

A Genetic-algorithm Approach for Balancing Heterogeneous Groups of Students

Anon Sukstrienwong

Bangkok University, 40/4 Rama IV Rd., Klongtoey, Kluaynamthai, Bangkok, Thailand.

Abstract: *In the context of cooperative learning, students in classrooms tend to learn more by sharing their experiences and knowledge. In addition, a diversity of educational backgrounds can be used to build heterogeneous groups of students. Therefore, in this paper, we propose an approach called Genetic Algorithm for Group Formation (GAGF) for the group composition regarding prior educational knowledge in order to ensure that several students at different levels of knowledge are distributed optimally within the group formation. The main aim of the algorithm is to balance the heterogeneous groups of students equally. The proposed algorithm mimics the natural process of a genetic algorithm in order to achieve optimal solutions. Based on our experimental results, it indicates that the algorithm improves the quality of the group formation of heterogeneous students leading to better solutions.*

Keywords: *Cooperative learning, Genetic Algorithm, Heterogeneous Grouping.*

1. Introduction

Cooperative learning is an approach in which students work together in small groups in order to achieve a common academic goal. Some researchers, such as Huang and his fellow [1] have focused on improving cooperative learning for students. In the work of Balmeceada et al. [2], they claimed that different characteristics of members might influence the group performance. In addition, they have considered three characteristics of students to form groups: psychological styles, team roles, and social networks. Paredes et al. [3] proposed a technique for integrating grouping mechanisms through each student's learning style. In this research, all students were required to answer a questionnaire in order to determine their learning style before generating the group formation. However, when we constructed groups according to the students' educational backgrounds, experiences, and attitudes, the groups formed became heterogeneous [4]. JuHou and Hu [5] claimed that the diversity of learner characteristics plays a critical role within the process of achieving learning goals. In addition, there are several researchers investigating the relationship between learning style and student performance in the field of computer science [6]. In some literature on cooperative learning, they claimed teachers need to mix dissimilar students together to ensure true heterogeneous composition. However, in the real world, there are quite a lot of factors that can be considered to construct heterogeneous groups of students. The factors based on class objectives can include student attributes such as attitude, grade point average, current classes, and previous academic background. Therefore, the search for an optimized group composition of all students is a time-consuming task.

The paper is divided into 5 parts including this section. The remainder is organized as follows. A brief overview of heterogeneous grouping is presented in section 2. Section 3 describes our algorithm including the

method formulation by GA. Section 4 demonstrates the experimental results. The last section includes conclusions and future work.

2. Heterogeneous Grouping

There have been numerous researches done during the past hundred years on ability grouping. Therefore, heterogeneous grouping now has been an important issue especially for primary, middle and high school level [7], [8]. In the work of Bosco [7], he created groups of university students by ability, which are homogeneous grouping and heterogeneous grouping. Bosco explicitly claimed that heterogeneous groupings in universities performed better than the control self-selected sections. Some literature such as Cohen and Lotan [9] efficiently stated that most schools are prepared for purposes of reducing the heterogeneity of the students. However, the students in the same class are different. It is difficult to make them similar in the terms of experiences and educational background because the students have their own attitudes, preferences, educational background and experiences. In general classrooms, heterogeneous groups are called mixed groups that include students with a wide variety of instructional levels and with varied learning levels. For example, when organizing groups of students in a class to accomplish a common task, the teacher deliberately has low, medium, and high ability students. Currently, there is ample research on heterogeneous grouping for students. Currently, there appear to be more advantages to heterogeneous grouping in terms of academic achievement. Heltemes [10] stated that Heterogeneous ability groups benefit students by enhancing their attitudes toward each other and school work, which is supported by several literature such as the work of Ballantine and Larres [11] and Robinson [12]. Moreover, Sapon-Shevin and Duncan [13] claimed that one of the principles of cooperative learning is the principle of heterogeneous grouping of students. The key assumption is that a group works better when peers are balanced in terms of diversity, functional roles, and personality differences. Moreover, one important aspect of constructing cooperative learning groups within small groups is the maximization of the heterogeneity of the students. Therefore, students should be allocated in groups that are mixed in terms of academic abilities and aptitudes, personality, social class, religion, language proficiency, race, and sex [14]. As a consequence, heterogeneous grouping is also termed mixed ability or achievement grouping. Several advantages of heterogeneous grouping have been shown in studies, such as the fact that it improves the academic achievement of students [15].

3. A Genetic-algorithm Approach for Balancing Heterogeneous Grouping

In this section, the proposed algorithm to form heterogeneous groups of students is described.

