An Algorithm Based on Self-balancing Binary Search Tree to Generate Balanced, Intra-homogeneous and Inter-homogeneous Learning Groups

Ali Ben Ammar[1], Amir Abdalla Minalla[2]
University College of Tayma, University of Tabuk, Tabuk, Kingdom of Saudi Arabia[1, 2]

***Abstract*—This paper presents an algorithm, based on the self-balancing binary search tree, to form learning groups. It aims to generate learning groups that are intra-homogeneous (student performance similarity within the group), inter-homogeneous (group performance similarity between groups), and of balanced size. The algorithm mainly uses the 2-3 tree and the 2-3-4 tree as two implementations of a self-balancing binary search tree to form student blocks with close GPAs (grade point averages) and balanced sizes. Then, groups are formed from those blocks in a greedy manner. The experiment showed the efficiency of the proposed algorithm, compared to traditional forming methods, in balancing the size of the groups and improving their intra- and inter-homogeneity by up to 26%, regardless of the used version of the self-balancing binary search tree (2-3 or 2-3-4). For small samples of students, the use of the 2-3-4 tree was distinguished for improving intra- and inter-homogeneity compared to the 2-3 tree. As for large samples of students, experiments showed that the 2-3 tree was better than the 2-3-4 tree in improving the inter-homogeneity, while the 2-3-4 tree was distinguished in improving the intra-homogeneity.**

Keywords*—Learning group formation; balanced size groups; homogeneous groups; self-balancing binary search trees; greedy algorithm

I. Introduction

The formation of learning groups is the first step to the success of the educational process, as it allows, depending on the tasks, students to be grouped into homogeneous or heterogeneous groups to be effective and lead to more effective learning. Heterogeneous groups are more effective on tasks that complete lessons and achieve specific learning outcomes, such as projects, assignments, and e-learning. In such a situation, students collaborate, learn from their peers, and share ideas to achieve common goals or accomplish group tasks that no one individual can complete alone. According to [1] and [3], more group heterogeneity has a negative impact on low-ability students, whereas peer effects were not found for high-ability students. Therefore, for such tasks, it is recommended to build groups with low heterogeneity. According to [19], homogeneous grouping is most useful for some types of learning activities, particularly those involving guided discovery, knowledge development, review of material already learned, or highly structured tasks to build competence, allowing students to progress at the same rate. In some cases, homogeneous groups are imposed due to the conditions and requirements of the courses.

Traditional methods that have been used to group students are basically random grouping, student-formed groups where the student chooses his group, and instructor-assigned groups where the teacher assigns the students into groups. There are many characteristics of the students on which the group formation depends, such as their knowledge level, personality traits, communication skills, etc. These characteristics were classified in [13] into static, such as gender, age, and knowledge level, and dynamic, such as interaction level or emotional status. The multiplicity of these characteristics and the multiplicity of students make the grouping process an NP-hard problem, as confirmed in [16], and so difficult to solve manually. Therefore, automated methods were required to form groups in an efficient manner based on several characteristics.

Most automatic grouping approaches studied in this paper have focused on the formation of heterogeneous groups for cooperative purposes and are of small size, with the number of members ranging between 3 and 5 in most cases. This may be due to their focus on e-learning and collaborative tasks. But, in fact, homogeneous groups are still needed in theoretical lectures and training courses, especially in in-person learning. The issue of forming homogeneous groups of large size was not addressed. Similarly, the balance between groups in terms of size and degree of homogeneity has not been addressed much, except in [2], where the authors examined the effect of group sizes on students in a gamification environment. They found that differing sizes between groups affected students' interest, comparison, and discouragement.

In this paper, an algorithm based on the self-balancing binary search tree is proposed to form learning groups. The goal is to build learning groups that are intra-homogeneous (a high level of similarity between the characteristics of the students within the group), inter-homogeneous (similarity or balance between the degree of homogeneity of the groups), and balanced in size. The idea is that we do not set intra-homogeneity as the sole goal of group formation because this will inevitably lead to groups with varying degrees of homogeneity, especially in the case of a high diversity of students before their distribution, which means groups that are highly homogeneous and others that are highly heterogeneous. In addition, if the focus is only on intra-homogeneity, then the issue of unbalanced sizes between groups is raised, as each student will be added to the group closest to him/her without any restrictions on group size. The connection between the three characteristics, namely intra-homogeneity, inter-homogeneity, and balanced size, is necessary for many uses, such as lectures where instructors wish to deliver to more than one group with the same plan and progress at the same rate. The self-balancing binary search trees were used at an initial stage to achieve the objectives of group formation, as they contribute to the formation of student blocks (which are tree branches) with close performances and balanced sizes. Then, the groups are formed from these blocks. The proposed algorithm uses GPA (grade point average) as a key feature for grouping students.

