Group Formation Based on Heterogeneous Grouping by Genetic Algorithm:
A Case Study of Bangkok University

Asst. Prof. Anon Sukstrienwong

This research was funded by Bangkok University

2013

Resarch Title Group Formation Based on Heterogeneous Grouping by Genetic Algorithm: A Case Study of Bangkok University

Researcher Asst. Prof. Anon Sukstrienwong

Working Duration May 2012 – November 2013

Research Supporter Bangkok University

Abstract

Forming student groups has long been considered an effective approach to collaborative work in several universities. This involves balancing several students at different levels of knowledge in the appropriate groups in order to learn from others. Due to the complexity of the developing computer software projects, the cooperation of the group's members is becoming more important, as it helps reduce an individual's workload. Thus, an assigned software project can be delivered successfully on time. In terms of the fairness of student's educational skills among formed groups, however, forming groups of students are becoming difficult when the number of students is great and the heterogeneity of students is more complex.

Since few studies have yet to consider the formation of groups for computer science students in software development projects based on a heterogeneous grouping, in this paper, we present an approach called Student Formation by Heterogeneous Grouping (SFHG), to generate student group formation based on a genetic algorithm (GA). The main aim of the approach is to form student groups based on the theory of heterogeneous grouping for the students majored in Information Technology, using Bangkok University as a case study. Various factors, such as previous grades and expertise are considered in group formation in order to mix and balance the students with different levels of programming skills equally among generated groups. Moreover, the algorithm aims to achieve fairness among the groups and compares the result to a random self-selecting method made by the students. The case study was performed with a group of 70 students to demonstrate the scalability and the ability of the approach. The paper also presents a web-based system for student group formation to facilitate teachers in gathering student's information and generating group formation in a class. Based on our case study and resulting experiment, the results illustrate that the performance of SFHG is efficient at distributing and mixing heterogeneous students in the established groups.

ชื่องานวิจัย การรวมกลุ่มนักศึกษาบนกลุ่มที่แตกต่างกันด้วยแบบคำนวณพันธุกรรม: กรณีศึกษามหาวิทยาลัยกรุงเทพ
ผู้วิจัย ผู้ช่วยศาสตราจารย์อานนท์ สุขเสถียรวงศ์
ระยะเวลาดำเนินงาน พฤษภาคม 2555- พฤศจิกายน 2556
ทุนอุดหนุนการวิจัย มหาวิทยาลัยกรุงเทพ

บทคัดย่อ

การจัดกลุ่มนักศึกษาได้รับการยอมรับว่าเป็นวิธีการที่มีประสิทธิภาพสำหรับการทำงานร่วมกันในหลายมหาวิทยาลัย ซึ่งการจัดกุล่มนักศึกษาจะเกี่ยวข้องกับการปรับสมดุลของนักศึกษาที่มีพื้นความรู้แตกต่างกันให้อยู่ในกลุ่มที่เหมาะสมในอันที่จะก่อให้เกิดการเรียนรู้แก่สมาชิกในกลุ่ม เนื่องจากความซับซ้อนของการพัฒนาคอมพิวเตอร์ซอฟแวร์ ความร่วมมือของสมาชิกในกลุ่มที่จะทวีความสำคัญมากขึ้น เพราะเป็นตัวช่วยลดภาระงานของแต่ละบุคคล จึงทำให้ซอฟต์แวร์ที่ได้รับมอบหมายสามารถพัฒนาให้สำเร็จรุล่วงได้ตรงตามกำหนด ในแง่ของความเสมอภาคและเป็นธรรมในการกระจายนักศึกษาที่มีระดับพื้นฐานความรู้แตกต่างกันในกลุ่มต่างๆ การจัดกลุ่มนักศึกษาจะกลายเป็นเรื่องยากและความซับซ้อนมากเมื่อนักศึกษามีขนาดใหญ่และความแตกต่างของความแตกต่างของนักศึกษาที่มีความซับซ้อนมากขึ้น

งานวิจัยฉบับนี้เสนอวิธีการจัดกลุ่มนักศึกษา ซึ่งเรียกว่า การรวมกลุ่มนักศึกษาบนกลุ่มที่แตกต่างกัน (Student Formation by Heterogeneous Grouping: SFHG) โดยอาศัยแบบคำนวณพันธุกรรม (Genetic Algorithm: GA) ทั้งนี้ จุดประสงค์หลักเพื่อทดสอบวิธีการรวมกลุ่มนี้กับกลุ่มนักเรียนคณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยกรุงเทพ โดยใช้เกรดการเรียนที่ผ่านมา และความเชี่ยวชาญเป็นปัจจัยประกอบในการพิจารณาจัดกลุ่มการของนักศึกษา เพื่อที่จะกระจายนักศึกษาที่มีระดับความรู้ด้านทักษะการเขียนโปรแกรมที่แตกต่างกันไปยังกลุ่มต่างๆ ที่สร้างขึ้นได้อย่างเท่าเทียม นอกจากนี้แบบคำนวณยังมีวัตถุประสงค์เพื่อให้มีความเท่าเทียมและเป็นธรรมระหว่างกลุ่มต่างๆที่ถูกสร้างขึ้น และนำไปเปรียบเทียบผลกับวิธีการสร้างกลุ่มด้วยตนเองของนักศึกษา กรณีศึกษาในงานวิจัยนี้ได้ดำเนินการกับกลุ่มนักศึกษาจำนวน 70 คน เพื่อแสดงให้เห็นถึงความยืดหยุ่นและความสามารถของวิธีการที่นำเสนอนี้ ในงานวิจัยฉบับนี้ยังได้นำเสนอโปรแกรม

ประยุกต์บนเว็บ ซึ่งถูกพัฒนาเพื่ออำนวยความสะดวกให้แก่อาจารย์ในการรับข้อมูลและสร้างกลุ่มของนักศึกษาในชั้นเรียน จากผลของกรณีศึกษา และการทดลอง ผลลัพธ์ที่แสดงให้เห็นว่า SFHG มีประสิทธิภาพในการจัดกลุ่มและการผสมผสานนักศึกษาที่มีความแตกต่างกันในกลุ่มที่สร้างขึ้นได้

TABLE OF CONTENTS

List of Tables

Page

List of Figures

Page

Chapter 1

Introduction

1.1 Introduction

The cooperation of a group's members in developing a software project is becoming more important, as it helps reduce an individual's workload and speeds up required tasks. Therefore, the assigned project can be completed and delivered successfully on time. Moreover, by working in the group it is possible to enhance a student's ability to manage and solve projects efficiently. In the current status, the problem of how to gain individual student achievement in groups is an important topic (Ting-Yi Chang, 2009). Therefore, various techniques of cooperative learning have been proposed by researchers (Chiu, 2008) (Pedro, Alvaro, & Pilar, 2010) (Wang, Sunny, & Chuen-Tsai, 2003).

Group formation plays a critical role in terms of enhancing the success of the learning process. This involves structuring groups in order to help students learn together. One principle of cooperative learning is heterogeneous grouping. Pedro et al. (Pedro, Alvaro, & Pilar, 2010) presented that heterogeneous grouping is necessary in order to ensure equal opportunities for all students to complete assignements. Also, Duan and Harley (Duan & Harley, 2011) studied heterogeneous grouping based on specific characteristics. They suggested that heterogeneous grouping is able to promote positive interdependence, better group performance, and effective interaction. However, few studies of learning styles consider the formation of groups for the Science and Technology students working on computer assignements and software development projects at a university (Gilbert & Swanier, 2008).

More over, due to the increasing complexity in computer software projects, forming groups is becoming more important for software development projects, to ensure that software products can be delivered successfully on time. The group project is comprised of several students working on a large project over several days for the length of the entire semester. In general, the success of student groups in developing a software project depends on various factors such as the personalities, expertise, performances, and the level of collaboration of the people involved in the group (Bekele, 2005). It has been observed by some researchers that heterogeneous groups are better in a broader range of tasks (Martin & Paredes, 2004).

In forming student groups, some instructors focus on student opinions by asking them who they would like to work with. Nevertheless, student choices may be the best way to form groups if they have already known each other for a period of time. The groups formed in this way often tend to have the same gender, personality, and educational skill. Due to this result, some teachers might set some kind of inquiring questions related to some sorts of rules of association for students to answer, so that we can obtain personalities and some other information directly from students. Moreover, teachers can get some of the student's academic records such as the average grade (GPA), and the register courses to help measure students when forming groups. In this paper, we propose an approach to form groups based on heterogeneous grouping in order to create a fairer and more equitable range in the group formation.

Theoretically, a heterogeneous group works with the assumption that the generated group can work better when the members are balanced in terms of diversity, based on functional roles, and personality differences. Moreover, it is stated by Wang et al. (Wang, Sunny, & Chuen-Tsai, 2003) that heterogeneous groups should be comprised of students whose prior knowledge should be unequal. Hence, an individual member accomplishes when the whole group succeeds. Everyone must work within the group in order to complete tasks efficiently. However, group formation becomes more complex to achieve when dealing with a large number of students (Harrison, Griffin, & Broughton, 2009). This will be achieved using a well-known genetic algorithm, in order to maximize the students' programming skills within the formed groups. Therefore, the main objective of this research is to develop and implement a system for 'Student Formation by Heterogeneous Grouping' (SFHG), tested on a desired group of students.

In general classes, most activities of cooperative learning are carried out by a teacher. The teacher randomly assigns the students to a certain group and asks the students some questions associated with class objectives. The teacher can get information directly from the students such as a student's academic records, for example the grade point average (GPA) and the registered courses, to help measure the students when forming the groups. To ensure fairness and equity among the groups, the constructed groups might be similar in as many attributes as possible, while all opportunities of learning for all students are concerned. Therefore, seeking the best group composition of the students may be an exhaustive search since this is time-consuming.

There are six parts to this paper including this introduction. The remainder of this paper is organized as follows. Section 2 gives a formal definition of the problem. The genetic algorithm which is a tool for forming the heterogeneous groups is described in Section 3. Section 4 shows how we can apply it in a real-world scenario. An experiment was performed on a case study of classes of CS250 and IT350 in 2011 to demonstrate the ability and scalability of the proposed approach. The experimental results and discussion are reported in Section 5. Finally, the conclusion and suggestions for further work are presented in Section 6.

1.2 Objectives of the Research

In this paper, we propose an approach to form groups based on heterogeneous grouping in order to achieve a fairer distribution of all attributes within the group formation. Therefore, the main objective of this research is to develop and implement a system for "Student Formation by Heterogeneous Grouping" (SFHG), tested on a desired group of students.

1.3 Scope of the Research

We will develop an approach for student group formation by using genetic algorithms. Euclidean distance[1] is employed in order to analyse the quality of the group formation. An empirical example of Science and Technology students at Bangkok University is selected as a case study. In addition, we will also develop a web-based system as a tool for teachers to collect the student's information and manage the coalition process.

1.4 Expected Benefits

The designed method developed in this proposal provides:

1) A paper that aims to study the group formation with the heterogeneous grouping methodology for Science and Technology students at a university, in order to achieve fairness and equity in group formation.
2) A guideline and clear understanding when forming student groups with the heterogeneous grouping method, by using a genetic algorithm.
3) Develop the system tool for instructors to form student groups, based on the system of heterogeneous grouping.

1.5 Research Methodology

The details of methodologies used in this proposal are listed below.

1) Review literatures concerning cooperative learning techniques and group formation based on the heterogeneous grouping.
2) Study the principal of genetic algorithms.
3) Design the student's vector for the problem and construct a genetic chromosome for the problem.

---

[1]Euclidean distance is also named Euclidean metric which is the "ordinary" distance between two points.

4) Develop a web-base application for the student formation by Java, to test our proposed method.
5) Guarantee the performance of the algorithm by comparing the results of this scheme's simulation with a self-selecting method.
6) Analyze and discuss the results.
7) Make conclusions and suggest future research.

Chapter 2

Literature Review

In order to understand the issues involved with forming groups of learners based on the heterogeneous grouping by using a genetic algorithm (GA), we need to understand the cooperative learning technique which relies upon students in the group, and the fundamentals of genetic algorithms.

2.1 A Brief Concept of Cooperative Learning

Cooperative learning is one of the most effective approaches of theory in academic and social learning experiences, in which students work together to accomplish the own and each other's learning (Johnson & Johnson, Making cooperative learning work, 1999); (Johnson, Johnson, & Stanne, 2000); (Johnson, Johnson, & Smith, The state of cooperative learning in postsecondary and professional settings, 2007); (Phuong-Mai, Terlouw, & Pilot, 2005); (Felder & Brent, 2007); (Gholami, 2011). It has been emphasized by several researchers who are concerned primarily with promoting students' academic achievement and social skills development (Johnson & Johnson, Learning together and alone, 1987). It has been observed by some researchers that heterogeneous groups are better in a broader range of tasks (Martin & Paredes, 2004). Thus, it should come as no surprise that various techniques of cooperative learning have been proposed by researchers (Ting-Yi Chang, 2009) (Pedro, Alvaro, & Pilar, 2010).

Therefore, there exists several studies, published worldwide in many disciplines, which demonstrate that cooperative learning within groups has a number of positive benefits for students, such as:

- Academic achievement, in which students taught in this manner will achieve higher grades and learn with a greater comprehension of the content;
- Social skills development, among learners because everyone in a cooperative learning group must be an active participant in the learning process. Students within the group communicate with others to help one another in sharing ideas. Students learn to take the benefits for their contribution;
- Learning different ways of thinking rather than through teacher-led discussion;
- Clarifying other students' misconceptions in which a student is able to see points of view other than their own.

2.1.1 What is Cooperative Learning?

The conceptual approach to cooperative learning for students in higher education has been broadly defined since the 1970s. Therefore, several definitions of cooperative learning have been extensively formulated. The term of cooperative learning refers to the teaching strategy formulating small groups to work on an assigned project, in which groups achieve an outcome at the end of cooperative learning. Each member of a team works together and is responsible, not only for learning what is taught, but also for helping teammates learn (Gholami, 2011). One of the most famous definitions of cooperative learning at the university is defined by Johnson *et al.* (Johnson, Johnson, & Smith, The state of cooperative learning in postsecondary and professional settings, 2007). Students in class involved in cooperative learning work together in small groups towards a shared or common goal, which they could not achieve individually. They are likely to benefit from cooperative learning by encouraging each other's efforts and assisting each other in completing the assigned task (Ounnas, Davis, & Millard, 2009). According to Johnson *et al*, cooperative learning can be used for a variety of assignments that can be given to students in lectures. There are five essential elements necessary for cooperative learning groups to be effective (Johnson, Johnson, & Smith, The state of cooperative learning in postsecondary and professional settings, 2007); (Felder & Brent, 2007).

1. Positive interdependence
2. Face-to-face primitive interaction
3. Individual accountability
4. Interpersonal and small group social skills
5. Group processing

2.1.1.1 Positive interdependence

Positive interdependence is considered to be the heart of cooperative activities in structuring cooperative learning. When working with a common idea, each individual student must believe that it is good to learn with other students, and that both individual learning and the task outcomes will be better as a result of collaboration. Students are willing to work together to get their assigned job done. In addition, all students can achieve their learning goals if, and only if, all members of their group also achieve their goals. In other words, students must perceive that they "sink or swim together" (Johnson, Johnson, & Holubec, 1998). In structuring positive interdependence, it is important to include the provision of (a) common rewards such as a shared grade because this is the most common reward given in a classroom, (b) shared resources; as each member has different expertise, and (c) specific roles assigned to team members that demonstrate the responsibilities required by the group in order to complete a shared task.

2.1.1.2 Face-to-face primitive interaction

Face-to-face primitive interaction is the second basic element of cooperative learning. It is also called "promotive interaction" because it occurs when group members promote each other's success by sharing resources and by helping, supporting, encouraging, and applauding each other's efforts to achieve. Once the positive interdependence is established, teachers must ensure that students interact to help each other to accomplish the task, and promote each other's success (Johnson & Johnson, Learning together and alone, 1987). Students are expected to share their conceptual knowledge and teach what they know and how to solve problems to other members. This includes discussing with each other concepts and connecting present with past learning.

2.1.1.3 Individual accountability (Personal responsibility)

Individual accountability exists when the performance of each individual student is assessed and the results are given back to the group and the individual (Johnson & Johnson, Making cooperative learning work, 1999). The essence of individual accountability in cooperative learning is to make the students learn together. To ensure that each student is strengthened, each individual student must be responsible for their own contribution to the group. When the students learn together, they can subsequently perform higher as individuals (Johnson, Johnson, & Smith, Cooperative learning returns to college: What evidence is there that it works? Change, 1998).

2.1.1.4 Interpersonal and small social skills

In cooperative learning groups, students learn academic subject matter. At the same time, they also gain knowledge of interpersonal and small group skills. Discussion may be helpful to explain to the students why they are required to work together and how the group can enhance their learning. The students must have and use the necessary leadership, decision-making, trust-building, communication, and conflict-management skills. These skills have to be taught just as purposefully and precisely as academic skills.

2.1.1.5 Group processing

Group processing occurs when members discuss how well they are achieving their goals and maintaining effective working relationships among the group, describe what actions are helpful and what are not, and make decisions about what behaviors to continue or change (Johnson & Johnson, Cooperation and the Use of Technology). Therefore, after completing their task, the students must be given time and procedures for analyzing how well their learning groups are functioning and how well their social skills are being employed. Such processing enables learning groups to focus on group activities to ensure that each member receives feedback on their participation and to maintain cooperative learnings consistently.

2.1.2 Cooperative learning techniques

Cooperative learning is actually a term that refers to various methods for grouping students, and it can be used for any type of assignment that can be given to the students in classes, laboratories, or project-based courses. Nevertheless, it is not a simple way of allowing the students to work in groups. Presently, with some recommendations for how the students may be effectively taught, a variety of cooperative learning techniques for students have been developed (Felder & Brent, 2007) such as:

1. Jigsaw
2. Peer-led team learning
3. Learning together

2.1.2.1 Jigsaw

Jigsaw is a cooperative learning technique in which students work in small teams. It was originally developed by an American psychologist named Elliot Aronson and his fellows (Elliot & Patnoe, 1997). After the invention of this technique in 1997, it was effectively considered to increase positive educational outcomes. Several modified techniques have been proposed to account for the concerns of both teachers and students. Just as in a jigsaw puzzle, each piece is important to complete the whole picture. Therefore, in the jigsaw classroom approach, each student is required for the completion and full understanding of the assigned task. The process begins with the teacher explaining to the students that they will be working in different cooperative groups to learn content. Then, the students are divided into small groups, mixed by race and by ability, to work co-operatively on an assigned content (Levinson, Cookson, & Sadovnik, 2002). As it is described in Levinson *et al.*, the classroom material, such as a biography of a historical figure and content, is carefully divided into interdependent sections, and one member of each group is responsible for reading each section. Members with the same role from each group gather to discuss and then return to their own groups to present what they have learned. Therefore, the students are motivated to listen to each other and improve intergroup relations.

