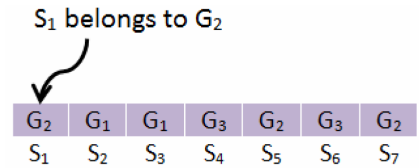

Figure 4. The chromosome representing $G_1 = \{S_2, S_3\}$, $G_2 = \{S_1, S_5, S_7\}$, and $G_3 = \{S_4, S_6\}$.

Generally, GAs are a class of evolutionary algorithms motivated by natural science [16]. They search the space of possible chromosomes in an attempt to find good solutions based on fitness function. The well-defined fitness measure helps solving this problem to research goal of this paper. And, the purpose of our proposed algorithm is to achieve the mechanism for fairness and equity of heterogeneity in each group and between groups, so multi-objective fitness functions are defined as follows.

1. First fitness function of HGGA

It is related to Euclidean distance of groups presented in (3). Hence, average value of all groups is illustrated in (5):

$$f_1(chromosome) = \frac{\sum_{q=1}^{p} ED_1(G_q)}{p}, \tag{5}$$

where p is the total number of formed groups.

As f_1 is the average value of all groups, this function is able to guide the method to search for optimal solution. If f_1 is high, then it indicates that most groups contain different level skills of students. In order to reach the heterogeneity of students, we construct groups consisting of students that are not of the same kind. Therefore, we expect to produce groups with high value of f_1.

2. Second fitness function of HGGA

This function is designed to help the algorithm remain the mechanism of fairness and equity among groups. It is involved with (4), as it is the Euclidean distance between of two groups. If there exist *p* groups created by the algorithm, the second fitness function is demonstrated below:

$$f_2(chromosome) = \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} | G_i - G_j |$$

$$= \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \sqrt{\sum_{k=1}^{m} (V_{ik} - V_{jk})^2}, \tag{6}$$

where V_{ik} is the attribute *k* of group *i*, and *m* is the total number of attributes.

If the value of f_2 is low, then it points that among formed groups are more balanced with heterogeneous students. Consequently, our algorithm focuses on the student groups with low value of f_2.

In theory, a multi-objective fitness function can be represented of a set of *n* objectives, where each objective is associated with its own attributes. Therefore, we use both (5) and (6) to build the final fitness function for our algorithm illustrated below:

$$f(x) = f_2(x) - f_1(x), \tag{7}$$

where *x* is the selected chromosome.

3.2. HGGA operators

Once we have decided what chromosome structure and fitness measure will look like, we then design HGGA operators to create new offspring. Keep in mind that our algorithm works on a fixed-length characters string. If we have n students to create groups, the length of chromosome is n. The flowchart of algorithm is presented in Figure 5. There are two operators used in the algorithm. The first operator is a selection operation. It just duplicates an existing population to the next generation. And, the second operator is a self-crossover operator. It is the most significant operator because it produces new chromosomes, which are different from their parents. Randomly chosen parents in the current population reproduce with each other in order to yield offspring for the next generation [11]. Unlike the conventional crossover operator, this operator combines two mechanisms of crossover and mutation. One parent of the current population is picked in a complete random. Next, the algorithm randomly selects two parts of the chosen parent and creates one offspring. Two portions of the selected chromosome are entirely interchanged. By doing this, the algorithm can remain the size of each group equally. As a result, new-born offspring will be valid for the next generation. However, the offspring are required measuring the fitness value. In applying the crossover operator, there is a small value called *crossover probability* (p_c) to control a ratio of how many parents will be picked for generating offspring. As the crossover operator is primarily responsible for the search of new offspring, it may be possible to produce a bad individual. The example of the self-crossover operator applying on a chromosome is described in Figure 6.

In a single generation, these operations will be functional to create a new population for the next generation. The initial population is originated by a complete random. Conversely, in our approach, the initial population of chromosomes must be qualified in some regulations, such as the total number of groups and the group size. In each generation, outperform population is selected, which is based on their fitness measure in (7). The algorithm repeatedly runs until the termination criterion is satisfied or it reaches the maximum number of generations. Finally, the best solution may be found.

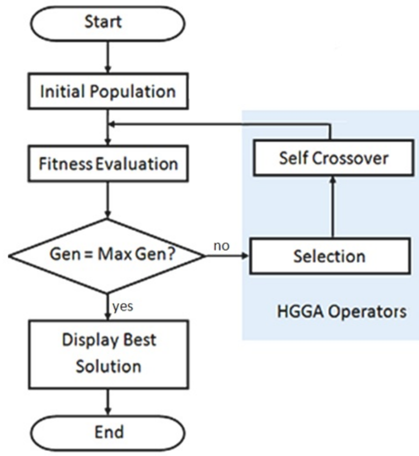

Figure 5. Flowchart of HGGA.