Following this part, the paper is organized as follows: The following section discusses the literature review. The methodology used to construct the suggested algorithm, as well as the experimentation and outcomes of applying the proposed algorithm to some student samples, are then provided. The results and recommendations are explored in the concluding parts, which bring the article to a close.

II. Literature

In the past two decades, the formation of learning groups has been an important educational issue that researchers have addressed for the success of the educational process. This interest has multiplied in the past decade with the development of e-learning platforms, collaborative learning platforms, and collaborative work platforms. This section presents the related works and highlights the three essential aspects: the nature of formed groups (homogeneous vs. heterogeneous), forming methods, and forming characteristics or criteria.

Regarding the nature of groups, heterogeneous grouping is the most widely used grouping type because it can better satisfy diverse learning scenarios, especially in cooperative education, as used by [4], [5], [6], [8], [12], [13], [14], [15], [18], [26], and [28] whereas [19] developed an algorithm to generate homogeneous groups. Some research has focused on intra- and inter-group relationships. In this context, [16], [19], [25], and [27] propose approaches to achieve groups with members that are as similar as possible (inter-homogeneous) but also to enable individual differences among students within such groups (intra-heterogeneous).

There is a consensus that the issue of forming groups cannot be solved manually due to the multiplicity of formation criteria and the multiplicity of students. So, the relationship among these variables and possible grouping alternatives is factorial, making this an NP-hard problem, as confirmed in [16]. In these cases, it becomes necessary to use heuristic search methods to find a satisfactory solution with a considerably lower computational effort. A widely used heuristic method is the genetic algorithm (GA), which is used in [5], [11], [13], [15], [16], [19], [27], and [28]. The study [7] used simulated annealing (SA) to form student groups based on past academic records.

The main characteristics or criteria that were used in the related works to form the groups were knowledge levels, learning styles, communicative skills, leadership skills, gender, age, and self-confidence. Grouping algorithms assign different weights to these characteristics to generate optimized groups. The research [13] classified these characteristics as static and dynamic. Static characteristics are those that do not change or at least do not change during a short period of learning, such as gender, age, previous levels of knowledge, or learning styles. Dynamic characteristics, which cannot be captured at a fixed point, are constantly changing during students' learning processes, such as levels of interaction or emotional status. According to [13], the main disadvantage of traditional non-automatic grouping methods, which include random grouping (used by [5]), student-formed groups (used by [11]), and instructor-assigned groups (used by [14]), is that they are generally based on static characteristics, and even if dynamic characteristics are used, they are not taken into account enough, which may lead to undesirable collaborative results. On the contrary, automatic grouping methods facilitated the use and good management of dynamic characteristics despite the problems associated with those characteristics, particularly how and when to measure them to form groups. [6], [12], [13], [16], [20], [21], and [25], developed automatic grouping methods that achieve collaborative learning outcomes. The phrase "cold start" was used in [13] to denote the problem of the inaccessibility of students' characteristics, such as personality traits, communication skills, and leadership capacities, at the starting point. Dynamic grouping is a solution to the "cold start" problem, in which groups are initiated and then modified by dynamic swapping. But its running time is expensive, and it takes time for groups to form and stabilize and for students to work on a regular basis. Thus, [6] proposes a dynamic grouping method where groups are initially formed based on students learning styles and knowledge levels, and then an activity-based dynamic group formation technique is proposed to swap students based on their knowledge levels. The authors in [17] propose a method to form dynamic groups for students who did not fit into any group and referred to them as “orphan students”. In addition, [10] use dynamic grouping or partial grouping methods to enable students to find the most suitable partners.

The size of groups is less addressed in related works because the interest is in collaborative activities in which the group size ranges from 3 to 5 students. In the study conducted by [2] to investigate the effect of group sizes on students in a gamification environment, they found that varying sizes between groups affected students' interest, comparison, and discouragement but did not affect their perceived effort, perceived choice, perceived competence, tension, or motivation.

In summary, almost all grouping approaches have focused on collaborative tasks that have proliferated rapidly thanks to technological development. But collaborative activities and collaborative learning can be complementary to regular lectures in which the teacher contributes more than the student and in which the number of students is large. For this reason and to facilitate the role of lecturers in achieving the learning outcomes, it is necessary to form learning groups that are homogeneous and balanced in terms of size and homogeneity. We are not aware of any work using a self-balancing binary search tree for forming intra-homogeneous, inter-homogeneous, and size-balanced learning groups.

III. Methodology

This section introduces an algorithm to automate the formation of learning groups. This algorithm aims to improve the homogeneity of students' competence within learning groups for the same course and to achieve a balance between those groups in terms of size and degree of homogeneity. The algorithm relies on two types of self-balancing binary search trees due to its ability to classify and sort data. So, the first stage is to define the homogeneity of the groups and how it should be measured. The second stage introduces self-balancing binary search trees. Next, the steps of the algorithm that generates the learning groups are explained.