3.1. Basic Mechanics of Genetic Algorithms

Genetic algorithms (GAs) are a very effective way of searching a solution to a complex problem. They are a kind of random heuristic search that mimics the process of natural selection; therefore they do an excellent search through a large and complex search space. In the 60's, John Holland was the first pioneer who applied genetic algorithms to artificial problems. Currently, GAs has been widely studied, experimented and applied in many fields in the real world; such as a multi-objective optimization problem [16], buyer coalition formation presented [17], and the audio watermarking process [18].

In the traditional design of GAs, a genetic representation of the problem must be defined. When we need to solve the problem with GAs, we need to encode our problem into the chromosome. This step is the beginning step that we need to do. For the first generation a set of chromosomes called "Genotype" is generated for the initial population. These populations are generated randomly. All individuals are evaluated by a fitness function. The fitness function is an objective function that is used in a process to choose which chromosomes will continue on to the next generation. Some chromosomes with a low fitness value will not be selected and die out. And, three common operators of GAs, reproduction, crossover, and mutation, perform during one generation in

order to generate new offspring. The objective of such operators is to preserve the chromosomes, or a part of them, which represent better solutions following natural selection principles [19]. As GAs are a kind of heuristic search algorithms that are shown to be structurally similar, the best individual based on the fitness value in the last generation may simply approximate the solution to the problem. The flowchart of traditional GAs is presented in Fig. 1.

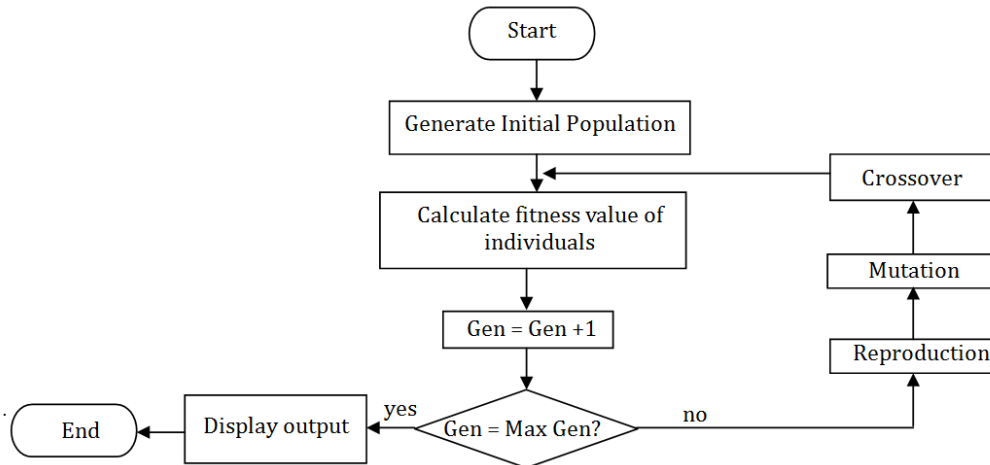


Fig. 1: GAs flowchart

3.2. The Student and Groups Vectors

Let a class contains n students, denoted by $S = \{s_1, s_2, \dots, s_{n-1}, s_n\}$. Each student has its attributes represented in a multidimensional space by a vector, in the form of a vector field $\{v_1, v_2, \dots, v_{k-1}, v_k\}$. These attributes are grade point average (GPA), prior classes, educational backgrounds, and student's personality and performance. That is, student i in k -dimensional space is represented as field $(v_1^i, v_2^i, \dots, v_k^i)$. All attributes must be converted to be numeric to help estimating the student knowledge levels. In the group formation, each student belongs only to one group. If there are n students to form q groups of m students, then $\lfloor n/m \rfloor$. That is, some groups may contain $m+1$ students. As stated earlier, we need to generate groups of students equally in the terms of educational skills. Therefore, the size of all groups must be the same if it is possible. As stated earlier, GA is a key process for our algorithm. To employ a genetic algorithm, we encode the problem as a chromosome. Since we have n students to create the group formation, the chromosome is composed of n fixed characters. Each element of the chromosome represents the group that the students belong to. If p groups are generated, an element of the chromosome can be assigned into any G_1, G_2, \dots, G_p . Therefore, a group i of m students is represented as $G_i = (s_1^i, s_2^i, \dots, s_m^i)$, where $s_m^i \in S$. For instance, if the student s_1 is assigned to the group G_1 , the first position of the chromosome is G_1 , as represented in Fig. 2. Generally, all attributes of a certain group are calculated by the average value of all students in the same group. As a result, a vector field of group G_i is presented as below:

$$G_i = \left(\frac{\sum_{j=1}^m v_1^j}{m}, \frac{\sum_{j=1}^m v_2^j}{m}, \dots, \frac{\sum_{j=1}^m v_k^j}{m} \right)$$

,where $|G_i| = m$ and $1 \leq i \leq p$.