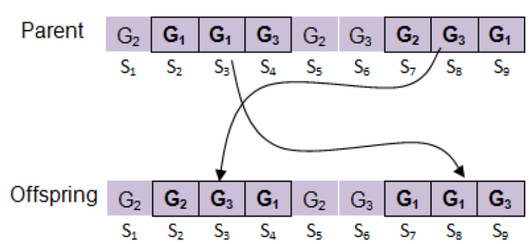

Figure 6. Self-crossover operator.

4. Experimentation and Results

To measure the grouping efficiency of our algorithm, we conducted a case study of 85 students who took a class of IT350 in academic year of 2011 at Bangkok University. A list of students' attributes, such as required courses and grade of the prerequisite courses which were set to organize heterogeneous grouping of students. After all information had been well prepared, we had run the program several times to see which initial values of parameters would direct the algorithm's search to the best solution. Based on the experimentation, several parameters for controlling the algorithm, such as the initial population size (M), the maximum number of generations (Gen), and the crossover probability (p_c), are found as illustrated in Table 1.

Table 1. Parameters setting for HGGA

Constants	Detail	Range
M	Population size	500
Gen	Maximum number of generations	1000
p_c	Crossover probability	0.65

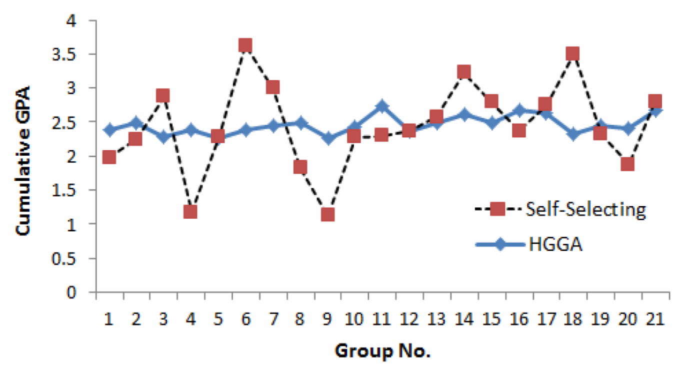

Figure 7. Average cumulative GPA of formed groups.

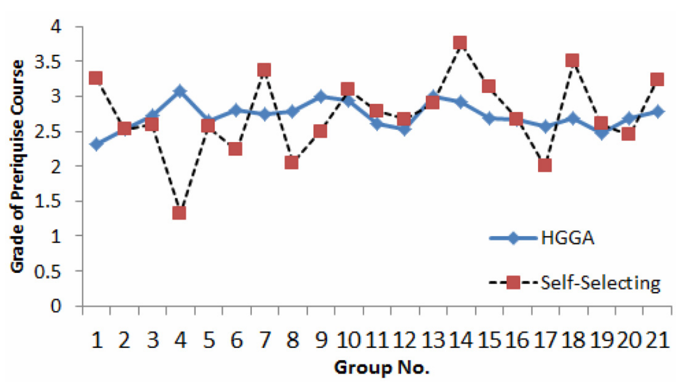

Figure 8. Average grade of a prerequisite course of formed groups.

We compared our algorithm with the manual grouping method, which was made manually by students themselves. Based on reaches done by both Wang et al. [12], and Rau and Heyl [13], we decided to create groups of four students because the small group gives more opportunities for each student to work [14, 15]. Anyway, it depends on the nature of the course. Since a set of 85 students registered in IT350 course is a case study of our experiment, the algorithm generates 20 groups of four students and a group of five students. As mentioned earlier, the algorithm aims to form group with equality and fairness by balancing dissimilar students into the formed groups equally. The results of our proposed algorithm are displayed in Figure 7 and Figure 8.

In Figure 7, for all formed groups generated by the HGGA, the graph shows that the average cumulative GPA in each group spans in a similar level, which is 2.29 to 2.73. It implies that different students; top students, moderate students, and low students; are distributed and mixed among groups equally. Nevertheless, if students are manually grouped together by their own, the formed groups will be noticeably different. And, it causes of unfairness among groups. As we can see form the cumulative GPA of this self-selecting approach, it declares a very wide range in grade from 1.19 to 3.62.