A. Group Homogeneity

The proposed algorithm uses GPA (grade point average) as a key feature for grouping students. That is, students with a homogeneous GPA (closet GPA) are more likely to be in the same group. According to [9], [24], and [22], GPA is positively correlated with subsequent academic performance. The need for other grouping criteria, such as students' personality traits and communication skills, is unnecessary because this work does not address collaborative activities. Thus, the homogeneity of the learning group boils down to the homogeneity of the GPAs of its students. The used terminology and the calculated formula of homogeneity are as follows:

- $\mu(g)$: The mean of the students' GPAs within a group g
- $S(g)$: The standard deviation of the students' GPAs within a group g. It measures the mean distance between each student's GPA and a reference point at the center of the range of GPAs, the $\mu(g)$. A small value of S means that the GPAs are distributed close to the central point, μ, and are therefore close to each other, which means that they are homogeneous. Otherwise, they are far from each other and therefore heterogeneous.
- $CV(g)=(S(g)/\mu(g))\times 100$: The coefficient of variation of the students' GPAs within a group g. The coefficient of variation measures GPAs' dispersion as a percentage of their mean to see how strong or weak that dispersion is. Hence, it is used as an indicator of both homogeneity and heterogeneity within a group g. The group g is considered heterogeneous from CV(g)=30% and above because the GPAs of the students in it differ from each other by more than a third of the average. Otherwise, less than 30% (CV(g)<30%), group g is considered homogeneous. Therefore, the CV is used for measuring the homogeneity of student groups created in different ways, such as traditional methods and the proposed algorithm. Thus, the intra-homogeneity of a group g is calculated as follows:

$$H_{intra}(g)=(S(g)/\mu(g))\times 100 \quad (1)$$

Consider a set of n learning groups $G=\{g_1,\ldots,g_n\}$, and their intra-homogeneity set $H=\{(S(g_1)/\mu(g_1)),\ldots,(S(g_n)/\mu(g_n))\}$ then the inter-homogeneity is calculated as follows:

$$H_{inter}(G)=S(H)/\mu(H)\times 100 \quad (2)$$

B. Self-Balancing Binary Search Tree: Definition and use for Forming Learning Groups

A tree is a hierarchical data structure consisting of a set of nodes joined together by edges and having one node called the root. One of the most common tree types is the binary search tree (BST), also called an ordered binary tree. It has the property that the key (data) of each inner node is greater than all keys in its left subtree and less than those in its right subtree. One of its main benefits is speeding up data searches since the time complexity of operations on a BST is directly proportional to the tree's height. Fig. 1 provides an example of a BST that records the following list of GPAs for 17 students, where the GPA is measured on a 5-point scale: {1.88, 1.62, 3.3, 2.52, 4.13, 2.78, 3.75, 2.85, 2.56, 4.18, 1.83, 2.3, 4.05, 1, 3.7, 2.55, 3.29}. As it is shown below in Fig. 1, for a GPA search of 2.78, only the nodes with keys 1.88, 3.3, and 2.52 will be accessed.

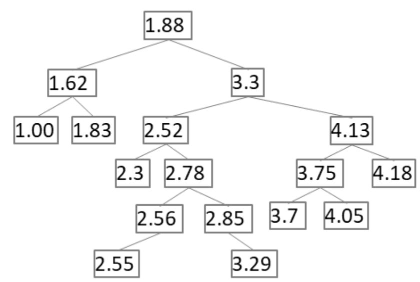

Fig. 1. Example of a BST recording the GPAs of 17 students.

The use of the tree in this work is to build small blocks of students with convergent GPA levels that will be used later to build learning groups with improved homogeneity. Each tree branch (the path from the root to the leaf of the tree) is considered a student block that is represented by their GPAs. The elements of any BST branch are often convergent and homogeneous. For example, in Fig. 1, branches B1 = {1.88, 1.62, 1} and B2 = {1.88, 3.3, 2.52, 2.78, 2.85, and 3.29} are two blocks of students represented by their GPAs. The homogeneities of these two branches, calculated according to formula 1, are 30.14% and 19.18%, respectively, which means that these two students' branches are homogeneous. However, the disadvantage of BST is that its branches are not always the same size. For example, the sizes (number of elements) of B1 and B2 are 3 and 6, respectively, which means they are completely different. This can lead to an imbalance in the size of learning groups being built. That is why we are going to use self-balancing BST instead.