2.1.2.2 Peer-led team learning

Since peer-lead team learning was developed by a chemistry educator in the 1990s, it is now becoming one of the most prominent techniques for teaching undergraduate science, math, and engineering courses. Peer-led team learning involves students working cooperatively in small groups. Student-leaders, who are considered as peers, work in a supplemental instructional environment as they work collaboratively to solve the assigned problem. It is claimed that Peer-led Team learning is an effective technique for improving retention and student performance in several courses (Horwitz & Rodger, 2009). The group meetings in peer-led team learning are prepared by faculty staff and/or peer leaders. The goal is to stimulate group members to learn together to solve the problem. Students who have done well in a lecture are recruited to become peer-leaders. Peer leaders act as facilitators rather than experts by doing their best to ensure that all participating students understand the subject material.

2.1.2.3 Learning together

Learning together is a cooperative learning technique that has been researched by David and Roger Johnson (Johnson & Johnson, Learning together and alone, 1987). The process involves students working in small groups of five-or-six members to complete a single worksheet, for which a group receives praise and recognition. It emphasizes (1) training students to be a good group member and (2) continuous evaluation of group functioning of the group members (Slavin, Sharan, Kagan, Lazarowitz, Webb, & Schmuck, 1985). Many researchers choose to use this technique because it helps students understand the content and get more information on the assigned task.

2.2 Basic Definition of Heterogeneous Groupings

Nowadays, several principles have been proposed for cooperative learning. One of the principles of cooperative learning is the principle of heterogeneous grouping (Sapon-Shevin, Ayres, &

Duncan, 2002). This is because one important aspect of creating cooperative learning groups is maximizing the heterogeneity of the students within the small groups. Therefore, students should be placed in groups that are mixed by academic abilities and aptitudes, personality, social class, religion, language proficiency, race, and sex (Jacobs, 2004).

Heterogeneous grouping works with the assumption that groups work better when the members are balanced in terms of diversity, based on functional roles or personality differences (Bekele, 2005). As a consequence, heterogeneous grouping is also termed mixed ability, collaborative grouping, or achievement grouping.

Several advantages of heterogeneous grouping have been proven, such as that it improves student's academic achievement (Kruse, 2011). It is argued by Jacobs (Jacobs, 2004) that heterogeneous grouping at the elementary grade levels has been found to produce academic and social benefits for high-ability, average-ability, and low-ability learners. In the research bulletin of North Carolina Middle School, it shows that heterogeneous classrooms maximize greater learning opportunities for low-ability students, without being detrimental to high-ability students (Strahan, Hartman, & Sikes).

The advantages of heterogeneous grouping are briefly detailed as follows.

1. It provides students with different personality attributes and performance levels, such as gender, ethnicities, culture, academic self-discipline, and academic self-confidence, with the opportunity to learn together.
2. It allows students to learn from each other's strengths and weaknesses.
3. It allows students to share their different experience.

2.2.1 Factors for Heterogeneous Grouping

In order to establish a mechanism for heterogeneous grouping, we need to understand the major attributes or factors related to students that can be considered in group composition and which have a significant effect on the students' academic performance.

2.2.1.1 Academic factors

Academic factors of academic success have been considered by several psychologists and educators, as cooperative learning has been implemented and experienced widespread use in schools and colleges. The following are some basic academic factors frequently used in literatures to explain the performance level of students.

1. Grade point average (GPA): This is a cumulative grade point average which a student has earned from previous classes, which has been found to be a significant indicator of student performance in all major academic fields. Ismail and Othman (Ismail & Othman, 2006) used the cumulative grade point average (CGPA) as one of the main academic factors to measure early student education and performance. However, Ismail and Othman claimed that this is not a good way to compare two students who come from different previous schools because they have experienced different curricula, different approaches of teaching, and grading criteria. In fact, it only tells whether the students from one faculty are 'better' or 'worse' than the students from the other (Ismail & Othman, 2006).
2. Admission points: The number of candidates seeking admission to university has been marked in recent years by rapid growth. Basically, most universities use an admission test as a tool in selecting academically able students. Several researchers have stated that admission testing tends to have a better impact on highschool grade point average (HSGPA). This is because admission points reflect on the previous academic performance of students. Therefore, admission testing for college admissions has also shown extraordinary growth, as it is a significant predictor of achievement.
3. School material and facilities: The literatures have stated that student achievement and learning performance are influenced by school material and the facilities provided (McGowen, 2007). Facilities provided in schools to support learning activities towards achieving are academic spaces, support facilities, laboratories, and option spaces (Nurul Syakimaj, Sapri, & Mohd Shahril, 2011).

2.2.1.2 Non-academic factors

Non-academic factors play an important role as well as academic factors, as both factors can influence academic performance. Several researchers and educators have attempted to determine which non-academic variables might help explain academic performance in school. According to the report of the American College Testing (ACT), a non-profit organization specializing in education and workforce development improvement (Lotkowski, Robbins, & Noeth, 2004), certain non-academic factors mainly enhance the student success as well. Examples of non-academic factors are student confidence in the classroom, time management skills, test-taking strategies, etc. Additionally, a further report of the ACT (ACT, 2007) demonstrated that relevant non-academic factors can be classified into three groups:

1. Psychosocial factors, such as motivation (e.g., academic self-discipline, commitment to school) and self-regulation (e.g., emotional control, academic self-confidence)
2. Family factors, such as attitude toward education involved in students' school activities, and geographic stability
3. Career planning which identifies a good fit between students' interests and their post-secondary work.

2.3 Genetic Algorithm

A genetic algorithm (GA) is a heuristic search that transforms a set of objects into a generation of the population based on the principles of biological evolution and natural selection. Therefore, it does not guarantee that it will find the best solution. GA was formally introduced in the United States in the 1970s by John Holland at the University of Michigan (Holland, 1992). In a general concept, a genetic algorithm transposes the notions of natural evolution to the world of computers, and imitates natural evolution. Population of genetic algorithms begins with a population of chromosomes which are randomly bred. And, populations of the current generation are the most promising solutions for survival of the previous generation, while populations of the solution are maintaining solution diversity. This prevents the algorithm being trapped in local optima. GA is suited for solving complex optimization problems (Renner & Ekárt, 2003). Therefore, it has become a popular methodology for a variety of complex problems (Brown & Sumichrast, 2001), such as fractal image compression (Xing-yuan, Fan-ping, & Shu-guo, 2009), telecommunications and networking (Cortés, Larrañeta, Onieva, García, & Caraballo, 2001) , nurse rostering problems (Aickelin & Dowsland, 2000), logistics network design (Gen, Altiparmak, & Lin, 2006), and coalition formation (Boongasame & Sukstrienwong, 2009).

Major processes in preparing GAs are presented as follows.

1. **Encoding the problem and designing a fitness function:**

   The basic principle of GAs is that the potential solution of any problem must be represented or encoded as a genome (or chromosome) by a set of parameters. Generally, the given problem can demonstrate a number of possible representations, but the proper design of a genome can successfully lead to better solutions, while others take much computational time and fail to converge (Renner & Ekárt, 2003). In fact, the problem of various objects can be encoded in various ways with the same objective of solving the problem. Traditionally, the problem is often encoded into a fixed-length bit string. However, the problem can be transformed into a fixed-length bit string of zeros and ones, or fixed-length character strings (Duan & Harley, 2011), (Mojdehi & Barati, 2009). If the problem is represented as an n-bit string, the size of the search space is 2^n. Then, each individual chromosome in the population represents a point in the search space. When a suitable representation for the given problem is designed by means of the genetic algorithm, the fitness measure is required for each possible genome of the population. A suitable fitness function is designed to help evaluate genomes to search for the optimum solution associated with the chromosome structure. Some chromosome's structures can productively yield good solutions, while others fail and take too much time to complete. The fitness function also represents how well the solution is decoded from the chromosome to solve the addressed problem (Hwang, Yin, Hwang, & Tsai, 2008). Therefore, the fitness value of each individual must be calculated in an exact way which is related to the chromosome structure.

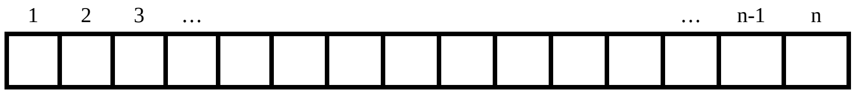


Figure 2.1 An example of a chromosome representation of the given problem

2. **Determining the parameters for controlling GAs:**

   When applying GAs to any problems, the primary parameters for controlling the genetic algorithm are compulsory. These parameters are the population size, the maximum number of generations to be run, the reproduction probability, the crossover probability, and the mutation probability, which are described in detail in the following section. Moreover, there are some other quantitative control parameters and qualitative control variables needed to be defined in order to complete the genetic algorithm.

3. **Designing the steps of GAs:**

   While the GA is running, a population of the current generation is the most promising solution for the previous generation. Following this, the genetic algorithm creates a population of solutions and applies three genetic operators; reproduction, mutation, and crossover operator. These are basic operators to search for the best solution(s). The major steps of GAs are demonstrated in Figure 2.2. In simple GAs, all computer programs start with an initial population (generation 0) by a completely random population. The GA then searches the space in an attempt to find the best solutions by applying three genetic operators which are reproduction, mutation, and crossover operators (Aickelin & Dowsland, 2000), (Sivanandam & Deepa, 2007). The purpose of these operators is to provide various new solution vectors which tend to be good. The new population is further evaluated and tested until termination criteria are found. If the termination criteria are not satisfied, the three operations keep repeatedly operating. One cycle of these operations and the following fitness evaluation of the offspring is known as a generation in GA terminology. Once the best genome based on the fitness value in any generation is selected as the result of the GA, the result may represent a fairly accurate solution to the problem.

```
Simple Genetic Algorithm
{
    initialize population;
    evaluate population;
    while TerminationCriteriaNotSatisfied
    {
        select parents for reproduction;
        perform recombination and mutation;
        evaluate population;
    }
}
```

Figure 2.2 Major steps of GAs

The operators of GAs are described in the following steps.

- **Reproduction operation** (selection operator): This is the first operator which necessarily applies to the population to maintain the generation of a new population. The operator starts by selecting genome parents of the current population based on the fitness value. The better genomes are chosen to form an intermediate population called a mating pool. Additionally, the operator makes more copies of better genomes into the mating pool. This is a way to eliminate bad parents with the lower fitness value and it is possible to generate feasible children in the next generation. As a result, the selection operation is also called a reproduction operator (Gandhi, Deeba, & Solank, 2012).
- **Crossover operation**: The crossover operator is primarily responsible for the search for the new offspring. There are a lot of approaches in implementing the crossover operations, but most variations of crossover operations create two new offspring from two existing members in the mating pool by recombining randomly chosen parts from selected parents. Three steps of the crossover operation are shown below:

1. Randomly select two existing parents from mating pool.
2. Randomly select portions of the two selected parents as crossover points.
3. Swap two portions of each parent to generate two new offspring.

A traditional crossover operator called a single-point crossover is demonstrated in Figure 2.3. It is called the single-point crossover since only one point is randomly selected at the same place on both parents. Then, two portions beyond the crossover point of each parent are interchanged to create new offspring represented as new points in the search space.

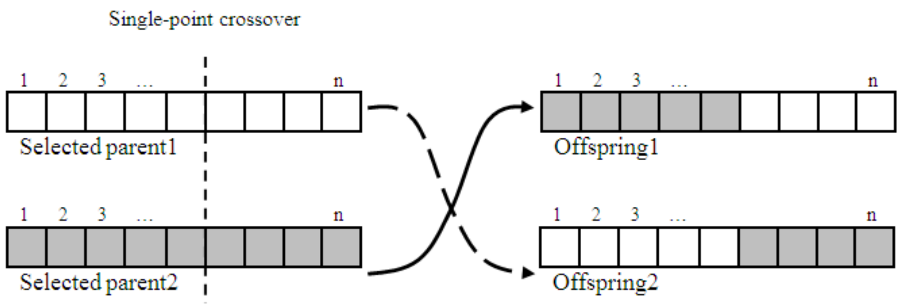


Figure 2.3 Single-point crossover

Another of the traditional crossover operations is a two-point crossover. This process randomly selects two points within a chromosome on the parent. A portion between the two points is interchanged between the parents, generating two new offspring. The two-point crossover is depicted in Figure 2.4. This crossover process is likely to generate a better generation because each new offspring receives a part of the parent's chromosome. It may be possible to generate an offspring with a low fitness value. This offspring will not be selected to be in the next generation. Therefore, the fitness measurement of each individual is essential. In the crossover process, there is a ratio of how many parents will be selected for mating, called the crossover probability (p_c). Typically, the crossover probability is usually very high (e.g., $P_c = 0.9$). This helps determine the level of accuracy and the convergence speed of genetic algorithms.

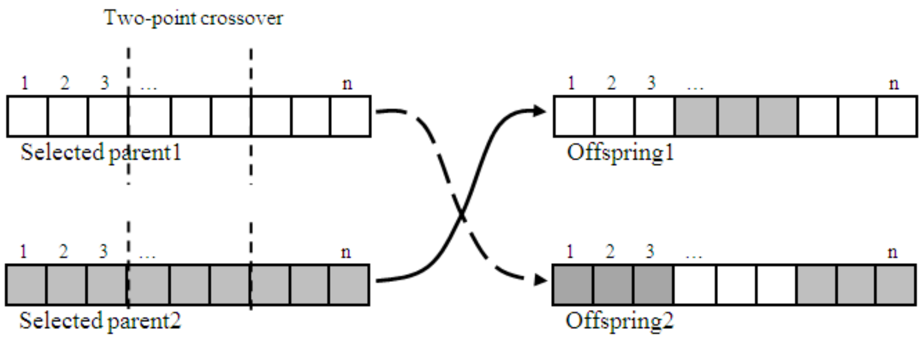


Figure 2.4 Two-point crossover

Beyond the traditional crossover, there exists another way of crossover called self-crossover. This randomly selects only one potential parent from the mating pool to produce an offspring (Kundu & Pal., 1999). Two portions within the selected parent are randomly chosen. Then, the selected portions are interchanged with each other creating a new offspring. An example of a self-crossover operator is demonstrated in Figure 2.5.

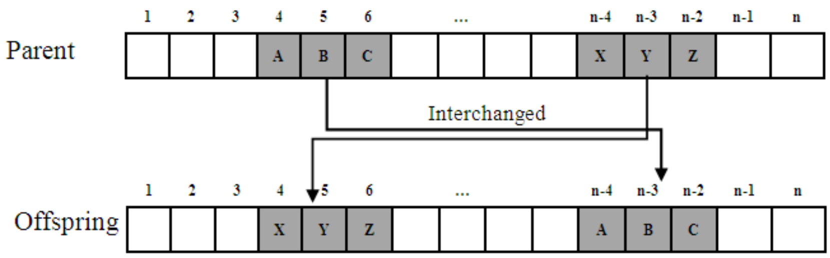


Figure 2.5 Self-crossover operator

- **Mutation operation:** This operation is frequently employed after the crossover operation. It allows a new offspring to be created as well. There are several approaches of mutation, but for a typical mutation operation, an existing parent in the mating pool is selected completely at random, and characters at one portion along the fixed-length string are randomly altered. The mutation operator may cause the chromosomes of individuals to be different from their parents. Thus, it helps to maintain diversity in the population, since the repeated use of both reproduction and crossover operators tend to reproduce a homogeneous genome population. Hopefully, this diversity successfully creates better offspring and leads to better solutions because the mutation operator may possibly produce some offspring with a low fitness value which will never be selected for the next generation. In the mutation process, a small ratio called mutation probability (p_m) is set to determine each character of the fixed-length string which will be mutated. Normally, the mutation probability is 0 - 1%.

  An example of a typical mutation operator is shown in Figure 2.6. The selected parent is a non-binary chromosome with the length of n. In this examplethe random mutation point is 3. A character at the mutation point is mutated to a new value making a new born offspring which is different from its parent.

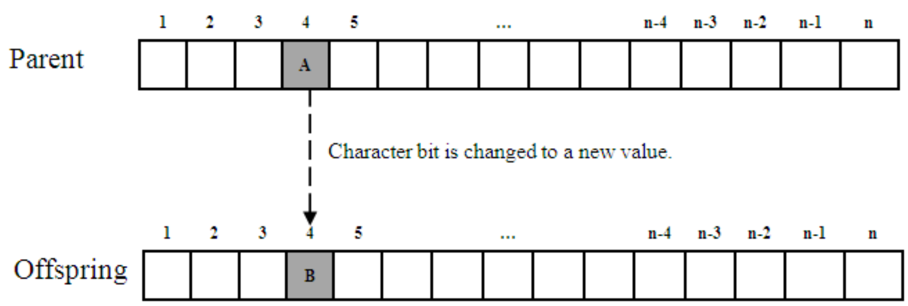


Figure 2.6 Typical mutation operator

2.4 Vectors and Euclidean distance

In mathematics and science, a vector has an exact magnitude and direction. It is simply represented by lists with numbers. Most vectors are usually denoted in lower case boldface, as $\vec{\boldsymbol{a}}$ or $\underline{\boldsymbol{a}}$. Other conventions use a tilde (~) as $\underset{\sim}{a}$. If the vector represents a directed distance between two points denoted A and B along a straight line, the vector can be noted as $\overrightarrow{\boldsymbol{AB}}$ or $\underline{\boldsymbol{AB}}$. Figure 2.5 shows two coordinates A and B. Therefore, this is a Cartesian coordinate system in two-dimensions. The Cartesian coordinates of a point are usually written in parentheses and separated by commas, e.g. (x_1, y_1) and (x_2, y_2). The notation representing the vector of coordinates A and B is $\overrightarrow{\boldsymbol{AB}}$. Hence, the distance of $\overrightarrow{\boldsymbol{AB}}$, where A is (x_1, y_1) and B is (x_2, y_2), is given by:

$$distance = |A - B| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

In mathematics, the distance formula of two points presented above is called the Euclidean distance (*ED*).