Based on the curriculum of computer science in Bangkok University, IT310 computer programming is the prerequisite class of IT350. For that reason, at the same time of generation student groups, IT310 is added into the student's attribute. The result associated to this prerequisite course is demonstrated in Figure 8. The graph represents the grade point average of IT310, which is about 2.25-3.00. However, it is even worse when students are grouped by the self-selecting method, as the average value of each group is in a much wider range, which is about 2.00-3.25. It is because students are more likely to arrange a group with the same personality and skill.

5. Conclusions and Future Work

This paper presents an algorithm, called *HGGA* for creating heterogeneous grouping of students by using GAs. It aims to generate student groups based on their prior education at different levels of knowledge related to the current course. The suitable fitness function helps searching the optimal student formation based on heterogeneous grouping. The quality of the constructed groups made by the HGGA is compared with the self-selecting method. According to our experiment, the empirical results of the case study clearly display that the HGGA is more efficient than the manual self-selecting method. The algorithm is able to balance different students while fairness and equity among dissimilar groups are highly concerned. As the result, we hope that the students who were assigned to the appropriate group are potential to improve their ability to learn from others.

In the future, we will continue to enhance our approach for most datasets to support the cooperative learning in the university. Moreover, the algorithm will be compared to other algorithms as well. Some other attitudes and student abilities will be included in the fitness function to achieve more fairness and equity among students.

References

[1] G. Renner and A. Ekárt, Genetic algorithms in computer aided design, Computer-Aided Design, Elsevier Science Ltd. 35(8) (2003), 709-726.

[2] G. V. Glass, Grouping students for instruction, A. Molnar, ed., School Reform Proposals: The Research Evidence, Research in Educational Productivity Series, Greenwich, CT: Information Age Publishing, 2002.

[3] L. Tach and G. Farkas, Learning-related behaviors, cognitive skills, and ability grouping when schooling begins, Social Science Research 35(4) (2006), 1048-1079.

[4] M. Sapon-Shevin, B. Ayres and J. Duncan, Cooperative learning and inclusion, J. S. Thousand, R. A. Villa and A. I. Nevin, eds., Creativity and Collaborative Learning: The Practical Guide to Empowering Students, Teachers, and Families, 2nd ed., Paul H. Brookes, Baltimore, 2002, pp. 209-222.

[5] S. O. Adodo and J. O. Agbayewa, Effect of homogenous and heterogeneous ability grouping class teaching on student's interest, attitude and achievement in integrated science, International J. of Psychology and Counselling 3(3) (2011), 48-54.

[6] S. R. Swing and P. L. Peterson, The relationship of student ability and small group ability and small group interaction to student achievement, American Educational Research J. 19(2) (1982), 259-274.

[7] E. Martin and P. Paredes, Using learning styles for dynamic group formation in adaptive collaborative hypermedia systems, The First International Workshop on Adaptive Hypermedia and Collaborative Web-based Systems (AHCW 2004), pp. 188-198.

[8] G. Harrison, S. Griffin and L. Broughton, The Innovation Bank: Project Group Formation and Much More, 10th Annual Conference of the Subject Centre for Information and Computer Sciences, 2009, pp. 44-49.

[9] X. Hu and R. C. Eberhart, Multiobjective optimization using dynamic neighborhood particle swarm optimization, Proc. IEEE Congr. Evol. Comput., Honolulu, HI, 2002, pp. 1677-1681.

[10] M. Oğuz Kutlu, Developing a scale on the usage of learner control strategy, Academic J. Educational Research and Reviews 7(10) (2012), 244-250.

[11] R. Devi Priya and S. Kuppuswami, Genetic algorithm based approach for imputing missing discrete attribute values in databases, WSEAS Transactions on Information Science and Applications 9(6) (2012), 169-178.

[12] D.-Y. Wang, S. L. Sunny and S. Chuen-Tsai, DIANA: A computer supported heterogeneous grouping system for teachers to conduct successful small learning groups, Computers in Human Behavior 23 (2007), 1997-2010.

[13] W. Rau and B. S. Heyl, Humanizing the college classroom: Collaborative learning and social organization among students, Teaching Sociology 18 (1990), 141-155.

[14] L. Savova and R. Donato, Group Activities in the Language Classroom, English Teaching Forum 24(2) (1991), 12-15.

[15] D. Brown, Principles of the Language Learning and Teaching, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1980.

[16] R. David and K. Eswar Prasad, Genetic algorithm approach to optimize the process parameters in roller burnishing operation, International J. Materials Engineering and Technology 2(2) (2009), 177-205.