A self-balancing BST is a tree that has the property of rearranging its nodes as necessary to ensure that it does not become too tall and thin. It generalizes the BST, allowing nodes to contain more than two children. There are several implementations for balanced binary search trees like AVL trees, 2-3 tree, 2-3-4 tree, and B-trees. For more information about balanced trees, you can review [23]. For the purpose of this work, 2-3 tree and 2-3-4 tree are applied to generate student blocks that will be used to form learning groups. The 2-3 tree allows each node to have one data element and two children, or two data elements and three children. The 2-3-4 tree allows each node to contain one to three data elements and two, three, or four children. In both trees, if a node contains more than one key, the keys must be in order. Fig. 2(a) and 2(b) redraw the data set of Fig. 1 as the 2-3 tree and the 2-3-4 tree, respectively.

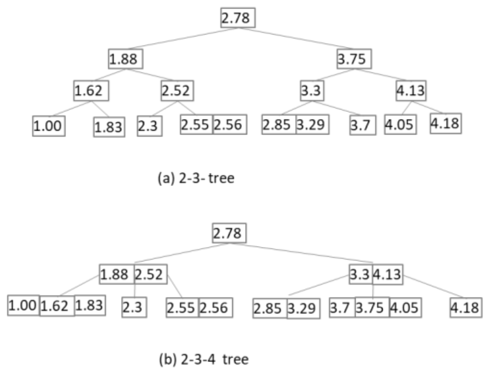

Fig. 2. Examples of self-balancing BST recording the GPAs of 17 students.

As shown in Fig. 2, the two trees are balanced so that all leaves are on the same plane. Using these two trees to form learning groups requires extracting all possible branches by decomposing nodes that are composed of more than one key and then associating each key with its children. The following table shows all possible branches of the 2-3 and 2-3-4 tree shown in Fig. 2.

TABLE I. BRANCHES GENERATED FROM THE 2-3 AND 2-3-4 TREES SHOWN IN FIG. 2.

	2-3 tree	**2-3-4 tree**
Branches	2.78, 1.88, 1.62, 1.0	2.78, 1.88, 1.0
	2.78, 1.88, 1.62, 1.83	2.78, 1.88, 1.62
	2.78, 1.88, 2.52, 2.3	2.78, 1.88, 1.83
	2.78, 1.88, 2.52, 2.55	2.78, 1.88, 2.3
	2.78, 1.88, 2.52, 2.56	2.78, 2.52, 2.3
	2.78, 3.75, 3.3, 2.85	2.78, 2.52, 2.55
	2.78, 3.75, 3.3, 3.29	2.78, 2.52, 2.56
	2.78, 3.75, 3.3, 3.7	2.78, 3.3, 2.85
	2.78, 3.75, 4.13, 4.05	2.78, 3.3, 3.29
	2.78, 3.75, 4.13, 4.18	2.78, 3.3, 3.7
		2.78, 3.3, 3.75
		2.78, 3.3, 4.05
		2.78, 4.13, 3.7
		2.78, 4.13, 3.75
		2.78, 4.13, 4.05
		2.78, 4.13, 4.18

Table I shows that the branches generated from the 2-3 tree are few in number compared to the 2-3-4 tree but are longer than those produced from the 2-3-4 tree. This difference may play an important role in the formation of learning groups and affect their homogeneity, and this will be examined in the experiment section.

The common feature of the two trees, as shown in Table I, is that the produced branches often have homogeneous elements. However, sometimes the generated branches contain GPAs that are different or far from the majority within the branch, such as the branches {2.78, 1.88, 1.62, 1.83} and {2.78, 4.13, 4.18}, where the GPA 2.78 was far from the rest of the GPAs in the two branches. In this case, the homogeneity of the branch will not be highly affected because most of its elements are close together. The homogeneity of branches and their balanced sizes in self-balancing BST serve the aims of this work, which is why these kinds of balanced trees were used.

C. A Greedy Algorithm for Forming Learning Groups

To form learning groups that are intra- and inter-homogeneous and of balanced size, a greedy algorithm was developed using branches of self-balancing BST. This algorithm forms *n* learning groups, where *n* is predetermined. It processes recursively and, at each iteration, selects the appropriate branch *br* to add to the group g_t, where g_t is the group with the least size. If the number of groups of least size is greater than one, the lowest order group is selected. This procedure is applied to balance the size of groups. *br* is the branch that, when added to g_t, gives the best homogeneity to g_t compared to the rest of the candidate branches for addition. At the end of each iteration, the algorithm reconstructs the candidate branches by removing the elements that are common with *br*. The following is the notation used to write the pseudocode for this algorithm:

- *GPAs*: Students' GPAs that will be divided into groups.
- *TT*: The used tree kind is either 2-3 or 2-3-4.
- *T*: The self-balancing BST of kind TT, which will be constructed to contain GPAs
- *S*: The generated branches from the TT tree
- *br*: a branch in S
- *n*: The predetermined number of learning groups
- *G*: The set of learning groups
- g_t: The selected group to add an appropriate branch, where 1⩽t⩽n.
- $H_{intra}(g_j)$: intra-homogeneity of the learning group g_j