From the Figure 2.7, the given points are (10,5) and (40,25), so the distance of $\overrightarrow{\boldsymbol{AB}}$ is

$$\begin{aligned} distance &= \sqrt{(10 - 40)^2 + (5 - 25)^2} \\ &= \sqrt{(-30)^2 + (-20)^2} \\ &= \sqrt{1300} \\ &= 36.5551. \end{aligned}$$

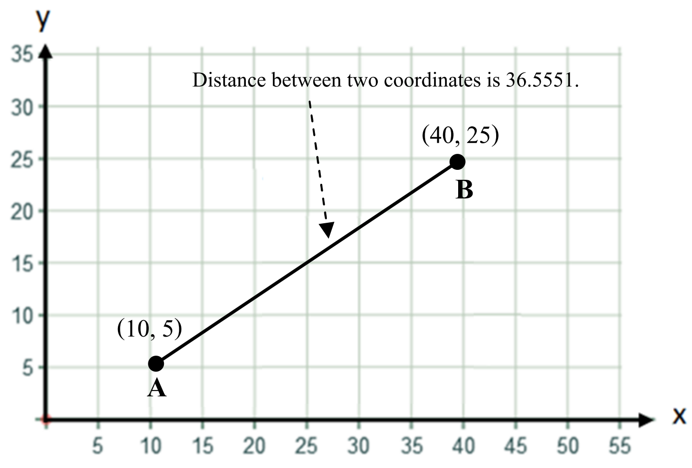


Figure 2.7 Cartesian coordinate system in two-dimensions

For a three-dimensional space such as that demonstrated in Figure 2.8, vectors are written in three numbers separated by commas which are usually written in parentheses as (x, y, z). In addition, the notation of the vector from the point A to B is simply written as $\overrightarrow{\boldsymbol{AB}}$, and the Euclidean distance between two coordinates A and B, where A is (x_1, y_1, z_1) and B is (x_2, y_2, z_2), is given by:

$$\text{Euclidean distance } = |A - B| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}.$$

From the Figure 2.6, the given points are (10, 15, 20) and (15, 35, 25), so the distance of $\overrightarrow{\boldsymbol{AB}}$ is found to be 32.4037 as presented below:

$$\text{Euclidean distance } = \sqrt{(15 - 35)^2 + (20 - 25)^2 + (10 - 15)^2}$$

$$= \sqrt{(-20)^2 + (-25)^2 + (-5)^2}$$

$$= \sqrt{400 + 625 + 25}$$

$$= \sqrt{1050}$$

$$= 32.4037.$$

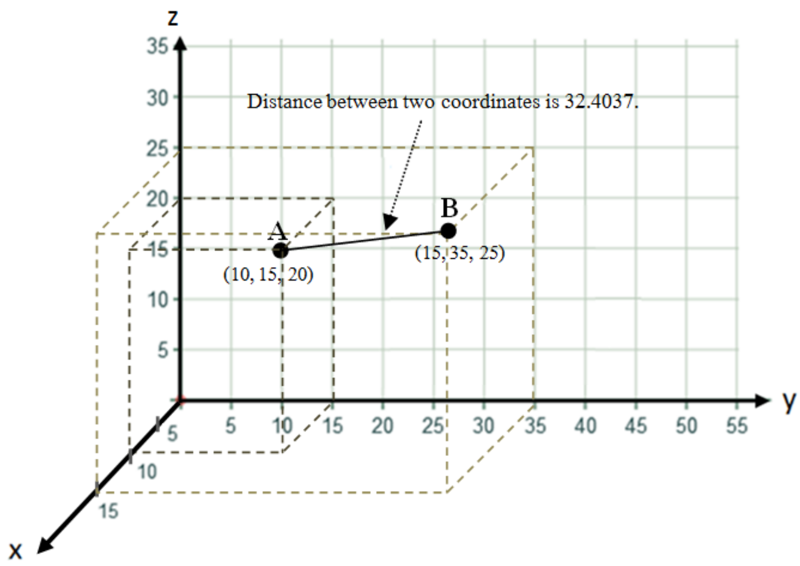


Figure 2.8 Cartesian coordinate system in three-dimension

In mathematics, a three-dimensional vector (x, y, z) can be arranged into a column matrix as below:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

It can be represented in a row matrix as follows:

$$[x \quad y \quad z].$$

In three dimensions, there are three basic standards:

$$e_1 = (1,0,0), e_2 = (0,1,0), e_3 = (0,0,1).$$

Therefore, any vector can be represented in terms of these standards. The three-dimensional vector $\vec{\boldsymbol{a}}$ = (x, y, z) can be expressed in terms of the hree basic standards as shown below:

$$\vec{\boldsymbol{a}} = \mathrm{xe}_1 + \mathrm{ye}_2 + \mathrm{ze}_3. \quad \text{or} \qquad \vec{\boldsymbol{a}} = x(1,0,0) + y(0,1,0) + z(0,0,1).$$

Chapter 3

Student Formation by Heterogeneous Grouping (SFHG)

3.1 The Mathematical Approach for Heterogeneous Grouping

In this paper, the proposed approach called Student Formation by Heterogeneous Grouping (SFHG) works on the assumption that every student has different values of attributes which are academic factors and background knowledge. As confirmed by Kutlu (Kutlu, 2012), as every student has individual information and experiences, the heterogeneous groups should comprise of students whose prior educational knowledge should be unequal. Therefore, at the time of forming student groups, teachers can set inquiring question with some sort of association rules for student to answer, thereby obtaining personalities and information directly from students. Searching for an optimized group of students by an exhaustive search is not practical, as it is time-consuming. To allocate heterogeneous students to appropriate groups, we define the mathematical term and its definition as follows. A class contains n students, denoted by $S = \{s_1, s_2, \ldots, s_n\}$. Each student contains exactly m attributes, which are prior grades, student preferences, and pre-test or exercise scores. This can be represented in an m-dimensional vector, denoted by $A_i = (a_{i1}, a_{i2}, \ldots, a_{im})$, where a_{im} is the value of attribute m of student i. For example, for student s_i, the two-dimentional attributes vector is represented as $A_i = (a_{i1}, a_{i2})$. The values of attributes have various kinds of worth. For instance, if the attribute represents the grade received in 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 where possible. In certain cases, we cannot construct groups with the same size. If n students are divided into p smaller groups, the *size* of the group is $\left\lfloor \frac{n}{p} \right\rfloor \leq size \leq \left\lceil \frac{n}{p} \right\rceil$.

Let G denote the whole group 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 $1 \leq m \leq n$. The average value of all students in the same group must be estimated, as this is the attribute of the group. In this paper, the cumulative GPA and the grades of prior programming courses are applied as the student's attributes because they represent the student's performance and educational skill. When the group is completely generated, each attribute of the formed group can be calculated. Consequently, 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 belonging to the group, which is shown in (1).

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

where $a_{f(g_{ik})}$ is the attribute of $f(g_{ik}) = s_m$, $1 \leq m \leq n$.

To achieve the mechanism for fairness and equity, it is necessary to compute the following values.

1. If two students called s_i, s_j belong to the same group, 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.

The graph in Figure 3.1 shows the distance of two students called s_i and s_j, where a two-attribute vector is applied.

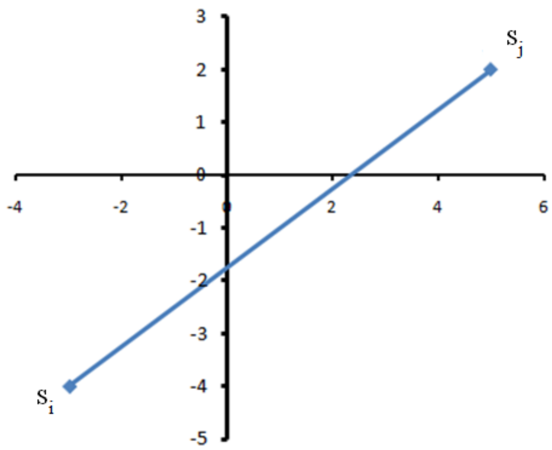


Figure3.1 The distance of students s_i and s_j, where the attribute vector of s_i is $A_i = (a_{i1}, a_{i2})$ and attribute vector of s_i is $A_j = (a_{j1}, a_{j2})$

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

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

, where k is the size of group G_q.

If group G_q has a higher $ED_1(G_q)$ than other groups, this implies that group G_q has mixed with different educational 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:

$$ED_1(G_q) = \sum_{d=1}^{2} \sum_{e=d+1}^{3} \left| g_{qd} - g_{qe} \right| = \left| g_{q1} - g_{q2} \right| + \left| g_{q1} - g_{q3} \right| + \left| g_{q2} - g_{q3} \right|$$

$$= \sqrt{\sum_{t=1}^{m} \left(a_{f(g_{q1})t} - a_{f(g_{q2})t} \right)^2} + \sqrt{\sum_{t=1}^{m} \left(a_{f(g_{q1})t} - a_{f(g_{q3})t} \right)^2} + \sqrt{\sum_{t=1}^{m} \left(a_{f(g_{q2})t} - a_{f(g_{q3})t} \right)^2}$$

$$= \sqrt{\sum_{t=1}^{m} \left(a_{S_1 t} - a_{S_2 t} \right)^2} + \sqrt{\sum_{t=1}^{m} \left(a_{S_1 t} - a_{S_3 t} \right)^2} + \sqrt{\sum_{t=1}^{m} \left(a_{S_2 t} - a_{S_3 t} \right)^2}$$

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

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

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

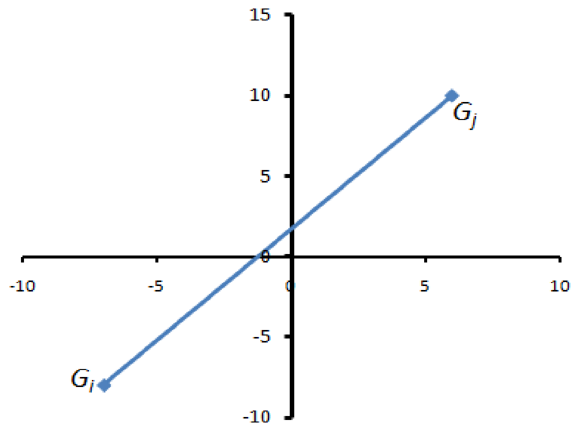


Figure 3.2 The distance of group G_i and G_j, attribute vector of $G_i = (V_{i1}, V_{i2})$ and attribute vector of $G_j = (V_{j1}, V_{j2})$.

If a 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 from the average value of all members in the group which has previously been presented in (1). For instance, the graph of group G_i and G_j is exhibited in Figure 3.2, and the Euclidean distance between the two groups is illustrated below:

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

3.2 Developing Student Formation for Heterogeneous Grouping

3.2.1 Problem Encapsulation

To use a genetic algorithm for forming student groups, we encode our problem as a genome (or chromosome). Regarding our specific problem, a set of n students is denoted by $S = \{s_1, s_2, \ldots, s_n\}$, 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 3.3. If the students are divided into m different groups, the *size* of the group is $\left\lfloor \frac{n}{m} \right\rfloor \le size \le \left\lceil \frac{n}{m} \right\rceil$. Then, the value of each element of the chromosome ranks from 1 to m. For instance, if a class of 12 students is divided into 3 groups, the smallest *size* of each group is $\left\lfloor \frac{12}{3} \right\rfloor = 4$. However, if the class contains 13 students, the biggest group size can be up to $\left\lceil \frac{13}{3} \right\rceil = 5$.

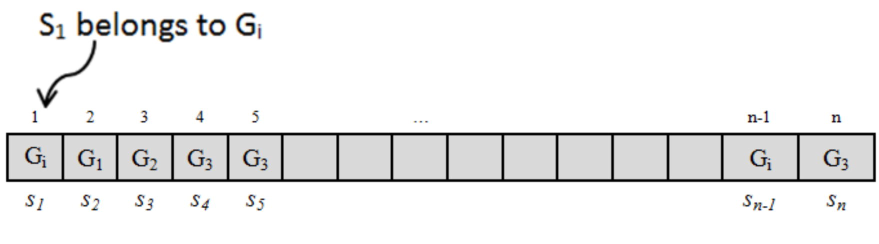


Figure 3.3 The chromosome structure for SFHG

An element of the chromosome can be assigned into any of $\{G_1, G_2, \ldots, G_p\}$. For instance, if a set of students is $\{s_1, s_2, s_3, s_4\, s_5, s_6, s_7\}$, and is separated to form three smaller groups, the smallest group comprises $\left\lfloor \frac{7}{3} \right\rfloor$ = 2 students, and the biggest group can make up to $\left\lceil \frac{7}{3} \right\rceil$= 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 3.4.

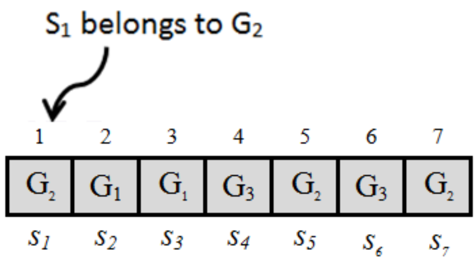


Figure 3.4 The chromosome representing $G_1= \{s_2, s_3\}$, $G_2= \{s_1, s_5, s_7\}$, and $G_3= \{s_4, s_6\}$.

3.2.2 Fitness Function

Generally, GAs are in a class of evolutionary algorithms motivated by natural science (David & Prasad, 2009). They search the space of possible chromosomes in an attempt to find suitable solutions based on the fitness function. The well-defined fitness measure helps solve this problem, which aids the research goal of this paper. As stated earlier, the purpose of the proposed algorithm is 1) to distribute and mix heterogeneous students in the established groups and 2) to achieve a mechanism for fairness and equity of heterogeneity among formed groups, and so multi-objective fitness functions are defined as follows.

1. **First Objection Function**

The first objective function (f_1) illustrated in (5) is the average value of the Euclidean distance of all groups.

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

where p is the total number of generated groups.

f_I is an objective function that is able to guide the algorithm to search for an optimal solution. If f_I is high, it indicates that most groups contain different skills levels of students. In order to reach optimum heterogeneity in each group, we prefer to construct all groups consisting of students that are not of the same kind. Therefore, we expect to produce groups with a high value of f_I. In order to get a clear picture of the objective function, an example is presented in Figure 3.5. Each point represents a student distributed throughout a two-dimensional space.

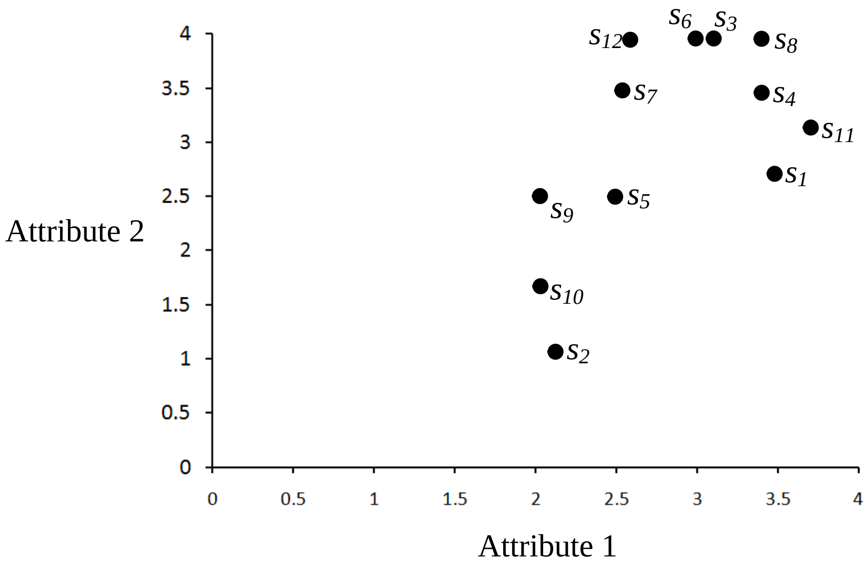


Figure 3.5 12 **s**tudents in a two-dimensional space

For example, assume the students are arranged to form three groups, $G_1 = \{s_3, s_6, s_7, s_{12}\}$, $G_2 = \{s_1, s_4, s_8, s_{11}\}$, and $G_3 = \{s_2, s_5, s_9, s_{10}\}$. Then, the chromosome representing the group formation is displayed in Figure 3.6.

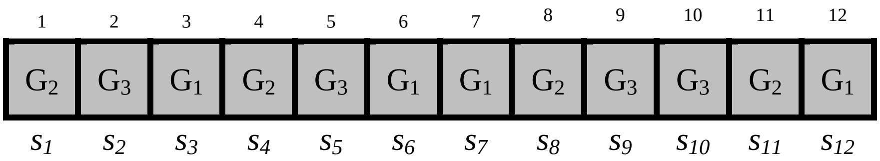


Figure 3.6 Chromosome representing $G_1 = \{s_3, s_6, s_7, s_{12}\}$, $G_2 = \{s_1, s_4, s_8, s_{11}\}$, and $G_3 = \{s_2, s_5, s_9, s_{10}\}$

Therefore, the student vectors shown previously in Figure 3.5 can be represented as in Fig 3.7, which shows where each student belongs to.

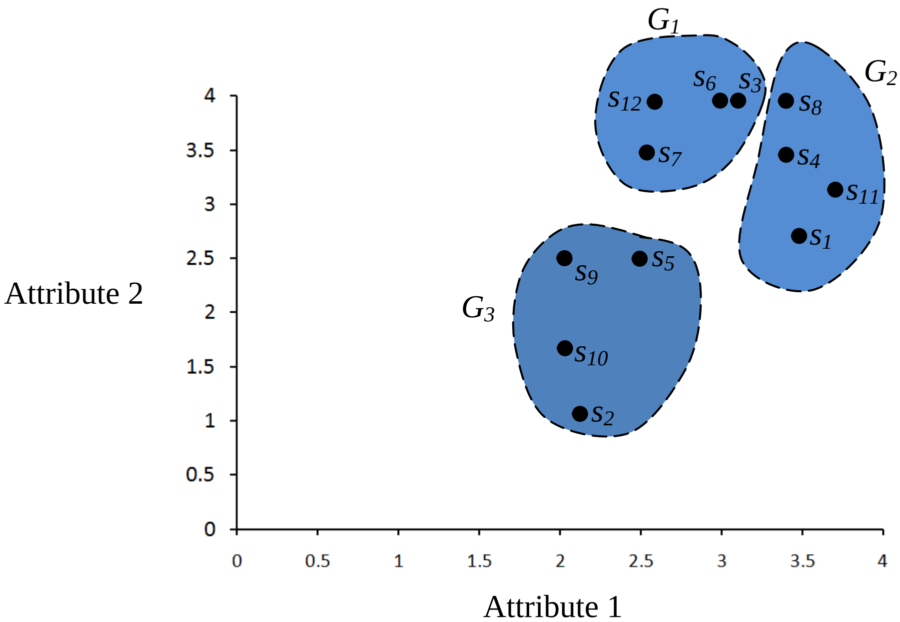


Figure 3.7 12 students in a two-dimensional space

Therefore, the objection function f_1 can be represented as follows:

$$f_1(chromosome) = \frac{\sum_{q=1}^{3} ED_1(G_q)}{p}$$
$$= \frac{ED_1(G_1) + ED_1(G_2) + ED_1(G_3)}{3}.$$

2. Second Objective Function

This function is designed to help the algorithm retain the mechanism of fairness and equity among groups in all attributes, as much as possible. It is related to (4), as it is the Euclidean distance between the two groups. If p groups exist which are created by the algorithm, the second fitness function is demonstrated below:

$$f_2(chromosome) = \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \left| G_i - G_j \right|$$
$$= \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \sqrt{\sum_{k=1}^{m} \left(V_{ik} - V_{jk} \right)^2} \quad (6)$$

where V_{ik} is the attribute k of group i which is calculated by (1), and m is the total number of attributes.

If the value of f_2 is low, it indicates that established groups are more balanced with heterogeneous students. The lower fitness value shows that the chromosome arranges different students into formed groups properly. Low performing students will have little chance to be together and high performing students are assigned to be in different groups. On the other hand, the higher fitness value of f_2 means that all formed groups are poor in distributing heterogeneous students. Weak performing students may have been chosen to stay together in the same group as well as strong performing students. For this reason, our algorithm is required to build the student groups with a low value of f_2.

Based on (1), an attribute value of the groups can be calculated by the average value of all members belonging to the group. Therefore, from Figure 3.7 each group can transform to a point in a two-dimensional space as well, which is displayed in Figure 3.8.