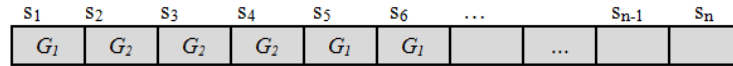


Fig. 2: The chromosome structure for n students where G_i representing the group number i .

3.3. Quality of Group Formation

In order to achieve the equity of academic attributes among heterogeneous groups, the formation of student groups may take into account more than one attributes. Therefore, a particular fitness function for evaluating all chromosomes is designed. Fundamentally, this function tells the quality of the group formation. To achieve the mechanism for equity, the fitness function is calculated by using the Euclidean Distance (ED) between generated groups. If two students named s_i and s_j belong to the same group, the distance between students is evaluated by the Euclidean distance (ED) as seen in (1). During one generation, a chromosome representing the possible group formation of heterogeneous students needs to be calculated for fitness value. For instance, the graph in Fig. 3 represents the distance of two s_i and s_j , where a two-attribute vector is applied. If the two vectors are similar then the value of ED is close to zero.

$$ED = |s_i - s_j| = \sum_{k=1}^m \sqrt{(v_i^k - v_j^k)^2} \quad (1)$$

,where v_i^k is the value of attribute k of student i .

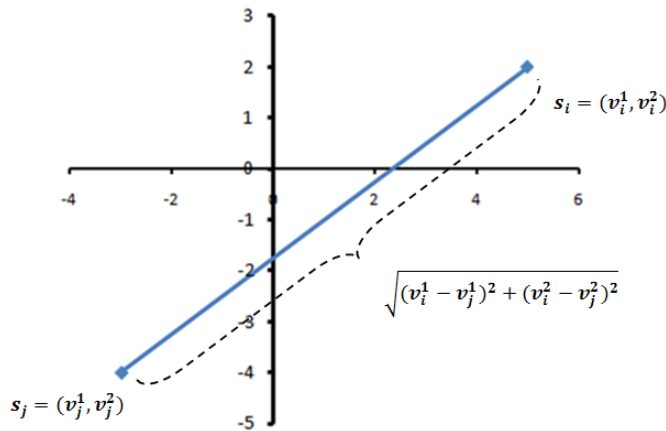


Fig. 3: The distance of students s_i and s_j , where $s_i = (v_i^1, v_i^2)$ and attribute vector of s_j is (v_j^1, v_j^2) .

Let's a t -attribute vector) of group G_i is $(V_i^1, V_i^1, \dots, V_i^t)$. Based on (1), if the total number of all groups is p , then the fitness function for our algorithm becomes as following equation.

$$f(\text{chromosome}) = \sum_{i=1}^p \sum_{j=i+1}^p \sqrt{\sum_{k=1}^r (V_i^k - V_j^k)^2} \quad (2)$$

, where p is the total number of all groups, v_g^k is the attribute k of group g and r is the total number of attributes.

The graph in Fig. 4 shows how to calculate the Euclidean distance (ED) of p groups, where a k -attribute vector is applied.

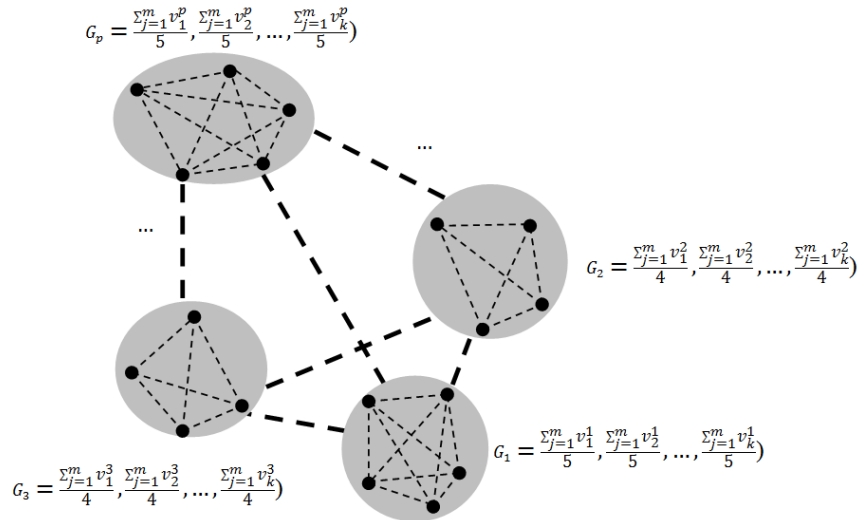


Fig. 4: The distance between groups.