The pseudocode of the proposed algorithm, denoted for simplicity as the GF-SBT (Group Formation based on Self-Balancing Tree) algorithm, is presented in Fig. 3 below.

```
GF-SBT Algorithm(GPAs, TT, n)
INPUT:
-  GPAs: list of students' GPAs
-  TT: the type of balanced tree
-  n: number of predetermined learning groups
OUTPUT:
-  G: the set of created learning groups
BEGIN
T← ConstructTree(GPAs, TT) // Construct the T tree of type TT from the
                                                        list of GPAs
S← generateBranches(T)    // Extract all the branches of T
G ← ∅                    // Initialize the learning groups list to be empty
For i ← 1 to n do   //Initialize all groups to an empty set and add them to  G
     g_i ← ∅
     G ← G ∪ {g_i}
End for
While (S not empty) do   // iterate to fill in the groups of G from S
   Find the group g_t ∈ G with the minimum size
   br ← first branch in S
   bestH ← H_intra(g_t ∪ br)   // homogeneity of the GPAs in the set {g_t ∪ br}
   For each:b ∈ S      // iterate to find the appropriate branch br to
    hr ← H_intra(g_t ∪ b)                                    add to g_t
      if (hr < bestH) then
         br ← b
         bestH ← hr
      End if
   End for
   g_t ← g_t ∪ {br}   // add the found appropriate branch br to g_t
   For each:b ∈ S // delete from any branch in S the elements in common
      b ← b − {b ∩ br}                                             with br
   End for
 Refresh S       // Remove from S any branch that has become empty after
End while                          deleting its elements in common with br
Return G        // return the set of created learning groups
```

Fig. 3. GF-SBT Algorithm for forming learning groups.

IV. Experiments

A. Experiment 1

This experiment investigates the ability of the 2-3 and 2-3-4 tree to help form intra- and inter-homogenous learning groups of balanced sizes. Moreover, it was used to measure the algorithm's ability to improve homogeneity and balance the size of learning groups compared to traditional group formation methods.

For this reason, five different tests were conducted to study the results of forming two learning groups in five different ways for 48 students enrolled in a computer programming course at the University of Tabuk. In the first three tests, traditional methods are used for grouping as follows: the first test keeps the same group formation that the university already made, which is self-formation (the student registered himself and chose the group). The other two tests were performed on random formations. The last two tests apply the algorithm GF-SBT using 2-3 and 2-3-4 tree, respectively. The results of these tests are summarized in Table II below. The mean intra-homogeneity is the average intra-homogeneity of the two groups.

Table II shows the learning group sizes generated by each test, the mean intra-homogeneity of the two groups, as well as the inter-homogeneity. For simplicity, in the rest of this paper, intra-homogeneity is used to denote mean intra-homogeneity.

TABLE II. RESULTS OF THE FIVE TESTS FOR FORMING LEARNING GROUPS FOR A SAMPLE OF 48 STUDENTS

	Self-grouping test	**Random test 1**	**Random test 2**	**2-3 Tree test**	**2-3-4 Tree test**
First group size	24	23	29	24	24
Second group size	24	25	19	24	24
Mean intra-homogeneity	20.50%	19.34%	21.16%	16.37%	15.63%
Inter-homogeneity	26.15%	50.88%	61.04%	31.95%	27.75%

The results in Table II showed that there was no correlation between the balance of group size and the improvement in homogeneity. For example, the first, fourth, and fifth tests produced groups of balanced size, with 24 students in each group. However, the intra-homogeneity was different between these three tests (20.50% for the first test, 16.37% for the 2-3 tree test, and 15.63% for the 2-3-4 tree test), which means that the contents of the groups differ from one test to another. In addition, the first test groups were balanced in size (24 per group) in contrast to the second test groups, which were not balanced (23 and 25), but the intra-homogeneity for the second test was better than the first.

The above readings of the results, presented in Table II, confirm that balancing group size is not sufficient to improve intra-homogeneity. Therefore, there is a need to use techniques that combine improving numerical balance with homogeneity when creating groups. In this context and based on the above results, the proposed algorithm, with the use of 2-3 and 2-3-4 trees, succeeded in meeting this need to adjust group sizes and improve group homogeneity compared to traditional methods. Thus, group construction by applying the GF-SBT algorithm is more efficient than traditional formation methods in balancing the number of students between groups and improving intra-homogeneity. The use of 2-3-4 tree was better than 2-3 tree in improving intra-homogeneity (16.37% for 2-3 tree and 15.63% for 2-3-4 tree). Furthermore, the use of the 2-3-4 tree has improved the intra-homogeneity of the self-grouping test by 4.87 percentage points (from 20.50% to 15.63%), the intra-homogeneity of the first random test by 3.71 percentage points (from 19.34% to 15.63%), and the intra-homogeneity of the second random test by 5.53 percentage points (from 21.16% to 15.63%). Therefore, in terms of percentages, the use of the 2-3-4 tree improved the intra-homogeneities resulting from the three traditional tests studied by 23.76% (calculated as 4.87/20.50) for the first test, 19.18% (calculated as 3.71/19.34) for the second test, and 26.13% (calculated as 5.53/21.16) for the third test. The use of the 2-3 trees also improved the intra-homogeneity of the three traditional tests by 20.15% (calculated as (20.5-16.37)/20.5), 15.35%, and 22.61%, respectively.