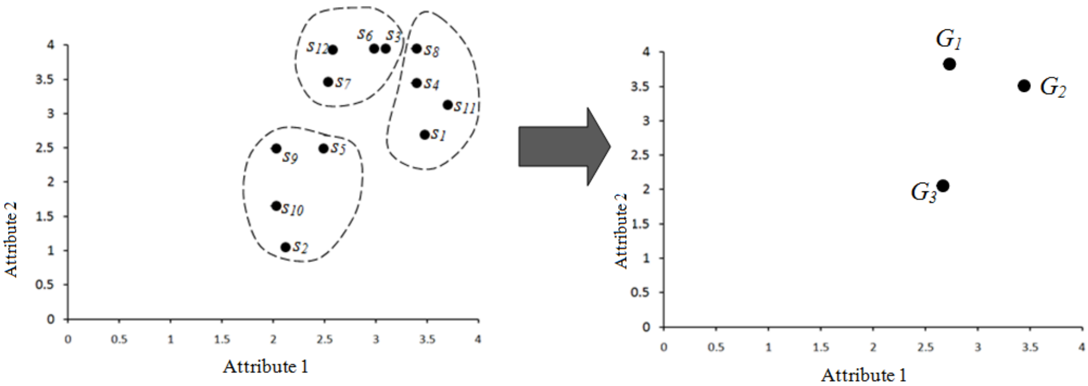


Figure 3.8 Representing groups in two-dimensional space.

As mentioned previously, each attribute of G_i is calculated by the average value of all members in the group as demonstrated below:

$$G_1 = \left(\frac{a_{S_3 1} + a_{S_6 1} + a_{S_7 1} + a_{S_{12} 1}}{4}, \frac{a_{S_3 2} + a_{S_6 2} + a_{S_7 2} + a_{S_{12} 2}}{4} \right),$$

$$G_2 = \left(\frac{a_{S_1 1} + a_{S_4 1} + a_{S_8 1} + a_{S_{11} 1}}{4}, \frac{a_{S_1 2} + a_{S_4 2} + a_{S_8 2} + a_{S_{11} 2}}{4} \right),$$

$$G_3 = \left(\frac{a_{S_2 1} + a_{S_5 1} + a_{S_9 1} + a_{S_{10} 1}}{4}, \frac{a_{S_2 2} + a_{S_5 2} + a_{S_9 2} + a_{S_{10} 2}}{4} \right).$$

When all groups' vectors are identified, the second fitness function (f_2) can be calculated:

$$f_2(chromosome) = \sum_{i=1}^{2} \sum_{j=i+1}^{3} \left| G_i - G_j \right|$$

$$= \sum_{i=1}^{2} \sum_{j=i+1}^{3} \sqrt{\sum_{k=1}^{2} \left(V_{ik} - V_{jk} \right)^2}$$

where V_{ik} is the attribute k of group i which is calculated by (1), and m is the total number of attributes.

In theory, a multi-objective fitness function can be represented by 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. In this paper, maximizing the heterogeneity of the students within the divided groups is required, therefore we need the high value of f_1(x), where x is a chromosome. The higher value of $f_1(x)$ indicates that on average each group is more likely to be heterogeneous. Nevertheless, in order to achieve fairness and equality in the group formation, we need a low value of $f_2(x)$ since the lower value implies that all generated groups are more likely to be equal in all attribute vectors. Regarding our problem in particular, the fitness function of each chromosome can be computed as follows:

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

where *x* is the selected chromosome.

In order to understand the meaning of the fitness function, therefore, both values of $f_1(x)$ and $f_2(x)$ are written in a tabular form to get more meaning foreach, as presented in Table 3.1.

Table 3.1 Fitness function

Fitness function	$f_1(x)$	$f_2(x)$	Characteristic of the generated groups
$f(x) = \frac{f_2(x)}{f_1(x)}$	Low	Low	On average, all groups are likely to be similar, but there is lower heterogeneity of student's attributes in each group.
	Low	High	This is a bad formation because all groups are different, meaning that they are unequal in most attributes and making little use of student's educational skills among formed groups. Additionally, students with the same skills are likely to be arranged into the same groups. Weak performing students

Fitness function	$f_1(x)$	$f_2(x)$	Characteristic of the generated groups
			may have been chosen to stay together in the same group as well as strong performing students.
	High	Low	This is a good formation because generated groups are likely to be similar, encouraging fairness in student's educational skills among formed groups. Each group is likely to compose of students with different educational skills. High performing students will have little chance to be together and low performing students are assigned to be in different groups.
	High	High	Each group composes of high heterogeneity of student's attributes, but it lacks fairness in distribution of student's educational skills among formed groups.

3.2.3 SFHG Operators

Once the chromosome structure and fitness function are completely constructed, SFHG operations must be designed to perform within a single generation. The flowchart of SFHG is presented in Figure 3.9.

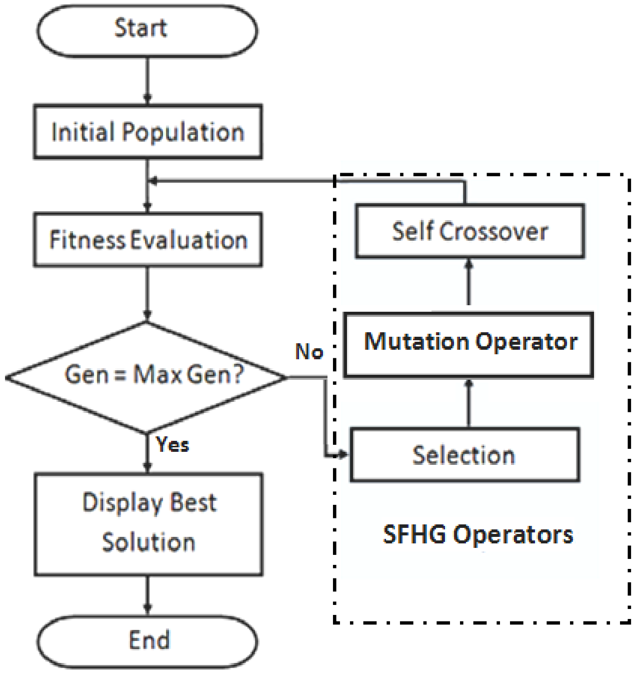


Figure 3.9 SFHG's flowchart

From SFHG's flowchart, the process starts with an initial population of chromosomes by a completely random selection. It should be noted that in our paper the algorithm works on fixed-length character strings. If the number of students is *n*, the fixed-length of character strings for our chromosome is equal to *n* as well.

There are three major operators used in the algorithm.

1. The first operator is a selection operation implemented to duplicate an existing population of the current population to the next generation.
2. The following operator is a mutation operator. Only one parent of the current population is picked in a completely random selection. Next, the algorithm randomly selects a point as a mutation point. A character located in the mutation point is randomly changed. As a result, newborn offspring will be different from the parent. It is required to measure the fitness value of the offspring. An example of the mutation operator applying on a chromosome is illustrated in Figure 3.10. The algorithm works on a fixed-length character string. If we have *n* students to create groups, the length of this chromosome is

n. The group formation derived from the parent is $G_1=\{s_2, s_3\}$, $G_2=\{s_1, s_5, s_7\}$, and $G_3=\{s_4, s_6\}$, but the offspring slightly changes from its parent to create a new student formation as $G_1=\{s_3\}$, $G_2=\{s_1, s_5, s_7\}$, and $G_3=\{s_2, s_4, s_6\}$. The operator helps in maintaining the diversity in the population, as mutation causes the chromosomes of offspring to be different from its parent.

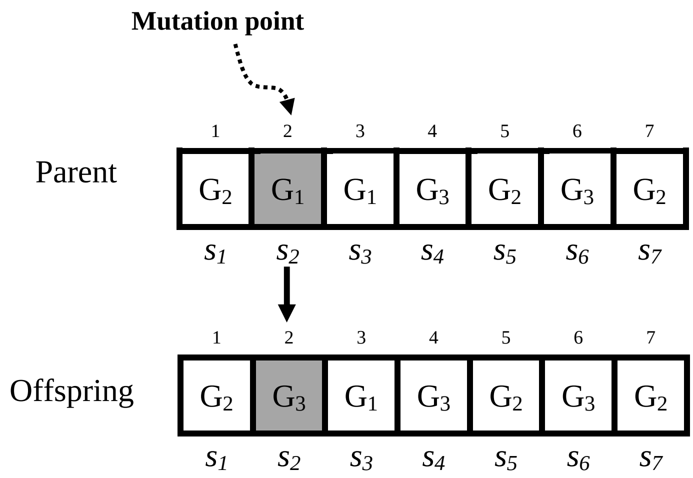


Figure 3.10 An example of mutation operator for SFHG

3. The last operator is a self-crossover operator that helps maintain the diversity in the population, as crossover causes the chromosomes of offspring to be different from their parents. This 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 their selves in order to yield offspring for the next generation (Kuppuswami, 1012). An example of the crossover operation applied on a randomly selected chromosome is depicted in Figure 3.11. A parent represents the groups where $G_1=\{s_2, s_3\}$, $G_2=\{s_1, s_5, s_7\}$, and $g_3=\{s_4, s_6\}$. Then, two selected portions of the parent are interchanged establishing an offspring where $G_1= \{s_5, s_6\}$, $G_2 = \{s_1, s_2, s_7\}$, and $G_3= \{s_3, s_4\}$.

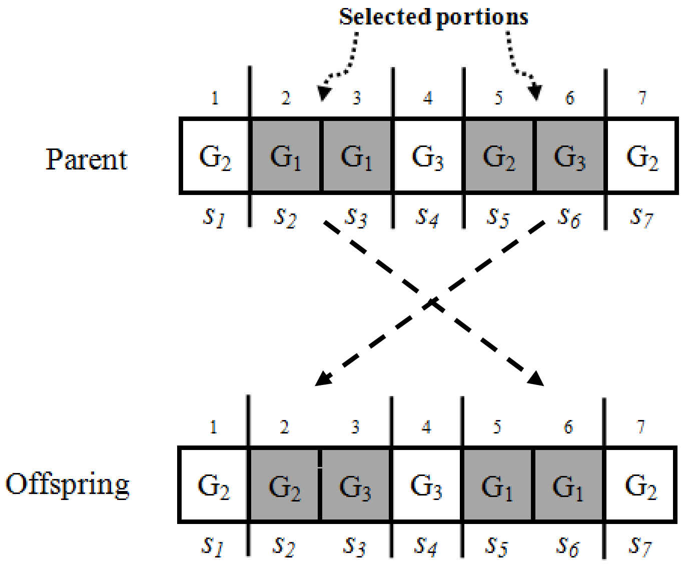


Figure 3.11 An example of self-crossover operator for SFHG

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 completely random selection. Conversely, in our approach, the initial population of chromosomes must be qualified by some regulations, such as the total number of groups and the group size. In each generation, an outperforming 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. New populations gathered from the previous generations are utilized to create subsequent populations and so on. Finally, the algorithm achieves the best solution that is closest to the optimum solution.

3.2.4 Examples of chromosome evaluation

Suppose that a class of 12 students is divided into three groups which are $G_1 = \{s_1, s_3, s_4, s_5\}$, $G_2 = \{s_2, s_7, s_9, s_{10}\}$, and $G_3 = \{s_6, s_8, s_{11}, s_{12}\}$. The chromosome named $chromosome_1$ can encode the group structure as shown in Figure 3.12.

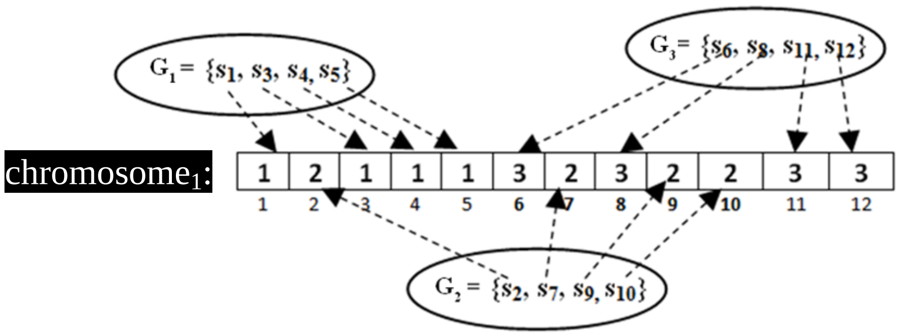


Figure 3.12 Chromosome$_1$ represents $G_1 = \{s_1, s_3, s_4, s_5\}$, $G_2 = \{s_2, s_7, s_9, s_{10}\}$, and $G_3 = \{s_6, s_8, s_{11}, s_{12}\}$

However, if the 12 students rearrange to form new groups where $G_1=\{s_5, s_6, s_7, s_8\}$, $G_2=\{s_1, s_2, s_3, s_4\}$, and $G_3=\{s_9, s_{10}, s_{11}, s_{12}\}$, the group formation can be encoded in a chromosome called chromosome$_2$ represented in Figure 3.13.

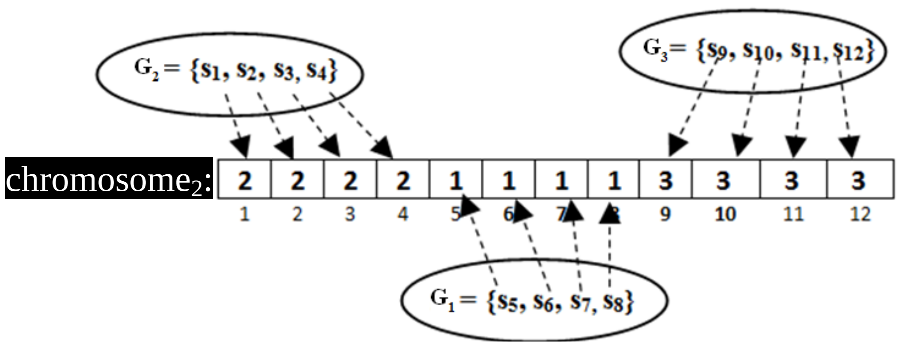


Figure 3.13 Chromosome$_2$ represents $G_1=\{s_5, s_6, s_7, s_8\}$, $G_2=\{s_1, s_2, s_3, s_4\}$, and $G_3=\{s_9, s_{10}, s_{11}, s_{12}\}$

Let student s_i be associated with the multi-attributes vector $A_{s_i} = (a_{s_i1}, a_{s_i2}, \ldots, a_{s_ip})$, where p is the maximum number of attributes. If a two-attributes vector is applied, a vector of student S_i is denoted $A_{s_i} = (a_{s_i 1}, a_{s_i 2})$ where $a_{s_i 1}$ is the student's GPA and $a_{s_i 2}$ is a previous grade of a prerequisite course. Suppose all vectors of 12 students are demonstrated in Figure 3.14.

$$\begin{cases} A_{s_1} = (3.50, C+), \\ A_{s_2} = (2.14, D), \\ A_{s_3} = (3.15, A), \\ A_{s_4} = (3.40, B+), \\ A_{s_5} = (2.50, C+), \\ A_{s_6} = (3.14, A), \\ A_{s_7} = (2.59, B+), \\ A_{s_8} = (3.40, A), \\ A_{s_9} = (1.98, C+), \\ A_{s_{10}} = (2.05, D+), \\ A_{s_{11}} = (3.65, B), \\ A_{s_{12}} = (2.65, A), \end{cases}$$

Figure 3.14 Attribute vector of 12 students, $A_{s_i} = (student's\,GPA,\ grade\ of\ prerequiste\ course)$

Let chromosome_i comprises of G_1, G_2, G_2 and so on. From this, we can calculate the fitness value of chromosome_1 and chromosome_2 as shown below.

1) The calculation of fitness value for chromosome$_1$:

Adapted from (3), the Euclidean distance (ED_1) of each group is calculated as the following equation.

$$ED_1(G_q) = \sum_{d=1}^{k-1} \sum_{e=d+1}^{k} \left| g_{qd} - g_{qe} \right|$$

,where q = 1,2,3 and k is the size of group G_q.

Then,

$$
\begin{aligned}
ED_1(G_q) &= \sum_{d=1}^{3} \sum_{e=d+1}^{4} \left| g_{qd} - g_{qe} \right| \\
&= \left| g_{q1} - g_{q2} \right| + \left| g_{q1} - g_{q3} \right| + \left| g_{q1} - g_{q4} \right| + \left| g_{q2} - g_{q3} \right| + \left| g_{q2} - g_{q4} \right| + \left| g_{q3} - g_{q4} \right| \\
&= \sqrt{\sum_{t=1}^{m} \left(a_{f(g_{q1})t} - a_{f(g_{q2})t} \right)^2} + \sqrt{\sum_{t=1}^{m} \left(a_{f(g_{q1})t} - a_{f(g_{q3})t} \right)^2} + \sqrt{\sum_{t=1}^{m} \left(a_{f(g_{q1})t} - a_{f(g_{q4})t} \right)^2} + \\
&\quad \sqrt{\sum_{t=1}^{m} \left(a_{f(g_{q2})t} - a_{f(g_{q3})t} \right)^2} + \sqrt{\sum_{t=1}^{m} \left(a_{f(g_{q2})t} - a_{f(g_{q4})t} \right)^2} + \sqrt{\sum_{t=1}^{m} \left(a_{f(g_{q3})t} - a_{f(g_{q4})t} \right)^2}
\end{aligned}
$$

where *m* is the number of student's attributes, which is equal to 2.