3.4. Crossover Operator

For our genetic-algorithm approach, the better chromosome that yields the lowest fitness value is considered as a promising solution, and it implies that the constructed groups are more balanced with heterogeneous students in terms of educational background.

Once the genetic representation and the fitness function are defined, we then design the genetic operators to create offspring. In this paper we use single-point crossover. One point (d) is randomly selected where the value of d is between 1 and chromosome length n . Then, all characters from beginning of chromosome to the crossover point is copied from one parent, the rest is copied from the second parent. For instance, let a chromosome length = 8 and an integer position p along the chromosome is selected randomly. Two new offspring are created as presented in Fig. 5.



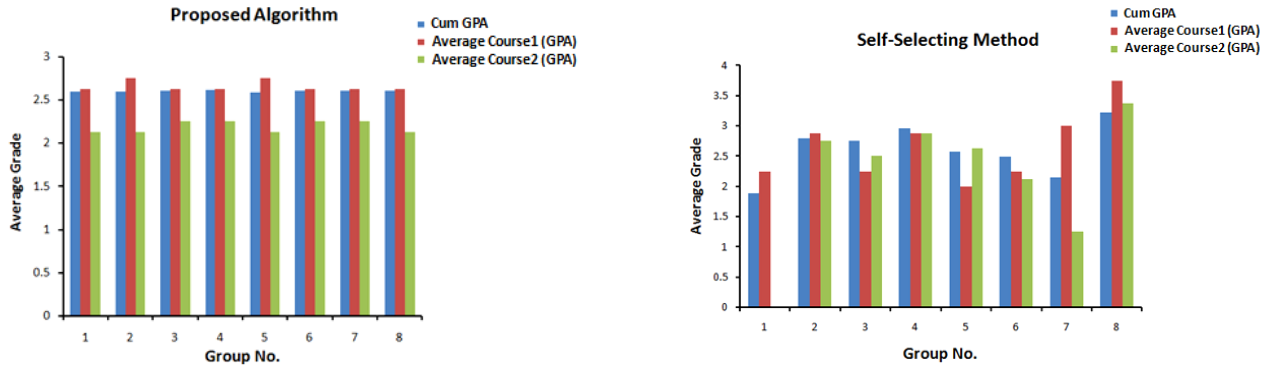
Fig. 5: Single point crossover example (chromosome length = 8).

4. Experimental Results

In this section, we present the results of a case study for generating student groups in CS350 course (Data Structures) during the 1st semester of 2015 at Bangkok University. It was performed using 32 students majoring in Computer Science. We had run the program several times to see which initial values of parameters would direct the algorithm's search to the optimized formation. Parameters for controlling our algorithm are found as illustrated in Table 1.

Table I: Initial parameters for GAGF algorithm

Constants	Detail	Value
M	Population size	500
Max_Gen	Maximum number of generations	400
p_c	Crossover probability	0.2



(a) Results of GAGF algorithm (fitness value = 2.97)

(b) Self-selecting method (fitness value = 44.5)

Fig. 6: Experimental results.

Based on the experimental results, eight groups of students are created by GAGF algorithm. The attribute values of the groups are illustrated in Fig. 6(a). The established groups are similar in all attribute values, since the algorithm try to make the equity among the groups. It indicates that our algorithm is able to distribute heterogeneous students optimally. To guarantee the performance of our algorithm we compared the results to the self-selecting method made by the students themselves. The results of the self-selecting method are presented in Fig. 6(b). As we can see, most groups made by students are greatly unbalanced. The groups are very much different in most attributes. Moreover, the fitness value is 44.5, which is higher than the fitness value of our algorithm. This means that the self-selecting method generates student groups with no direction, and the fairness among generated groups is difficult to be occurred in terms of academic abilities.

5. Conclusion and Future Work

In this paper, we present the algorithm called Genetic Algorithm for Group Formation (GAGF) to help constructing heterogeneous grouping based on GAs by concerning prior educational knowledge. The algorithm aims to achieve fairness among constructed groups and ensure that heterogeneous students are allocated optimally. The quality of the groups generated by our proposed algorithm is compared to the self-selecting method. According to our experiment, the algorithm performed better than self-selecting method. The results have shown that our algorithm achieves the mechanism for equity among constructed groups in terms of academic abilities and optimally distributes heterogeneous students within the group formation. The research findings of this work may insist teachers and researchers to form the student groups in the environment of cooperative learning. In the future work, further more experiments need to be conducted in order to determine the quality of solutions. We also plan to investigate the algorithm to support the teachers with less manual process.