The inter-homogeneity showed that the GF-SBT algorithm was more efficient at narrowing the gap between the two group homogeneities than the randomized tests. In addition, the use of the 2-3-4 tree approach is more effective than the use of the 2-3 tree in adjusting the homogeneity between the two groups, given that the inter-homogeneity was 27.75% for the 2-3-4 tree and 31.95% for the 2-3 tree.

The bottom line from this experiment is that the GF-SBT algorithm, with its two trees, accurately balances learning group sizes and effectively improves intra-homogeneity caused by traditional grouping methods, with rates ranging from 15% to 26%. Also, compared to the random construction of groups, the GF-SBT algorithm is more efficient at balancing group homogeneity. For this small sample of students, the use of the 2-3-4 tree is more appropriate than the use of the 2-3 trees to create learning groups of equal sizes and optimized and balanced homogeneities.

B. Experiment 2

This experiment aims to investigate the effect of multiplying the number of students and the number of learning groups on the effectiveness of the GF-SBT algorithm in improving the intra- and inter-homogeneity of learning groups and balancing their size. To achieve this aim, five tests were conducted that used the GF-SBT algorithm with the 2-3 and 2-3-4 tree to construct learning groups for a broad sample of students. The number of groups differs in each test and ranges between 2 and 6. The sample consisted of 96 students who self-enrolled in four educational groups in the communication skills course at Tabuk University, as shown in Table III below.

TABLE III. CHARACTERISTICS OF 4 LEARNING GROUPS FROM A SAMPLE OF 96 STUDENTS

	Group size	**Mean intra- homogeneity**	**Inter-homogeneity**
Group 1	24	55.55%	9.05%
Group 2	25		
Group 3	32		
Group 4	15		

Table IV and Table V present the results of the five tests that apply the GF-SBT algorithm, with its two trees, to distribute this sample of students into 2, 3, 4, 5, and 6 learning groups.

TABLE IV. LEARNING GROUP SIZE IN A SAMPLE OF 96 STUDENTS

		Test1: 2 learning groups		Test2: 3 learning groups		Test3: 4 learning groups		Test4: 5 learning groups		Test5: 6 learning groups	
		2-3 Tree	2-3-4 Tree	2-3 Tree	2-3-4 Tree	2-3 Tree	2-3-4 Tree	2-3 Tree	2-3-4 Tree	2-3 Tree	2-3-4 Tree
Group size	Group 1	48	48	32	32	24	24	20	20	16	17
	Group 2	48	48	32	32	24	24	19	19	17	16
	Group 3			32	32	24	24	19	19	16	16
	Group 4					24	24	19	19	16	16
	Group 5							19	19	16	15
	Group 6									15	16

It is shown in Table IV that the GF-SBT algorithm yields groups of very balanced sizes, with some very slight differences resulting either from the non-divisibility of the total number of students evenly, such as in test 4, where 96 is not divisible by 5, or in cases where the number of groups is high, such as in the fifth test. It is also noted that the number of groups in the third test and in the formation proposed by the university are the same, but the groups produced by the third test are more balanced in size than the ones made by the university.

TABLE V. INTRA- AND INTER-HOMOGENEITIES OF LEARNING GROUPS IN A SAMPLE OF 96 STUDENTS

	Test1: 2 groups		Test2: 3 groups		Test3: 4 groups		Test4: 5 groups		Test5: 6 groups	
	2-3 Tree	2-3-4 Tree	2-3 Tree	2-3-4 Tree	2-3 Tree	2-3-4 Tree	2-3 Tree	2-3-4 Tree	2-3 Tree	2-3-4 Tree
Intra-homogeneity	56.04%	55.13%	53.39%	46.95%	54.16%	47.18%	55.38%	53.56%	54.56%	48.75%
Inter-homogeneity	3.25%	2.52%	10.87%	36.86%	13.67%	32.80%	9.22%	19.04%	18.30%	23.26%