Since the chromosome$_1$ comprises of $G_1 = \{s_1, s_3, s_4, s_5\}$, $G_2 = \{s_2, s_7, s_9, s_{10}\}$, and $G_3 = \{s_6, s_8, s_{11}, s_{12}\}$, we can see that

for $q = 1$

$$f(g_{11}) \rightarrow s_1,\ f(g_{12}) \rightarrow s_3,\ f(g_{13}) \rightarrow s_4,\ f(g_{14}) \rightarrow s_5;$$

for $q = 2$

$$f(g_{21}) \rightarrow s_2,\ f(g_{22}) \rightarrow s_7,\ f(g_{23}) \rightarrow s_9,\ f(g_{24}) \rightarrow s_{10};$$

for $q = 3$

$$f(g_{31}) \rightarrow s_6,\ f(g_{32}) \rightarrow s_8,\ f(g_{33}) \rightarrow s_{11},\ f(g_{34}) \rightarrow s_{12}.$$

Then, the calculation of $ED_1(G_q)$, where q = 1, 2, 3, can be shown as below.

$$
\begin{aligned}
ED_1(G_1) &= \sqrt{\left(a_{S_1 1} - a_{S_3 1}\right)^2 + \left(a_{S_1 2} - a_{S_3 2}\right)^2} + \sqrt{\left(a_{S_1 1} - a_{S_4 1}\right)^2 + \left(a_{S_1 2} - a_{S_4 2}\right)^2} + \sqrt{\left(a_{S_1 1} - a_{S_5 1}\right)^2 + \left(a_{S_1 2} - a_{S_5 2}\right)^2} + \\
&\quad \sqrt{\left(a_{S_3 1} - a_{S_4 1}\right)^2 + \left(a_{S_3 2} - a_{S_4 2}\right)^2} + \sqrt{\left(a_{S_3 1} - a_{S_5 1}\right)^2 + \left(a_{S_3 2} - a_{S_5 2}\right)^2} + \sqrt{\left(a_{S_4 1} - a_{S_5 1}\right)^2 + \left(a_{S_4 2} - a_{S_5 2}\right)^2} \\
&= \sqrt{(3.50 - 3.15)^2 + \left(C^+ - A\right)^2} + \sqrt{(3.50 - 3.40)^2 + \left(C^+ - B^+\right)^2} + \sqrt{(3.50 - 2.50)^2 + \left(C^+ - C^+\right)^2} + \\
&\quad \sqrt{(3.15 - 3.40)^2 + \left(A - B^+\right)^2} + \sqrt{(3.15 - 2.50)^2 + \left(A - C^+\right)^2} + \sqrt{(3.40 - 2.50)^2 + \left(B^+ - C^+\right)^2}
\end{aligned}
$$

Let A=4, B+=3.5, B=3,… , D+=1.5, D=1, F=0, and others, such as W (Withdrawal) and I (Incomplete), are equal to zero.

$$ED_1(G_1) = \sqrt{(3.50-3.15)^2+(2.5-4)^2} + \sqrt{(3.50-3.40)^2+(2.5-3.5)^2} + \sqrt{(3.50-2.50)^2+(2.5-2.5)^2} + \sqrt{(3.15-3.40)^2+(4-3.5)^2} + \sqrt{(3.15-2.50)^2+(4-2.5)^2} + \sqrt{(3.40-2.50)^2+(3.5-2.5)^2}$$
$$= 6.83$$

$$ED_1(G_2) = \sqrt{(a_{S_2 1}-a_{S_7 1})^2+(a_{S_2 2}-a_{S_7 2})^2} + \sqrt{(a_{S_2 1}-a_{S_9 1})^2+(a_{S_2 2}-a_{S_9 2})^2} + \sqrt{(a_{S_2 1}-a_{S_{10} 1})^2+(a_{S_2 2}-a_{S_{10} 2})^2} + \sqrt{(a_{S_7 1}-a_{S_9 1})^2+(a_{S_7 2}-a_{S_9 2})^2} + \sqrt{(a_{S_7 1}-a_{S_{10} 1})^2+(a_{S_7 2}-a_{S_{10} 2})^2} + \sqrt{(a_{S_9 1}-a_{S_{10} 1})^2+(a_{S_9 2}-a_{S_{10} 2})^2}$$
$$= \sqrt{(2.14-2.59)^2+(D-B^+)^2} + \sqrt{(2.14-1.98)^2+(D-C^+)^2} + \sqrt{(2.14-2.05)^2+(D-D^+)^2} + \sqrt{(2.59-1.98)^2+(B^+-C^+)^2} + \sqrt{(2.59-2.05)^2+(B^+-D^+)^2} + \sqrt{(1.98-2.05)^2+(C^+-D^+)^2}$$
$$= \sqrt{(2.14-2.59)^2+(1-3.5)^2} + \sqrt{(2.14-1.98)^2+(1-2.5)^2} + \sqrt{(2.14-2.05)^2+(1-1.5)^2} + \sqrt{(2.59-1.98)^2+(3.5-2.5)^2} + \sqrt{(2.59-2.05)^2+(3.5-1.5)^2} + \sqrt{(1.98-2.05)^2+(2.5-1.5)^2}$$
$$= 8.80$$

$$ED_1(G_3) = \sqrt{(a_{S_6 1}-a_{S_8 1})^2+(a_{S_6 2}-a_{S_8 2})^2} + \sqrt{(a_{S_6 1}-a_{S_{11} 1})^2+(a_{S_6 2}-a_{S_{11} 2})^2} + \sqrt{(a_{S_6 1}-a_{S_{12} 1})^2+(a_{S_6 2}-a_{S_{12} 2})^2} + \sqrt{(a_{S_8 1}-a_{S_{11} 1})^2+(a_{S_8 2}-a_{S_{11} 2})^2} + \sqrt{(a_{S_8 1}-a_{S_{12} 1})^2+(a_{S_8 2}-a_{S_{12} 2})^2} + \sqrt{(a_{S_{11} 1}-a_{S_{12} 1})^2+(a_{S_{11} 2}-a_{S_{12} 2})^2}$$
$$= \sqrt{(3.14-3.40)^2+(A-A)^2} + \sqrt{(3.14-3.65)^2+(A-B)^2} + \sqrt{(3.14-2.65)^2+(A-A)^2} + \sqrt{(3.40-3.65)^2+(A-B)^2} + \sqrt{(3.40-2.65)^2+(A-A)^2} + \sqrt{(3.65-2.65)^2+(B-A)^2}$$
$$= \sqrt{(3.14-3.40)^2+(4-4)^2} + \sqrt{(3.14-3.65)^2+(4-3)^2} + \sqrt{(3.14-2.65)^2+(4-4)^2} + \sqrt{(3.40-3.65)^2+(4-3)^2} + \sqrt{(3.40-2.65)^2+(4-4)^2} + \sqrt{(3.65-2.65)^2+(3-4)^2}$$
$$= 5.057$$

Finally, the value of f_l(chromosome$_1$) can be calculated as below:

$$f_1(chromosome_1) = \frac{\sum_{q=1}^{3} ED_1(G_q)}{3} = \frac{6.83+8.80+5.057}{3} = 6.90$$

Based on the attributes shown in Figure 3.15, the average value of the first attribute (GPA) of G_1 = {s_1, s_3, s_4, s_5} is

$$V_{11} = \frac{3.50 + 3.15 + 3.40 + 2.50}{4} = 3.14$$

And, the average value of the second attribute (previous grade of a prerequisite course) of the group G_1 is

$$V_{12} = \frac{C^{+} + A + B^{+} + C^{+}}{3} = \frac{2.5 + 4.0 + 3.5 + 2.5}{3} = 3.125.$$

where A=4, B+=3.5, B=3,… , D+=1.5, D=1, F=0, and others, such as W (Withdrawal) and I (Incomplete), are equal to zero.

Consequently, the vector attribute of G_1 is represented in the form of (V_{11}, V_{12}) = (3.14, 3.125). Conducted in the same way, the calculation of the vector attributes of $G_2 = \{s_2, s_7, s_9, s_{10}\}$ can be shown as

$$V_{21} = \frac{2.14 + 2.59 + 1.98 + 2.05}{4} = 2.19$$

$$V_{22} = \frac{D + B^{+} + C^{+} + D^{+}}{4} = \frac{1 + 3.5 + 2.5 + 1.5}{4} = 2.125.$$

Therefore, the vector attribute of G_2 is also represented in the form of (V_{21}, V_{22}) = (2.19, 2.125). Finally, the last group is calculated and the vector attribute of the third group ($G_3 = \{s_6, s_8, s_{11}, s_{12}\}$) is

$$V_{31} = \frac{3.14 + 3.40 + 3.65 + 2.65}{4} = 3.21$$

$$V_{32} = \frac{A + A + B + A}{4} = \frac{4 + 4 + 3 + 4}{4} = 3.75.$$

The vector attribute of G_3 is $(V_{31}, V_{32}) = (3.21, 3.75)$.

Finally, $G_1 = (3.14, 3.125)$, $G_2 = (2.19, 2.125)$, and $G_3 = (3.21, 3.75)$, which can be represented in the two-dimensional space as in Figure 3.15.

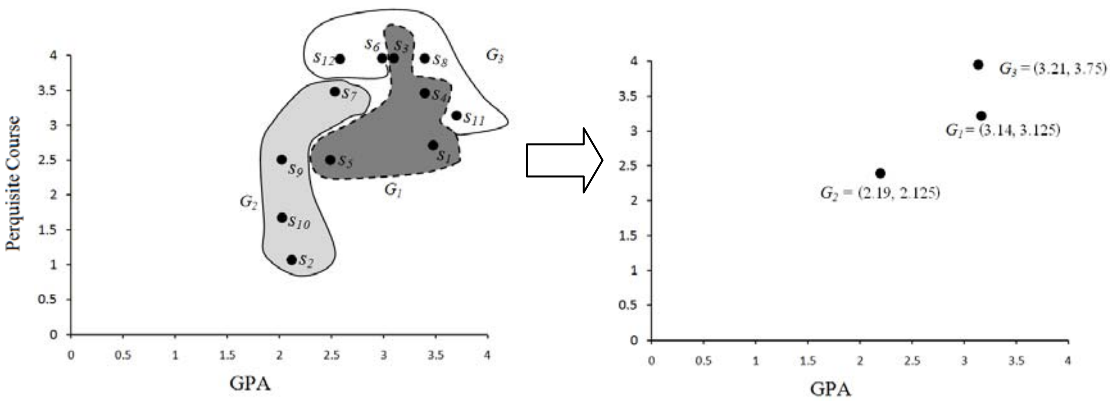


Figure 3. 15 Students and generated groups of chromosome$_1$ represented in a 2-dimensional space

The calculation of the fitness value for chromosome$_1$ are presented below.

$$f_2(chromosome_1) = \sum_{k=1}^{2} \sum_{j=k+1}^{3} \sqrt{\sum_{i=1}^{2} (V_{ki} - V_{ji})^2}$$

$$= \sqrt{(V_{11} - V_{21})^2 + (V_{12} - V_{22})^2} + \sqrt{(V_{11} - V_{31})^2 + (V_{12} - V_{32})^2} + \sqrt{(V_{21} - V_{31})^2 + (V_{22} - V_{32})^2}$$

$$= \sqrt{(3.14 - 2.19)^2 + (3.125 - 2.125)^2} + \sqrt{(3.14 - 3.21)^2 + (3.125 - 3.75)^2} + \sqrt{(2.19 - 3.21)^2 + (2.125 - 3.75)^2}$$

$$= \sqrt{0.95^2 + 1.0^2} + \sqrt{0.07^2 + 0.625^2} + \sqrt{1.02^2 + 1.625^2}$$

$$= 0.3793 + 0.6289 + 1.9186$$

$$\approx 2.93$$

Based on (7), the fitness value of chomrosome$_1$ can be estimated as follows:

$$f(chromosome_1) = \frac{f_2(chromosome_1)}{f_1(chromosome_1)} = \frac{2.93}{6.90} = 0.42$$

2) The calculation of fitness value for chromosome$_2$:

We start with the calculation of the Euclidean distance (ED_1) of each group. Since the chromosome$_2$ comprises of $G_1 = \{s_5, s_6, s_7, s_8\}$, $G_2 = \{s_1, s_2, s_3, s_4\}$, and $G_3 = \{s_9, s_{10}, s_{11}, s_{12}\}$, the calculation of $ED_1(G_q)$, where q = 1, 2, 3, can be shown as below.

$$\begin{aligned}
ED_1(G_1) &= \sqrt{(a_{S_5 1}-a_{S_6 1})^2+(a_{S_5 2}-a_{S_6 2})^2}+\sqrt{(a_{S_5 1}-a_{S_7 1})^2+(a_{S_5 2}-a_{S_7 2})^2}+\sqrt{(a_{S_5 1}-a_{S_8 1})^2+(a_{S_5 2}-a_{S_8 2})^2}+\\
&\quad \sqrt{(a_{S_6 1}-a_{S_7 1})^2+(a_{S_6 2}-a_{S_7 2})^2}+\sqrt{(a_{S_6 1}-a_{S_8 1})^2+(a_{S_6 2}-a_{S_8 2})^2}+\sqrt{(a_{S_7 1}-a_{S_8 1})^2+(a_{S_7 2}-a_{S_8 2})^2}\\
&= \sqrt{(2.50-3.14)^2+(C^+-A)^2}+\sqrt{(2.50-2.59)^2+(C^+-B^+)^2}+\sqrt{(2.50-3.40)^2+(C^+-A)^2}+\\
&\quad \sqrt{(3.14-2.59)^2+(A-B^+)^2}+\sqrt{(3.14-3.40)^2+(A-A)^2}+\sqrt{(2.59-3.40)^2+(B^+-A)^2}\\
&= \sqrt{(2.50-3.14)^2+(2.5-4)^2}+\sqrt{(2.50-2.59)^2+(2.5-3.5)^2}+\sqrt{(2.50-3.40)^2+(2.5-4)^2}+\\
&\quad \sqrt{(3.14-2.59)^2+(4-3.5)^2}+\sqrt{(3.14-3.40)^2+(4-4)^2}+\sqrt{(2.59-3.40)^2+(3.5-4)^2}\\
&= 6.34
\end{aligned}$$

$$\begin{aligned}
ED_1(G_2) &= \sqrt{(a_{S_1 1}-a_{S_2 1})^2+(a_{S_1 2}-a_{S_2 2})^2}+\sqrt{(a_{S_1 1}-a_{S_3 1})^2+(a_{S_1 2}-a_{S_3 2})^2}+\sqrt{(a_{S_1 1}-a_{S_4 1})^2+(a_{S_1 2}-a_{S_4 2})^2}+\\
&\quad \sqrt{(a_{S_2 1}-a_{S_3 1})^2+(a_{S_2 2}-a_{S_3 2})^2}+\sqrt{(a_{S_2 1}-a_{S_4 1})^2+(a_{S_2 2}-a_{S_4 2})^2}+\sqrt{(a_{S_3 1}-a_{S_4 1})^2+(a_{S_3 2}-a_{S_4 2})^2}\\
&= \sqrt{(3.50-2.14)^2+(C^+-D)^2}+\sqrt{(3.50-3.15)^2+(C^+-A)^2}+\sqrt{(3.50-3.40)^2+(C^+-B^+)^2}+\\
&\quad \sqrt{(2.14-3.15)^2+(D-A)^2}+\sqrt{(2.14-3.40)^2+(D-B^+)^2}+\sqrt{(3.15-3.40)^2+(A-B^+)^2}\\
&= \sqrt{(3.50-2.14)^2+(2.5-1)^2}+\sqrt{(3.50-3.15)^2+(2.5-4)^2}+\sqrt{(3.50-3.40)^2+(2.5-3.5)^2}+\\
&\quad \sqrt{(2.14-3.15)^2+(1-4)^2}+\sqrt{(2.14-3.40)^2+(1-3.5)^2}+\sqrt{(3.15-3.40)^2+(4-3.5)^2}\\
&= 11.47
\end{aligned}$$

$$ED_1(G_3)=\sqrt{\left(a_{S_9 1}-a_{S_{10} 1}\right)^2+\left(a_{S_9 2}-a_{S_{10} 2}\right)^2}+\sqrt{\left(a_{S_9 1}-a_{S_{11} 1}\right)^2+\left(a_{S_9 2}-a_{S_{11} 2}\right)^2}+\sqrt{\left(a_{S_9 1}-a_{S_{12} 1}\right)^2+\left(a_{S_9 2}-a_{S_{12} 2}\right)^2}+$$
$$\sqrt{\left(a_{S_{10} 1}-a_{S_{11} 1}\right)^2+\left(a_{S_{10} 2}-a_{S_{11} 2}\right)^2}+\sqrt{\left(a_{S_{10} 1}-a_{S_{12} 1}\right)^2+\left(a_{S_{10} 2}-a_{S_{12} 2}\right)^2}+\sqrt{\left(a_{S_{11} 1}-a_{S_{12} 1}\right)^2+\left(a_{S_{11} 2}-a_{S_{12} 2}\right)^2}$$
$$=\sqrt{(1.98-2.05)^2+\left(C^+-A\right)^2}+\sqrt{(1.98-3.65)^2+\left(C^+-B\right)^2}+\sqrt{(3.14-2.65)^2+\left(C^+-A\right)^2}+$$
$$\sqrt{(2.05-3.65)^2+\left(D^+-B\right)^2}+\sqrt{(2.05-2.65)^2+\left(D^+-A\right)^2}+\sqrt{(3.65-2.65)^2+(B-A)^2}$$
$$=\sqrt{(1.98-2.05)^2+(2.5-4)^2}+\sqrt{(1.98-3.65)^2+(2.5-3)^2}+\sqrt{(3.14-2.65)^2+(2.5-4)^2}+$$
$$\sqrt{(2.05-3.65)^2+(1.5-3)^2}+\sqrt{(2.05-2.65)^2+(1.5-4)^2}+\sqrt{(3.65-2.65)^2+(3-4)^2}$$
$$=11.00$$

Finally, the value of f_I(chromosome$_2$) can be established as below:

$$f_1(chromosome_2)=\frac{\sum_{q=1}^{3} ED_1(G_q)}{3}$$
$$=\frac{6.34+11.47+11.0}{3}$$
$$=9.60\ .$$

Next, each vector for G_i, where i =1, 2, 3, can be calculated as below and can be represented in two-dimensional space as presented in Figure 3.16.

$$G_1=(V_{11},V_{12})$$
$$=\left(\frac{2.50+3.14+2.59+3.40}{4},\frac{C^++A+B^++A}{4}\right)$$
$$=(2.9075,3.5)$$

$$G_2=(V_{21},V_{22})$$
$$=\left(\frac{3.50+2.14+3.15+3.40}{4},\frac{C^++D+A+B^+}{4}\right)$$
$$=(3.0475,2.75)$$

$$G_3=(V_{31},V_{32})$$
$$=\left(\frac{1.98+2.05+3.65+2.65}{4},\frac{C^++D^++B+A}{4}\right)$$
$$=(2.5825,2.75)$$

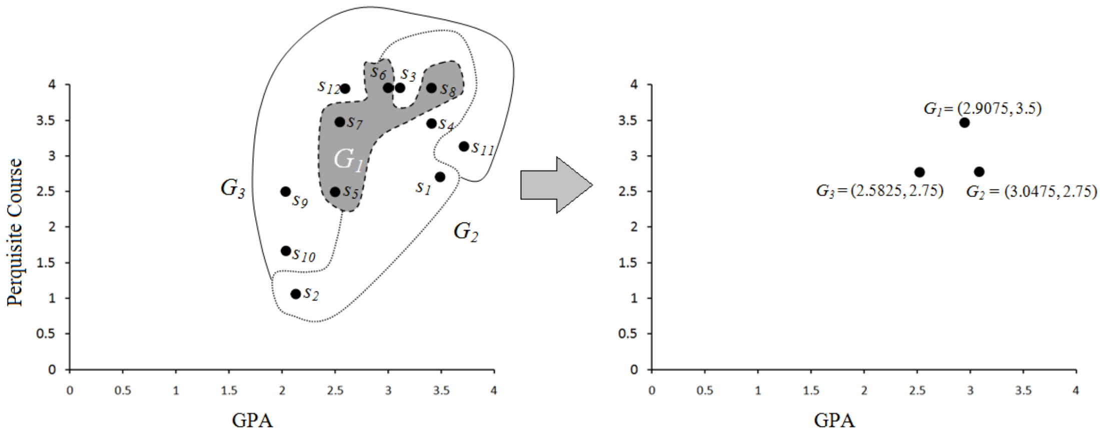


Figure 3.16 Students and generated groups of chromosome_2 represented in a two-dimensional space

Then, the calculation of $f_2(\text{chromosome}_2)$ is presented below.

$$
\begin{aligned}
f_2(\text{chromosome}_2) &= \sum_{k=1}^{2}\sum_{j=k+1}^{3}\sqrt{\sum_{i=1}^{2}(V_{ki}-V_{ji})^2} \\
&= \sqrt{(V_{11}-V_{21})^2+(V_{12}-V_{22})^2}+\sqrt{(V_{11}-V_{31})^2+(V_{12}-V_{32})^2}+\sqrt{(V_{21}-V_{31})^2+(V_{22}-V_{32})^2} \\
&= \sqrt{(2.9075-3.0475)^2+(3.5-2.75)^2}+\sqrt{(2.9075-2.5825)^2+(3.5-2.75)^2}+ \\
&\quad \sqrt{(3.0475-2.5825)^2+(2.75-2.75)^2} \\
&= \sqrt{0.14^2+0.75^2}+\sqrt{0.325^2+0.75^2}+\sqrt{0.465^2+0^2} \\
&= 0.7629+0.8174+0.465 \\
&\approx 2.05
\end{aligned}
$$

Finally, Based on (7), the fitness value of chromosome_2 can be estimated as follows:

$$
\begin{aligned}
f(\text{chromosome}_2) &= \frac{f_2(\text{chromosome}_2)}{f_1(\text{chromosome}_2)} \\
&= \frac{2.05}{9.60} \\
&= 0.21
\end{aligned}
$$

When the parent chromosomes are evaluated, we can see that $f(chromosome_1) < f(chromosome_2)$. This indicates that $chromosome_2$ is more likely to construct a better coalition than $chromosome_1$. Following this, a reproduction operator and a crossover operator are implemented on the current population to construct a new population in a single generation. The reproduction operator is able to allow productive parents to survive based on the fitness function and reproduce. This new population, gathered from the previous generation, is utilized to create subsequent populations and so on. Finally, the SFHG achieves the best solution that is closest to the optimum solution.