6. References

- [1] Y. M. Huang, Y. W. Liao, S. H. Huang, and H. C. Chen, "Jigsaw-based Cooperative Learning Approach to Improve Learning Outcomes for Mobile Situated Learning." *Journal of Educational Technology & Society*, 17(1), pp. 128-140, 2014.
- [2] J. M. Balmaceda, S. N. Schiaffino, and J. A. D. Pace, "Using constraint satisfaction to aid group formation in CSDL." *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 17(53), pp. 35-45, 2014.

- [3] P. Paredes, A. Ortigosa, and P. Rodriguez, "A Method for Supporting Heterogeneous-Group Formation through Heuristics and Visualization." *J. UCS*, 16(19), pp. 2882-2901, 2010.
- [4] E. Martín and P. Paredes, "Using Learning Styles for Dynamic Group Formation in Adaptive Collaborative Hypermedia Systems." in *ICWE Workshops*. 2004, pp.188-198.
- [5] H. JuHou and H. Hui, "EBFO-based Personalized Learning Materials Recommendation in Cooperative Learning Group." *Journal of Convergence Information Technology*, 8(10), pp. 1057, 2013.
<http://dx.doi.org/10.4156/jcit.vol8.issue10.128>
- [6] L. Chen, and J. M. C. Lin, "Learning styles and student performance in java programming courses." in *Proc. of the 2011 international conference on frontiers in education: CS and CE, 2011*, pp. 53-58.
- [7] K. R. Bosco, "The impact of ability grouping on college student's performance in introductory geology labs at a Midwestern university," Doctoral dissertation, Kansas State University. U.S.A. 2009.
- [8] M. Bahar, "Student Attitudes Towards Change From Ability Grouping To Heterogeneous Grouping at a University Class." *Mevlana International Journal of Education (MIJE)*, 2015.
- [9] E. G. Cohen and R. A. Lotan, *Designing Groupwork: Strategies for the Heterogeneous Classroom* Third Edition. Teachers College Press, 2014.
- [10] L. Heltemes, "Social and Academic Advantages and Disadvantages of Within-class Heterogeneous and Homogeneous Ability Grouping" (2009). Mathematical and Computing Sciences Masters. Paper 93.
- [11] J. Ballantine, and P. M. Larres, "Cooperative learning: a pedagogy to improve students' generic skills?", *Education + Training*, Vol. 49 Iss: 2, pp.126 – 137, 2007.
- [12] J. P. Robinson, "Evidence of a differential effect of ability grouping on the reading achievement growth of language-minority Hispanics." *Educational Evaluation and Policy Analysis*, 30, pp. 141-180, 2008.
<http://dx.doi.org/10.3102/0162373708317742>
- [13] M. Sapon-Shevin, B. Ayres, and J. Duncan, Cooperative learning and inclusion. In J.S. Thousand, R.A. Villa, & A.I. Nevin (Eds.), *Creativity and collaborative learning: The practical guide to empowering students, teachers, and families* (2nd Ed.), pp. 209-222, 2002.
- [14] G. Jacobs, (2004). Cooperative Learning: Theory, Principles, and Techniques. Reading Matrix Inc. Available: <http://www.readingmatrix.com/conference/pp/proceedings/jacobs.pdf>
- [15] A. J. Kruse. (July 2011). The Benefits of Heterogeneous Groupings over Homogeneous Groupings. Andrew Kruse. Available: <http://andrewkruse.efoliomn.com>
- [16] O. H. Choon and S. L. Tilahun (2011). "Integration fuzzy preference in genetic algorithm to solve multiobjective optimization problems." *Far East Math. Sci*, 55, pp. 165-179.
- [17] A. Sukstrienwong, (2012). "Buyer coalition formation by hierarchical genetic algorithm." *Far East Journal of Mathematical Sciences*, 71(2).
- [18] M. Ketcham and S. Vongpradhip, (2013). Digital audio watermarking using multi-objective genetic algorithms." *Far East Journal of Mathematical Sciences*, 77(2), 275.
- [19] D. Whitley, "A genetic algorithm tutorial." *Statistics and computing*, 4(2), pp. 65-85, 1994.
<http://dx.doi.org/10.1007/BF00175354>