Table V shows that the homogeneity of the groups is very high compared to the homogeneity of experiment 1. This is because this sample is characterized by the large number of its students and the great diversity of their GPAs. It shows that the difference between the intra-homogeneity of the two uses of trees in most tests was significant and ranged between 5 and 7 percentage points, except for the tests of two groups and five groups. It displayed that the use of the 2-3-4 tree was the best, compared to the 2-3 tree approach, in improving intra-homogeneity in all five tests for this heterogeneous sample. Compared to the self-grouping results presented in Table IV, the algorithm GF-SBT, with the two uses of trees, improves intra-homogeneity. In this respect, the use of the 2-3-4 tree approach yields an improvement of more than 8 percentage points (from 55.55% to 47.18%), which is equivalent to a 15.07% improvement. Thus, the algorithm GF-SBT with the two uses of trees can improve the intra-homogeneity for homogeneous samples, as in experiment 1, and heterogeneous samples, as in this experiment.

The results presented in the Table V did not show any clear correlation between the number of groups and the intra-homogeneity improvement in the two uses of trees. It is shown that intra-homogeneity does not follow the same pattern as the number of groups. For example, for the use of the 2-3 tree, the mean intra-homogeneity in the five-group test (55.38%) was greater than the mean intra-homogeneity for the three, four, and six group tests (53.39%, 54.16%, and 54.56%, respectively), but smaller than the mean intra-homogeneity in the two-group test (56.04%). The same phenomenon is observed with the 2-3-4 tree approach.

In Table V, by measuring the inter-homogeneity, it is shown that the use of the 2-3 tree was more capable compared to the use of the 2-3-4 tree in constructing homogeneity-balanced groups (well inter-homogeneous) in all tests except the two-group test. Fig. 4 below supports this finding and plots the intra-homogeneity of each test. It reveals that the results of the use of the 2-3 tree appear closer to each other than the ones of the use of the 2-3-4 tree, which appear disparate. However, despite their convergence, the levels of homogeneity resulting from the use of the 2-3 tree were mostly high, compared to the results of the use of 2-3-4 tree. This explains the excellence of the 2-3-4 tree in improving the mean intra-homogeneity. In addition, no correlation was observed between the inter-homogeneity and the number of groups.

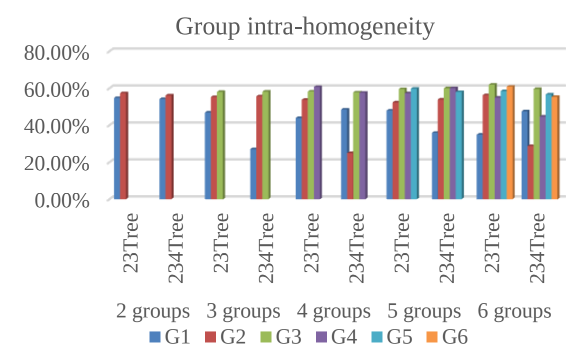

Fig. 4. Groups' intra-homogeneities of the five tests.

All three parameters measured and analyzed here above (group size, intra-homogeneity, and inter-homogeneity) depend on candidate branches (the initial contents of S set in the algorithm GF-SBT) that are used by the GF-SBT algorithm in the group formation process. Table VI below presents the features of the candidate branches of the two trees used by the GF-SBT algorithm for this sample.

TABLE VI. FEATURES OF THE CANDIDATE BRANCHES GENERATED FROM THE 2-3 AND 2-3-4 TREES

	2-3 Tree	2-3-4 Tree
Branche size	6	5
Total Number of candidate branches	71	100
Standard deviation of homogeneities of candidate branches	8.17	9.05

It is evident in Table VI that the candidate branches of the 2-3-4 tree are shorter with dispersed homogeneities, i.e. its standard deviation is bigger than that of the 2-3 tree, and more numerous than those of the 2-3 tree. This is why using the 2-3-4 tree produces better intra-homogeneity than using a 2-3 tree and poorer inter-homogeneity between groups, as shown in Table V.

V. DISCUSSION

Based on the results of the experiments presented in the previous section, the GF-SBT algorithm proved effective in forming learning groups of balanced size and improving their intra-homogeneity. In this regard, it exceeded the traditional methods (self-grouping and random grouping methods) that have been adopted in building learning groups. Similarly, these results agree with what is presented in [16], [19], [25], and [27], which propose to improve both the internal and external relationships of the groups. In spite of the similarities in the obtained results, the methods used for measuring the effectiveness of algorithms were different. For instance, [16], [19], [25], and [27] measure the effect of the grouping algorithm on the students' abilities, while this work measures intra- and inter-homogeneity of groups. The measure of students' ability was allowed for the former because they intended a small group made up of five students for collaborative work, whereas the latter was designed to deal with any size of group that may be greater than twenty students.