Chapter 4

Web-base Application of Student Formation by Heterogeneous Grouping (SFHG)

In this section we present the user interface for our application named Student Formation Heterogeneous Grouping (SFHG). The application is able to be accessed via the internet using a web browser as shown in Figure 4.1. The web application is written using the ASP. The aim of this software development is to help faculties in the School of Science and Technology, Bangkok University, to form student groups with little management.

4.1 User requirement and Use Case Diagram

User requirement is one of the most common elements of designing particular application software. This aspect helps us to understand what features should be included and how they should perform. The brief details of the user requirement are presented as follows:

"The system must allow teachers to generate, edit, print, and archive student lists."

"The system must allow teachers to set the criteria for constructing student groups."

"The system must be able to calculate each group's attributes accurately such as the average grade of prior courses, and average GPA, on the teacher's demand"

"The system must be able to compute the group attribute of generated groups shown in (7) accurately."

"The system must provide two methods for teachers to construct the student groups, which are randomized method and student formation by genetic algorithm."

"The system must provide the function for teachers to adjust the formed groups in order to recreate the groups in line with specific objectives."

After the requirements of web-base application of student formation by heterogeneous grouping are well identified, the use case diagram can be designed as it is the simplest representation of a user's interaction with the system shown in Fig 4.1.

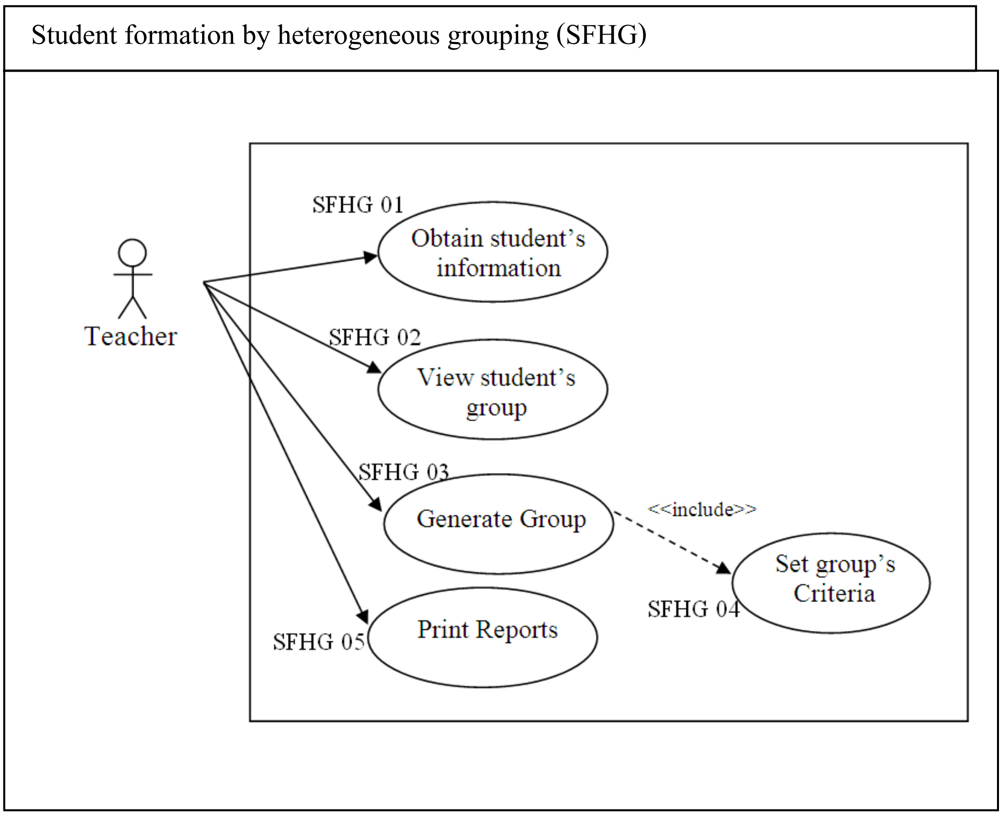


Figure 4.1 Use case Diagram

4.2 Web Site Diagram and User Interfaces

Once the user requirement and use case digam have been defined, the website diagram can be designed. During the design, incompatibilities and conflicts can be found. The user requirement can be reviewed to clarify the problems which have arisen. Figure 4.2 presents the relationships between the webpages that are included in designing SFHG.

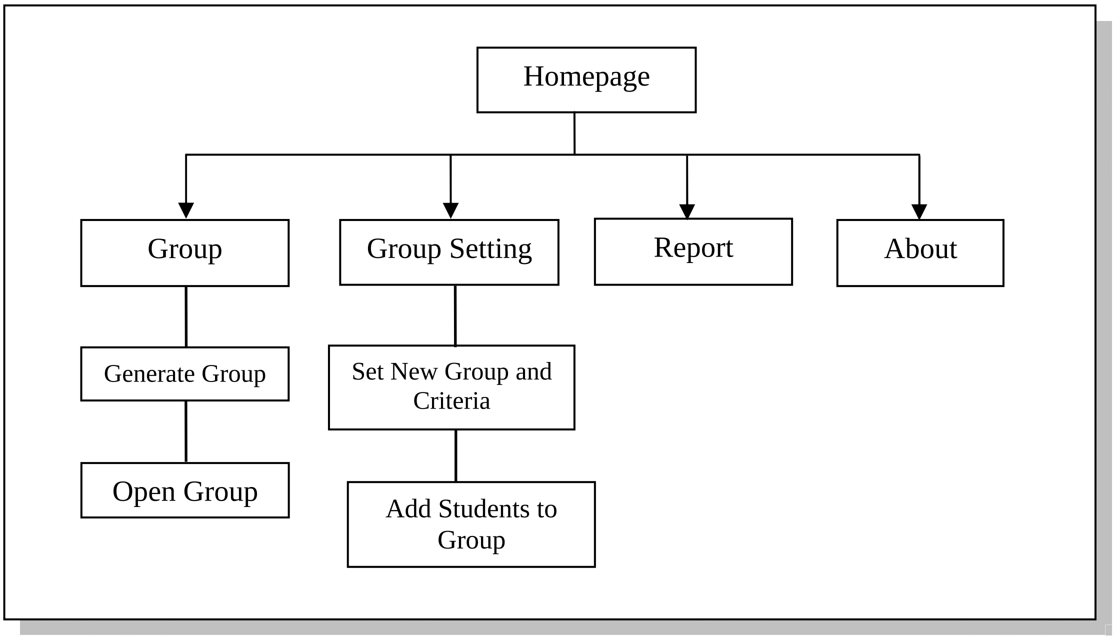


Figure 4.2 Web Site Diagram

The homepage is presented in Figure 4.3, which captures the attention of teachers and creates first impressions. It consists of a set of menus, through which teachers can interact with data related to the student's list, group's setting requirement, course's lists, and reports.

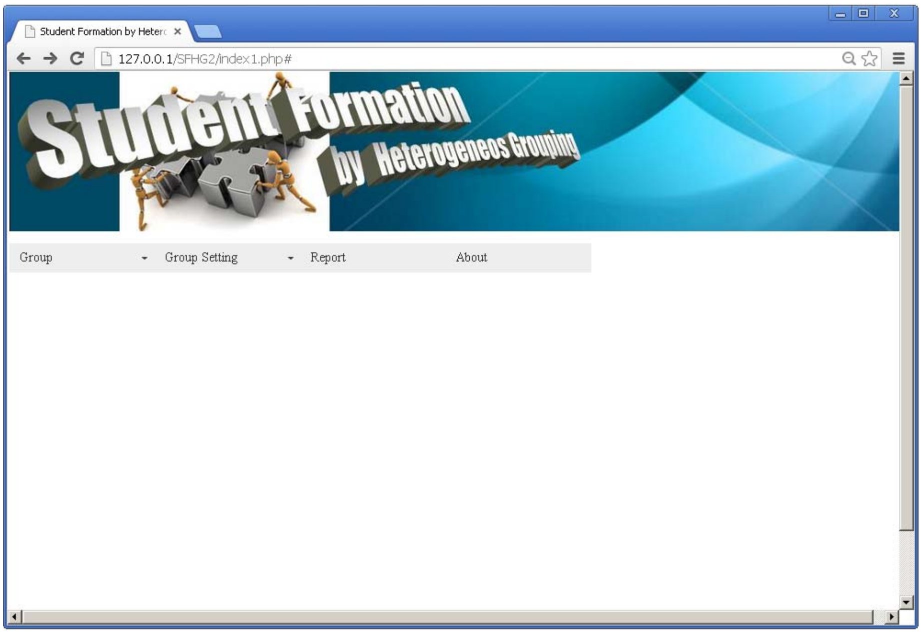


Figure 4.3 Home page of SFHG

The "Group" menu consists of two submenus, which are "Generate Groups" and "Open Group" as presented in Figure 4.4.

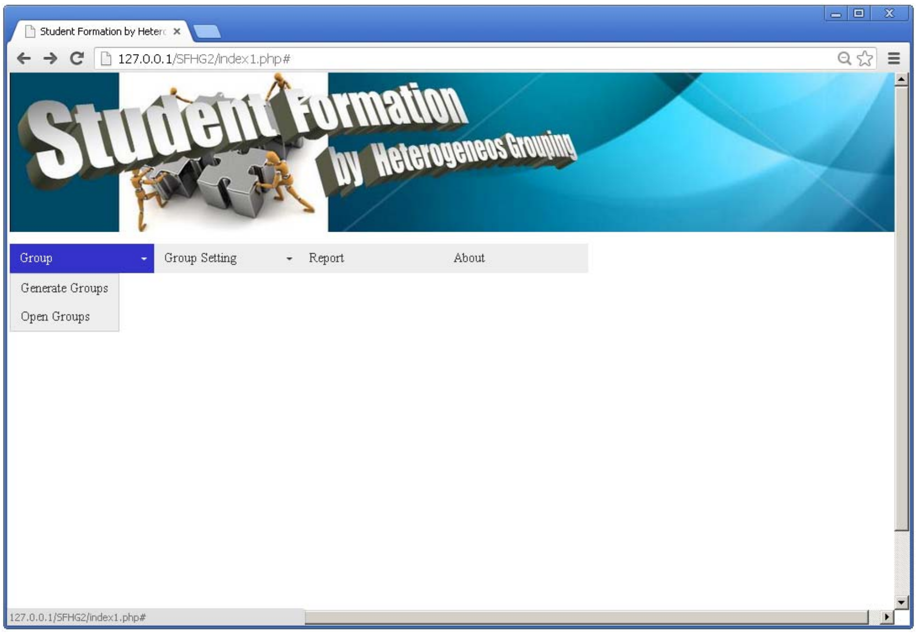


Figure 4.4 "Group" menu and submenus

The “Generate Groups” submenu is designed for users to create student groups while the “Open Groups” submenu is for the user to open the existing groups for editing and review. As illustrated in Figure 4.5, the “Generate Groups” submenu provides two methods for generating student groups, SFHG and the randomized method. The users select the group id, which has previously been set in the “Group Setting” menu. In addition, student lists together with their attributes, such as student’s GPA and received grades, are required to enter to are required to enter to the system under the “Group Setting” menu. When the group’s attributes and the size of the group are set, we can then click on the process to generate the student groups. The example of generating groups by randomized method is shown in Figure 4.6 (a).

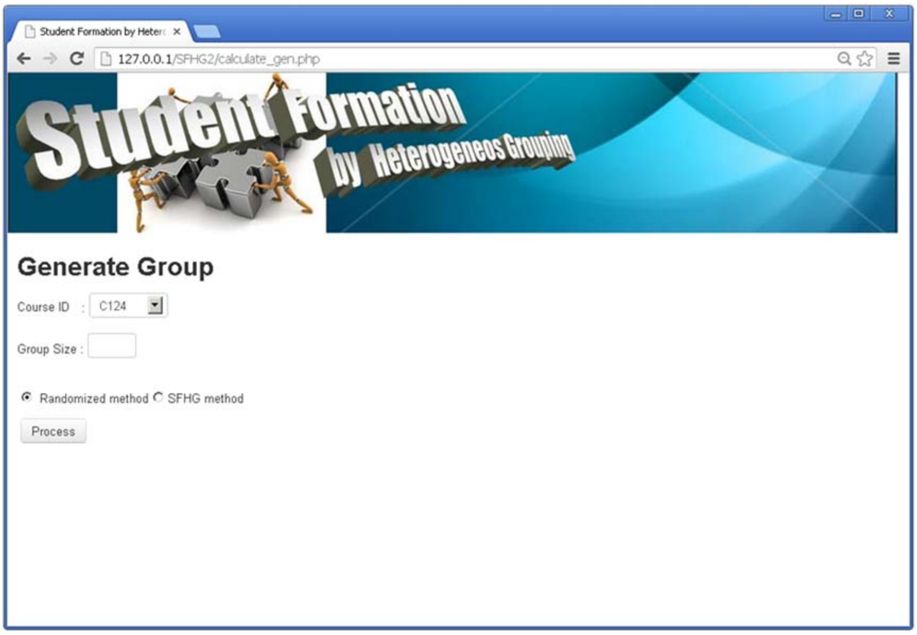
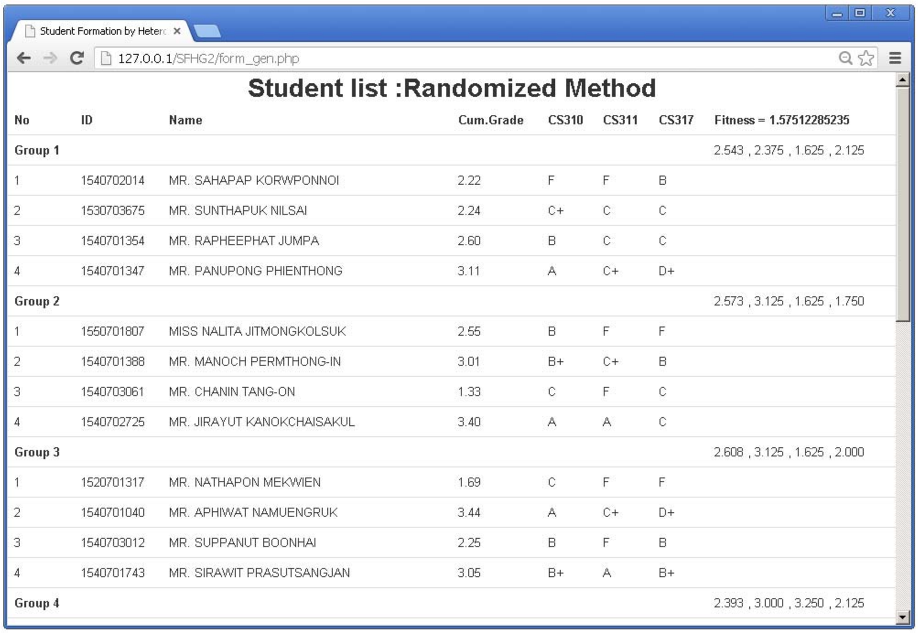


Figure 4.5 Generate Group

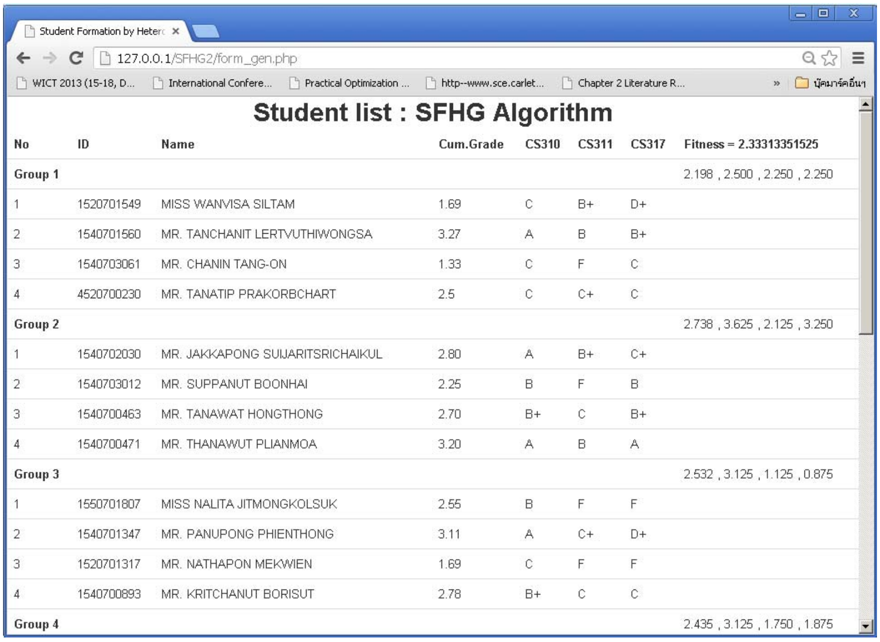
Once the student list is completely entered into the database of the system, the users can click the “Process” button to create the student groups. There are two methods provided for forming student groups, which are randomized method and SFHG. The example result of SFHG method is shown in Figure 4.6(b). The fitness value indicating the quality of the student's coalition is calculated by (5), (6), and (7).

In addition, the system also provides the feature presented in Figure 4.7 for the users to modify the generated groups. The users can relocate certain students of one group to another group. However, if there are any changes at this stage, the system automatically recalculates the fitness value.



No	ID	Name	Cum.Grade	CS310	CS311	CS317	Fitness = 1.57512285235
Group 1							2.543 , 2.375 , 1.625 , 2.125
1	1540702014	MR. SAHAPAP KORWPONNOI	2.22	F	F	B	
2	1530703675	MR. SUNTHAPUK NILSAI	2.24	C+	C	C	
3	1540701354	MR. RAPHEEPHAT JUMPA	2.60	B	C	C	
4	1540701347	MR. PANUPONG PHIENTHONG	3.11	A	C+	D+	
Group 2							2.573 , 3.125 , 1.625 , 1.750
1	1550701807	MISS NALITA JITMONGKOLSUK	2.55	B	F	F	
2	1540701388	MR. MANOCH PERMTHONG-IN	3.01	B+	C+	B	
3	1540703061	MR. CHANIN TANG-ON	1.33	C	F	C	
4	1540702725	MR. JIRAYUT KANOKCHAISAKUL	3.40	A	A	C	
Group 3							2.608 , 3.125 , 1.625 , 2.000
1	1520701317	MR. NATHAPON MEKWIEN	1.69	C	F	F	
2	1540701040	MR. APHIWAT NAMUENGRUK	3.44	A	C+	D+	
3	1540703012	MR. SUPPANUT BOONHAI	2.25	B	F	B	
4	1540701743	MR. SIRAWIT PRASUTSANGJAN	3.05	B+	A	B+	
Group 4							2.393 , 3.000 , 3.250 , 2.125

(a) Randomized Method

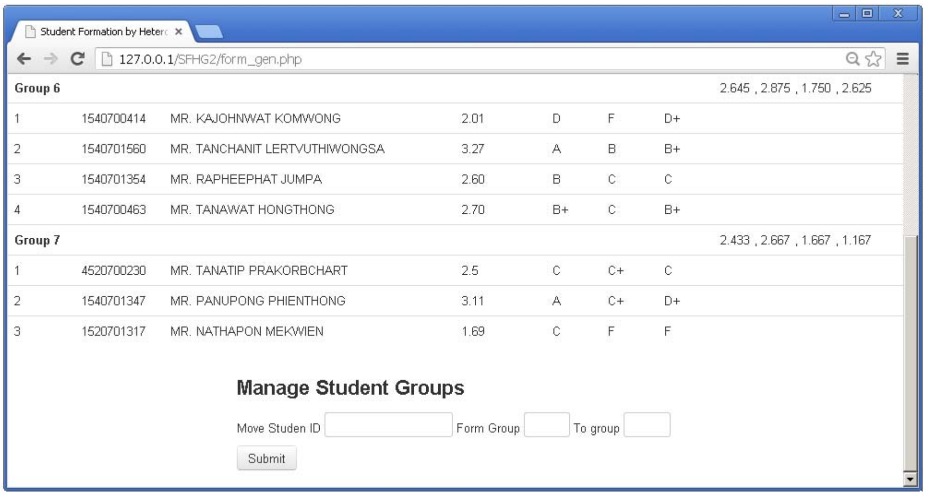


Student list : SFHG Algorithm

No	ID	Name	Cum.Grade	CS310	CS311	CS317	Fitness = 2.33313351525
Group 1							2.198 , 2.500 , 2.250 , 2.250
1	1520701549	MISS WANVISA SILTAM	1.69	C	B+	D+	
2	1540701560	MR. TANCHANIT LERTVUTHIWONGSA	3.27	A	B	B+	
3	1540703061	MR. CHANIN TANG-ON	1.33	C	F	C	
4	4520700230	MR. TANATIP PRAKORBCHART	2.5	C	C+	C	
Group 2							2.738 , 3.625 , 2.125 , 3.250
1	1540702030	MR. JAKKAPONG SUIJARITSRICHAIKUL	2.80	A	B+	C+	
2	1540703012	MR. SUPPANUT BOONHAI	2.25	B	F	B	
3	1540700463	MR. TANAWAT HONGTHONG	2.70	B+	C	B+	
4	1540700471	MR. THANAWUT PLIANMOA	3.20	A	B	A	
Group 3							2.532 , 3.125 , 1.125 , 0.875
1	1550701807	MISS NALITA JITMONGKOLSUK	2.55	B	F	F	
2	1540701347	MR. PANUPONG PHIENTHONG	3.11	A	C+	D+	
3	1520701317	MR. NATHAPON MEKWIEN	1.69	C	F	F	
4	1540700893	MR. KRITCHANUT BORISUT	2.78	B+	C	C	
Group 4							2.435 , 3.125 , 1.750 , 1.875

(b) SFHG Algorithm

Figure 4.6 The example result of student groups made by randomized method and SFHG algorithm



Group 6							2.645 , 2.875 , 1.750 , 2.625
1	1540700414	MR. KAJOHNWAT KOMWONG	2.01	D	F	D+	
2	1540701560	MR. TANCHANIT LERTVUTHIWONGSA	3.27	A	B	B+	
3	1540701354	MR. RAPHEEPHAT JUMPA	2.60	B	C	C	
4	1540700463	MR. TANAWAT HONGTHONG	2.70	B+	C	B+	
Group 7							2.433 , 2.667 , 1.667 , 1.167
1	4520700230	MR. TANATIP PRAKORBCHART	2.5	C	C+	C	
2	1540701347	MR. PANUPONG PHIENTHONG	3.11	A	C+	D+	
3	1520701317	MR. NATHAPON MEKWIEN	1.69	C	F	F	

Figure 4.7 Manage student Groups

As mentioned earlier, criteria settings for grouping students must be input based upon the teacher's needs. The basic requirements are the size of student groups, and the student's attributes. A screenshot of the "Group Setting" menu is presented in Figure 4.8. It consists of two submenus, "Set New Group and Criteria" and "Add Students to Group". When the users want to create a new student formation, the users must start by using the "Set New Group and Criteria" submenu. As presented in Figure 4.9, the group ID and attribute lists required for generating groups must be entered. In order to maximize the students' programming skills, the attributes can be prior programming courses or the scores from the first test of the current class. However, by default, the system sets Cum. Grade as the main criterion to build such heterogeneous groups. Once the attributes for constructing student groups have been completely described, student lists are obligatory to be entered into the system by using the "Add Students to Group" submenu. An example of a screenshot of the "Add Students to Group" is presented in Figure 4.10, where student's criteria are Cum. Grade, CS310, CS311, and CS317.

Figure 4.8 "Group Setting" menu and submenus

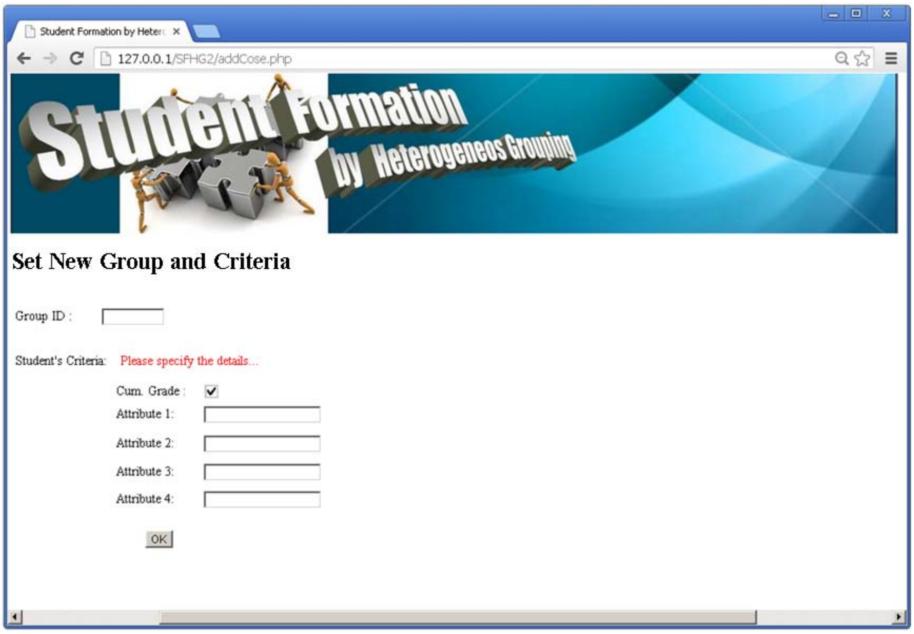


Figure 4.9 Set New Group and Criteria

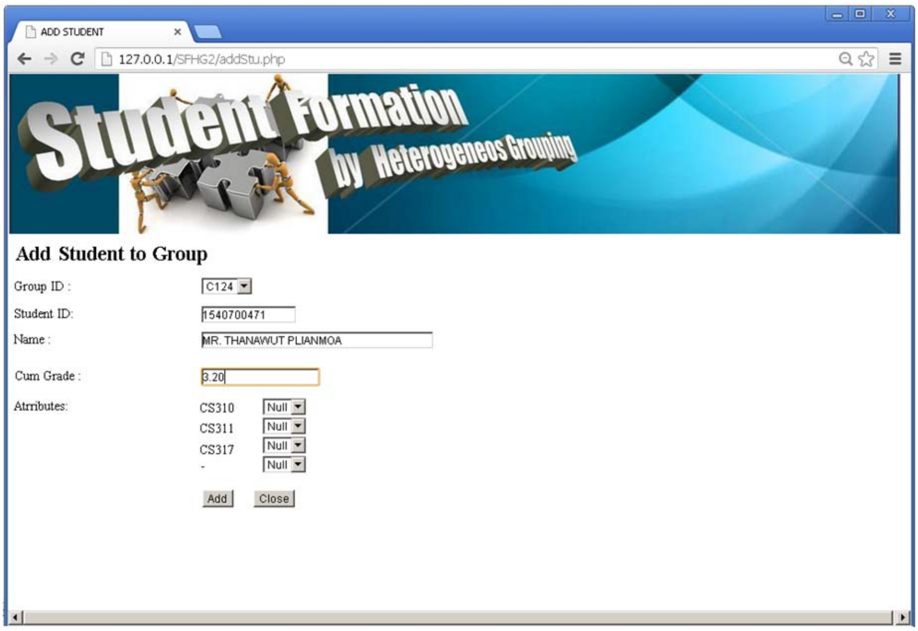


Figure 4.10 An example of adding a student to the group

Chapter 5

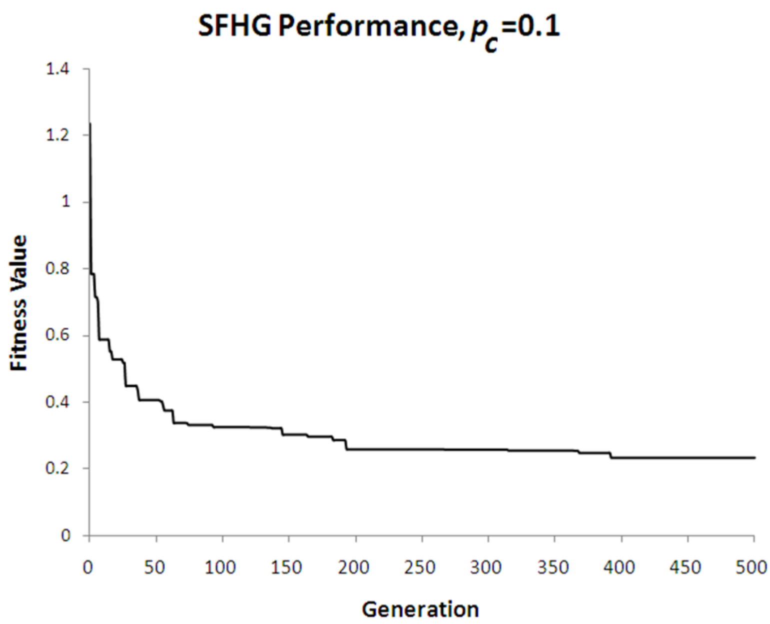
Illustrative Example and Results

Based on the curriculum of Science and Technology, and of those majoring in Information Technology at Bangkok University, IT310 computer programming was the prerequisite class of IT350. In addition, IT401 Management Information Systems was a class that most students registered for before IT350. Therefore, not only Cum. Grade was used as the main attribute for forming student group, but also both IT310 and IT401 as prerequisite classes. We conducted a case study of 70 students who took the class IT350 in the 2nd semester of the academic year 2011 at Bangkok University. Properties of the dataset used for the case study are presented in Table 5.1.

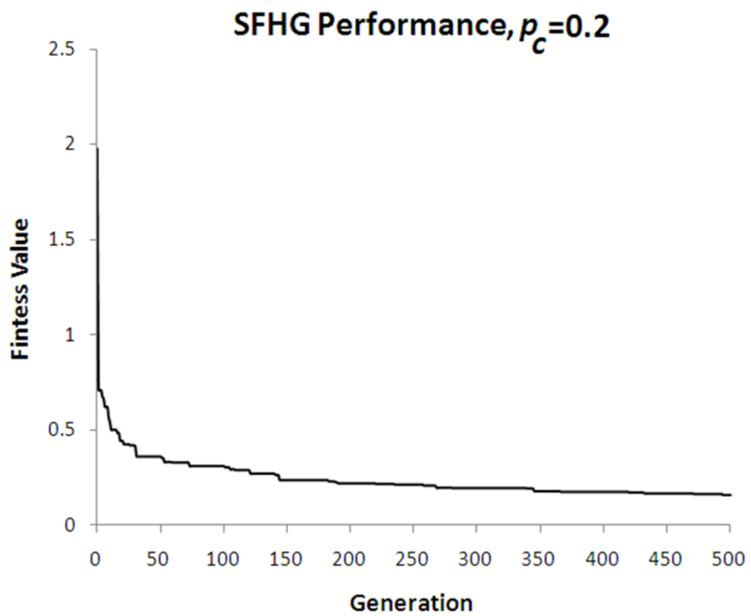
Table 5.1 Properties of the dataset

Constants	**Detail**	**Value**
NumberStudent	Number of students	70
GroupSize	Number of students in each group	5
NumberofAttributes	Number of attributes in forming student groups	3

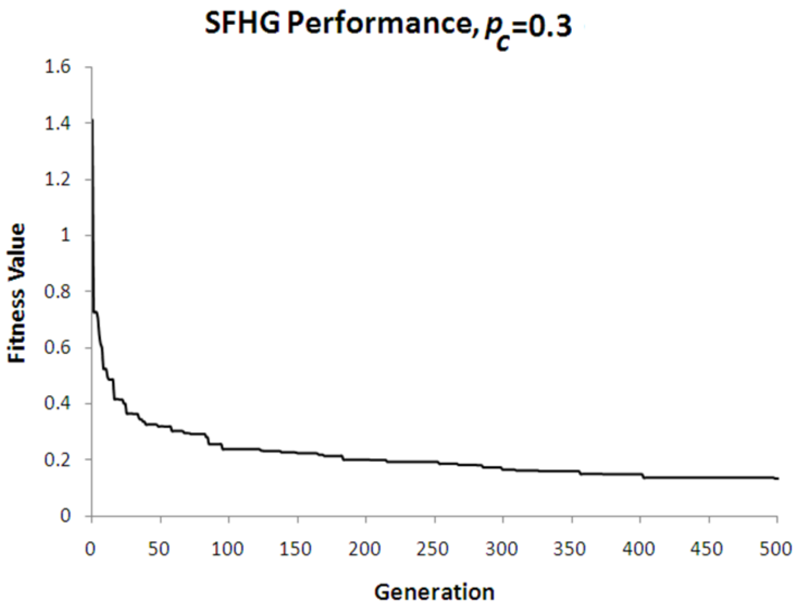
After all information had been well prepared, we ran the program several times to see which initial values of parameters would direct the algorithm's search for the best solution. The results of SFHG with different crossover probability (p_c), ranging from 0.1-0.5, are presented in Figure 5.1.



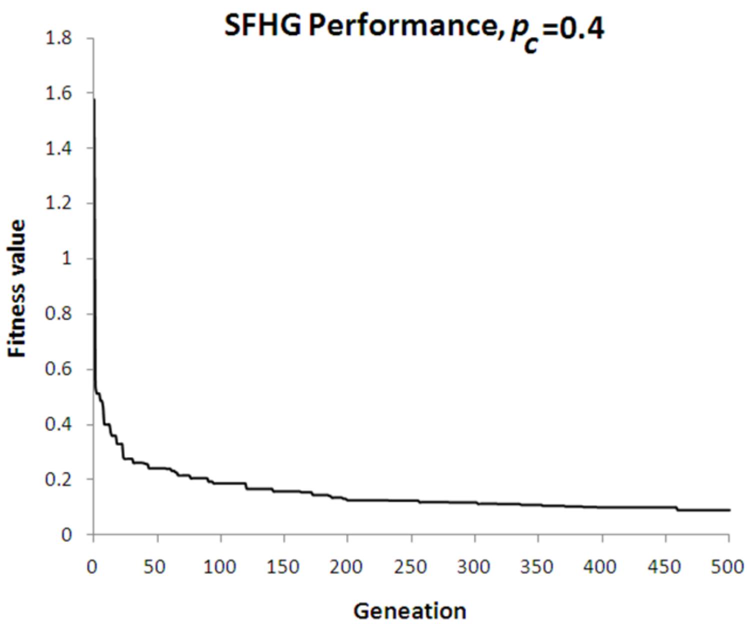
a)



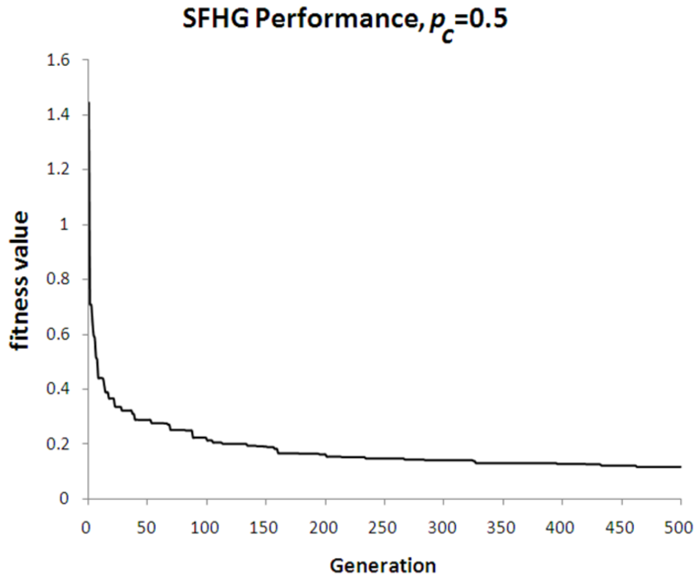
b)


SFHG Performance, p_c=0.3
Fitness Value
Generation
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
0
50
100
150
200
250
300
350
400
450
500

c)


SFHG Performance, p_c=0.4
Fitness value
Geneation
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
0
50
100
150
200
250
300
350
400
450
500

d)



e)

Figure 5.1 Results of SFHG algorithm with different values of p_c where *NumberStudent* = 70, *GroupSize* =5, and *NumberofAttributes* = 3.

Based on the experimentation shown in Figure 5.1, several values of p_c were used to test the algorithm. When p_c was at a level of 0.4, the SFHG algorithm tended to give the best fitness value. Moreover, in most cases, when the generation was approximately 500, the fitness value was fairly good and near the optimal result. Therefore, we decided to use p_c = 0.4 with the number of generations (*Gen*) = 500. The summarized parameter settings for SFHG are illustrated in Table 5.2.

Table 5.2 Parameters setting for SFHG

Constants	Detail	Value
M	Initial population size	300
Gen	Number of generations	500
p_c	Crossover probability	0.4

In this paper, we decided to create groups of five students because the small group gave more opportunities for each student to work (Savova & Donato, 1991), (Brown D. , 1980).This of course depended on the nature and structure of the course. A set of 70 students registered for the IT350 course provided a case study of our experiment, therefore the algorithm generated 14 groups. It should be noted that the algorithm aims to generate the student groups with fairness by balancing dissimilar students into the formed groups equally. Figure 5.2 presents the experiment result created by SFHG, where the fitness value was approximately 0.1702. The graph clearly illustrates that the created groups were well-balanced as the average Cum. GPA in each group spanned a very small gap, which was 2.46 to 2.62. Additionally, the average grades of each group for both IT310 and IT410 were distributed in a small margin of grade as well. The average grade per group of IT310 ranged from 2.40-2.70, while the average grade per group of IT401 was between1.98 to 2.00. These results implied that most groups generated by SFHG were quite similar. Moreover, the fitness function of SFHG was high if the students were mixed in a heterogeneous manner; where top students, moderate students, and low students, were mixed and distributed optimally in the groups, which was also supported by the graph presented in Figure 5.3. This indicated that most groups created by SFHG had a good value of Euclidean distance.

To guarantee the performance of this algorithm we compared the algorithm to the manual grouping method, which was carried out manually by students themselves. If the students were manually grouped together by their own, the students were mixed randomly. As we can see in Figure 5.4, most groups created by the self-selecting method were greatly unbalanced. Some groups were better at some attributes, but some were worse in value than the others. The Cum. GPA per group spanned in a wide range from 0.62 to 3.25. This definitely caused a level of unfairness among the formed groups. Hence, the self-selecting method made by students themselves was proven to be unbalanced in mixing students with different educational skills, because the students were more likely to arrange a group with the same personality and skills. Therefore, the Euclidean distance value of each group made by the students themselves is lower than the SFHG, see previous figure presented in Fig 5.3.

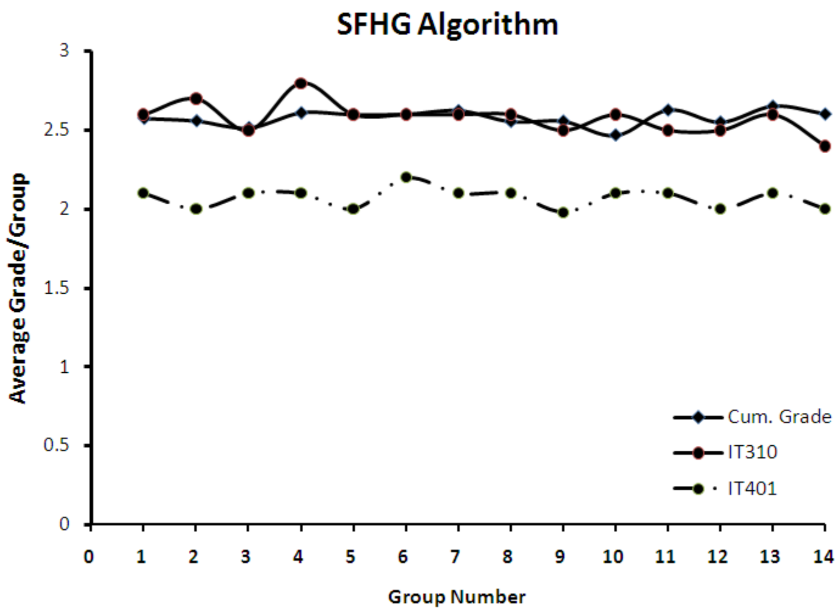


Figure 5.2 Average grades of Cum. Grade and prerequisite courses of formed groups made by SFHG, where the fitness value was approximately 0.1702

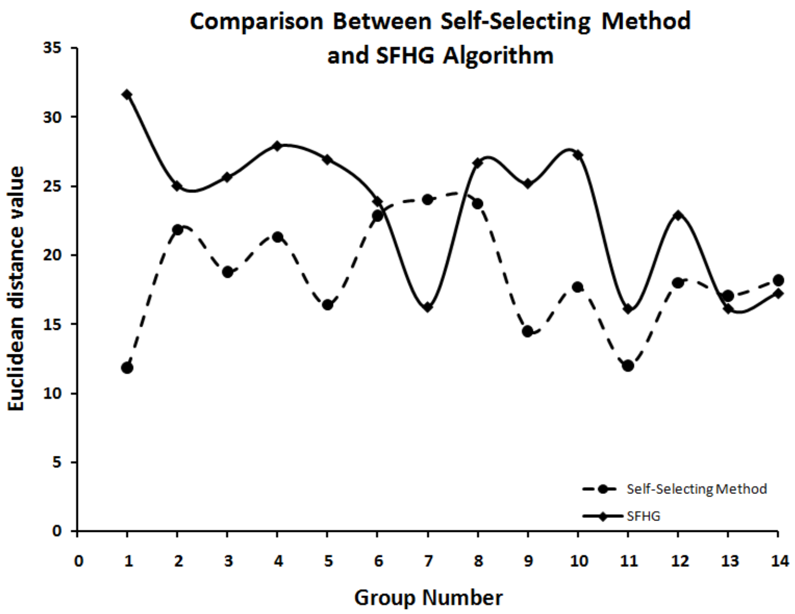


Figure 5.3 Comparison between self-selection method and SFHG algorithm where *NumberStudent* = 70, *GroupSize* =*5* and *NumberofAttributes* = 3.

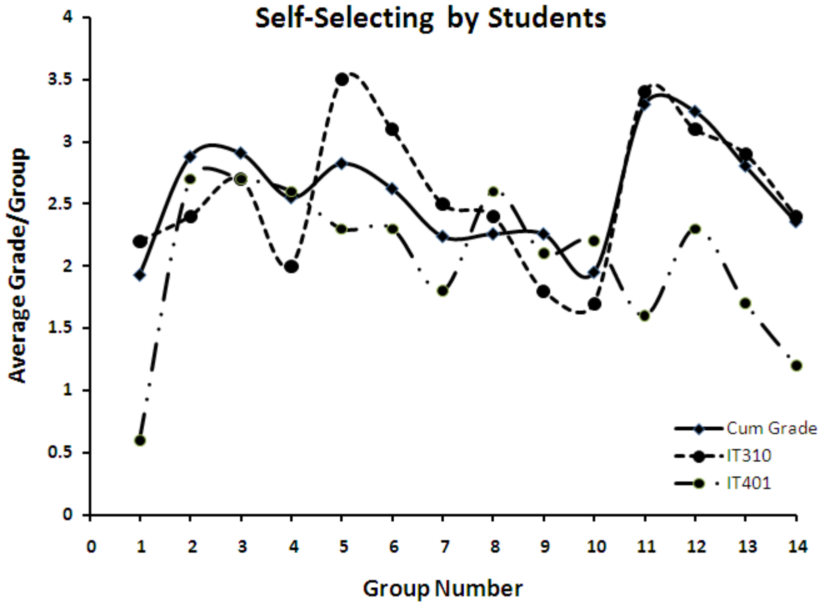


Figure 5.4 Average grades of Cum. Grade and prerequisite courses of formed groups made by self-selecting method

Chapter 6

Conclusions and Future Work

This paper presents the algorithm, termed Student Formation based on Heterogeneous Grouping (SFHG), for constructing heterogeneous grouping based on a genetic algorithm (GA). The case study was performed using 70 students, who majored in Information Technology at Bangkok University, in the 2nd semester of the academic year 2011-2012. The proposed algorithm 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 in searching for the optimal student formation based on a heterogeneous grouping.

The quality of the constructed groups generated by the SFHG is compared to the self-selecting method. According to our empirical experiment, the results of the case study clearly demonstrate that SFHG performed better than the self-selecting method. The results illustrated that the performance of SFHG was effective at balancing and mixing heterogeneous students within the established groups, and creating a fairer and more equitable range of student's educational skills among the formed groups. We hope that by assigning the students to the appropriate group it is possible to improve their abilities by allowing them to learn from others. Hence, the groups may be eligible for improving complex software projects and all students can manage their computer software projects efficiently. In addition, the overall outcome of these groups may be enhanced.

For future research, we shall continue to enhance our algorithm for most datasets to support co-operative learning in the university. The algorithm will be compared to other algorithms and optimized methods as well. Some other attitudes and student abilities will be included in the fitness function to achieve more fairness and equity among students. Moreover, the website application should be able to obtain feedback from teachers in order to enhance its efficiency. The knowledge and programming skills that each student gains from participating in the established groups will be assessed to view the algorithm's efficiency. Furthermore, we plan to upgrade the application to monitor both students' learning progress and the success of projects assigned to each group.

Bibliography

ACT. (2007). The Role of Nonacademic Factors in College Readiness and Success. Iowa City, AI: Author.

Aickelin, U., & Dowsland, K. A. (2000). Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem. *Journal of Scheduling* , 3, pp. 139-153.

Bekele, R. (2005). *Computer-Assisted Learner Group Formation Based on Personality Traits, PhD thesis.* Universität Hamburg, Hamburg.

Boongasame, L., & Sukstrienwong, A. (2009). Buyer Coalitions with Bundles of Items by Using Genetic Algorithm. *Lecture Notes in Computer Science, LNCS 5754* , 674-685. .

Brown, D. (1980). *Principles of the Language Learning and Teaching.* Englewood Cliffs, New Jersey: Prentice Hall Inc.

Brown, E. C., & Sumichrast, R. T. (2001). CF-GGA: a grouping genetic algorithm for the cell formation problem. *International Journal of Production Research* , vol. 39, no. 16, pp. 3651-3669.

Chiu, M. M. (2008). Flowing Toward Correct Contributions During Group Problem Solving: A Statistical Discourse Analysis. *The Journal of the Learning Sciences* , 17:3, 415-463.

Cortés, P., Larrañeta, J., Onieva, L., García, J. M., & Caraballo, M. S. (2001). Genetic algorithm for planning cable telecommunication networks. *Applied Soft Computing* , Vol. 1, Issue 1, June 2001, pp. 21–33.

David, R., & Prasad, K. E. (2009). Genetic Algorithm Approach to Optimize the Process Parameters in Roller Burnishing Operation. *International Journal of Materials Engineering and Technology* , Volume 2, Issue 2, (October 2009), pp. 177-205.

Duan, Y., & Harley, R. (2011). A Novel Method for Multiobjective Design and Optimization of Three Phase Induction Machines. *IEEE Transactions on Industry Applications* , Volume: 47 , Issue: 4 pp. 1707-1715.

Elliot, A., & Patnoe, S. (1997). *The Jigsaw Classroom.* New York: Longman. 150 pp. 2nd Ed.

Felder, R. M., & Brent, R. (2007). Cooperative Learning. In P. A. Mabrouk, *Active Learning: Models from the Analytical Sciences, ACS Symposium Series 970, Chapter 4* (pp. 34-53). Washington, DC: American Chemical Society.

Gandhi, S., Deeba, K., & Solank, V. S. (2012). A Comparative Analysis of Selection Scheme. *International Journal of Soft Computing and Engineering* , Volume-2, Issue-4, September 2012.

Gen, M., Altiparmak, F., & Lin, L. (2006). A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR Spectrum* , 28(3), pp. 337-354.

Gholami, G. (2011). The Effect of Cooperative Learning on Academic Achievement. *Journal of American Sciences* , Vol. 7(12), 596-599.

Gilbert, J. E., & Swanier, C. A. (2008). Learning Styles: How Do They Fluctuate? *Institute for Learning Styles Journal* , Volume 1, Fall 2008, pp. 29-40.

Harrison, G., Griffin, S., & Broughton, L. (2009). The Innovation Bangk: Project Group Formation and Much More. *10th Annual Conference of the Subject Centre for Information and Computer Sciences*, (pp. 44-49).

Holland, J. H. (1992). *Adaptation in Natural and Artijlcial Systems.* MIT Press, Cambridge.

Horwitz, S., & Rodger, S. H. (2009). Using Peer-Led Team Learning to Increase Participation and Success of Under-represented Groups in Introductory Computer Science. *SIGCSE'09.* March 3–7: Chattanooga, Tennessee, USA.

Hwang, G. J., Yin, P. Y., Hwang, C. W., & Tsai, C. C. (2008). An Enhanced Genetic Approach to Composing Cooperative Learning Groups for Multiple Grouping Criteria. *International Forum of Educational Technology & Society (IFETS)* , Vol. 11 (1), pp. 148-167.

Ismail, N. A., & Othman, A. (2006). Comparing university academic performances of HSC students at the three art-based faculties. *International Education Journal* , 7(5), 668-675.

Jacobs, G. (2004). *Cooperative Learning: Theory, Principles, and Techniques*. Retrieved July 19, 2012, from Reading Matrix Inc.: http://www.readingmatrix.com/conference/pp/proceedings/jacobs.pdf

Jamesa, T. L., Barkhib, R., & Johnson c, J. D. (2006). Platform impact on performance of parallel genetic algorithms: Design and implementation considerations. *Engineering Applications of Artificial Intelligence* , Vol. 19, Issue 8, December 2006, pp. 843–856.

Johnson, D. W., & Johnson, R. T. (n.d.). Cooperation and the Use of Technology. *Handbook of research for educational communications and technology* , New York: Macmillan., 1017-1043.

Johnson, D. W., & Johnson, R. T. (1987). *Learning together and alone.* Englewood Cliffs, NJ: . Englewood Cliffs, NJ: Prentice-Hall.

Johnson, D. W., & Johnson, R. T. (1999). Making cooperative learning work. *Theory Into Practice* , Vol 38(2), 67-73.

Johnson, D. W., & Johnson, R. T. (1999). Making cooperative learning work. *Theory Into Practice* , Vol 38(2), 67-73.

Johnson, D. W., Johnson, R. T., & Smith, K. (1998). Cooperative learning returns to college: What evidence is there that it works? Change. 30(4), 26–35.

Johnson, D. W., Johnson, R. T., & Smith, K. (2007). The state of cooperative learning in postsecondary and professional settings. *Educational Psychology Review* , 19: 15-29.

Johnson, D., Johnson, R., & Stanne, M. B. (2000). *Cooperative Learning Methods: A Meta-Analysis.* Minneapolis, MN: University of Minnesota Press.

Johnson, R., Johnson, D., & Holubec, E. (1998). *Cooperation in the Classroom.* Boston: Allyn and Bacon.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Natural Selection.* MA: MIT press, Cambridge.

Kraut, R. E., & Streeter, L. A. (1995, March). Coordination in software development. *Communications of the ACM* , pp. Volume 38 Issue 3, pp. 69 - 81.

Kruse, A. J. (2011, July). *The Benefits of Heterogeneous Groupings over Homogeneous Groupings.* Retrieved July 19, 2012, from Andrew Kruse: http://andrewkruse.efoliomn.com

Kundu, M., & Pal., N. (1999). Self-crossover and its application to the Traveling Salesman Problem. *In Proceedings of IEA/AIE-99* (pp. 326-332). LNAI 1611.

Kuppuswami, R. D. (1012). Genetic Algorithm Based Approach for Imputing Missing Discrete Attribute values in Databases. *WSEAS Transactions on Information Science and Applications* , Issue 6, Volume 9, June 2012, pp. 169-178.

Kutlu, M. O. (1012). Developing a scale on the usage of learner control strategy. *Academic Journals, Educational Research and Reviews* , Vol. 7(10), 12 March, 2012, pp. 244-250.

Levinson, D., Cookson, P. W., & Sadovnik, A. R. (2002). *Education and sociology: an encyclopedia.* Taylor & Francis.

Lotkowski, V. A., Robbins, S. B., & Noeth, R. J. (2004). *The Role of Academic and Non-Academic Factors in Improving College Retention.* ACT Policy Report (ERIC Document Reproduction Service no. ED485476).

Martin, E., & Paredes, P. (2004). 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).

McGowen, ,. R. (2007). *The impact of school facilities on student achievement, attendance, behaviour, completion rate and teacher turnover rate in selected Texas high schools.* Texas A&M University.

Mojdehi, M. A., & Barati, M. (2009). Network operation and reconfiguration to maximize social welfare with distributed generations. *EUROCON 2009, EUROCON '09. IEEE*, (pp. 552 - 557). St.-Petersburg.

Nurul Syakimaj, M., Sapri, M., & Mohd Shahril, A. (2011). Measuring Performance for Classroom Facilities. *International Conference on Sociality and Economics Development* (pp. 209-213). Singapore: IACSIT Press.

Ounnas, A., Davis, H. C., & Millard, D. E. (2009). A Framework for Semantic Group Formation in Education. *Eductional Technology & Society* , 12(4), 43-55.

Pedro, P., Alvaro, O., & Pilar, R. (2010). A Method for Supporting Heterogeneous-Group Formation through Heuristics and Visua. *Journal of Universal Computer Science* , vol. 16, no. 19 (2010), 2882-2901.

Phuong-Mai, N., Terlouw, C., & Pilot, A. (2005). Cooperative learning vs Confucian heritage culture's collectivism: confrontation to reveal some cultural conflicts and mismatch. *Asia Europe Journal* , Springer-Verlag, 403-419.

Renner, G., & Ekárt, A. (2003). Genetic algorithms in computer aided design. *Computer-Aided Design* , Elsevier Science Ltd., Vol. 35, Issue 8, July 2003, pp.709-726.

Sapon-Shevin, M., Ayres, B., & Duncan, J. (2002). 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, Baltimore: Paul H. Brookes.

Savova, L., & Donato, R. (1991). Group Activities in the Language Classroom. *English Teaching Forum* , 24(2) (1991): 12-15.

Sivanandam, S. N., & Deepa, S. N. (2007). *Introduction to Genetic.* Springer Verlag.

Slavin, R., Sharan, S., Kagan, S., Lazarowitz, R. H., Webb, C., & Schmuck, R. (1985). *Learning to Cooperate, Cooperating to Learn.* New York: Plenum Press.

Strahan, D., Hartman, K., & Sikes, S. (n.d.). *Heterogeneous Grouping*. Retrieved July 23, 2012, from North Carolina Middle School Association: http://www.ncmsa.net/heterogr.htm

Ting-Yi Chang, Y.-T. C. (2009). Cooperative learning in e-learning: A peer assessment of student-centered using consistent fuzzy preference. *Expert Systems with Applications* , 36 (2009) 8342–8349.

Wang, D.-Y., Sunny, S. L., & Chuen-Tsai, S. (2003). DIANA: A computer supported heterogeneous grouping system for teachers to conduct successful small learning groups. *Computers in Human Behavior* , 23 (2007) 1997–2010.

Xing-yuan, W., Fan-ping, L., & Shu-guo, W. (2009). Fractal image compression based on spatial correlation and hybrid genetic algorithm. *Journal of Visual Communication and Image Representation genetic algorithm* , 505-510.

Vitae

Name:	Anon Sukstrienwong
Address:	1709 Lasalle Rd, Bangna, Bangkok 10260
Present Position:	Full-time lecturer
Office:	School of Science and Technology, Bangkok University Kluay Namthai, Klong Toey, Bangkok

Educational Background

1989	Phanatpittayakarn school, Chonburi, Thailand
1993	King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
1999	University of Colorado, Colorado, USA.