The results of using the two self-balancing BSTs, 2-3 and 2-3-4, with a small sample are nearly identical in terms of group size, intra-homogeneity, and inter-homogeneity. Thus, small samples are recommended to be used with any of both trees, with bit priority for 2-3-4 tree.

In the case of large samples, the algorithm produces different results for both trees because of the differences in structure of their candidate branches. The candidate branches of the 2-3-4 tree are more numerous than those of a 2-3 tree and are characterized by various homogeneities and short sizes. They allow the algorithm to create learning groups that are often better intra-homogeneous than the 2-3 tree groups but are less inter-homogeneous. Thus, the use of the 2-3-4 tree is preferable if priority is given to intra-homogeneity more than inter-homogeneity of learning groups. On the contrary, the 2–3 tree produces fewer candidate branches with less various homogeneity and a larger size. That is why the algorithm generates learning groups that are often less intra-homogeneous than 2-3-4 tree groups, but good inter-homogeneously. So, 2-3 tree use is beneficial if the priority of group formation is given to inter-homogeneity. The summary of this paragraph is that the formation of groups through short student blocks improves the intra-homogeneity of learning groups at the expense of their inter-homogeneity, and the opposite occurs through large student blocks.

It has not been proven through experiments that the grouping process adopted by the GF-SBT algorithm is affected by the number of groups to be built. Therefore, the use of the algorithm is effective and recommended regardless of the number of groups to be formed.

The limitation of this work is that, with large samples, neither the use of the 2-3 tree nor the use of the 2-3-4 tree succeeded in integrating improvements in both intra- and inter-homogeneity of the generated learning groups. This limitation will be studied in future work.

VI. CONCLUSION

In this paper, an algorithm based on self-balancing binary search trees has been implemented and experimented with to form intra-homogeneous (student performance similarity within the group) and inter-homogeneous (group performance similarity between groups) learning groups with a balanced size. The self-balancing binary search trees were used at an initial stage to achieve the objectives of group formation, as they contribute to the formation of student blocks (which are tree branches) with close GPAs and balanced sizes. Then, the groups are formed from these blocks. The algorithm uses two versions of self-balancing binary search trees (the 2-3 tree and the 2-3-4 tree), where the difference between them lies in the number and length of branches they produce.

The experiments have shown, with samples from different numbers of students, the efficiency of the proposed algorithm in balancing the size of the groups, balancing the homogeneity between them (inter-homogeneity), and improving their internal homogeneity (intra-homogeneity) compared to the traditional forming methods by up to 26%, whatever the kind of self-balancing binary search tree (2-3 or 2-3-4).

With small samples of students, using the 2-3-4 tree was more effective than the 2-3 tree for improving intra- and inter-homogeneity. However, with large samples, using the 2-3-4 tree was more effective than the 2-3 tree in improving intra-homogeneity but less effective for balancing homogeneity between groups. In this case, the choice between using 2-3 or 2-3-4 trees depends on the instructor's preference, whether intra-homogeneity or inter-homogeneity. The inability of the algorithm to combine intra- and inter-homogeneity optimization for large samples of students using both kinds of self-balancing binary search trees is a limitation that will be worked on in the future.

REFERENCES

[1] Adodo, S. O., & Agbayewa, J. O., "Effect of Homogenous and Heterogeneous Ability Grouping Class Teaching on Student's Interest, Attitude And Achievement in Integrated Science". International Journal of Psychology and Counseling. 3(3), 48-54. 2011.

[2] Ahmad, A., Zeeshan, F., Marriam, R., Samreen, A., & Ahmed, S., "Does one size fit all? Investigating the effect of group size and gamification on learners' behaviors in higher education". J Comput High Educ 33, 296–327. 2021

[3] Booij, A., Leuven, E., & Oosterbeek, H., “Ability peer effects in university: Evidence from a randomized experiment”. Rev. Econ. Stud., 84, 547–578. 2017. DOI:10.1093/restud/rdw045

[4] Chan, T., Chen, C. -M., Wu, Y. -L., Jong, B. -S., Hsia, Y. -T., & Lin, T. -W., “Applying the genetic encoded conceptual graph to grouping learning”. Expert Systems with Applications, 37(6), 4103–4118. 2010. DOI:10.1016/j.eswa.2009.11.014

[5] Chen, C. -M., & Kuo , C. -H., “An optimized group formation scheme to promote collaborative problem-based learning”. Computers & Education, 133, 94–115. 2019. DOI:10.1016/j.compedu.2019.01.011

[6] Haq, I. U., Anwar, A., Rehman, I. U., Asif, W., Sobnath, D., Sherazi, H. H., & et al. , “Dynamic Group Formation With Intelligent Tutor Collaborative Learning: A Novel Approach for Next Generation Collaboration”. IEEE Access, vol. 9 , 143406-143422. 2021. DOI:10.1109/ACCESS.2021.3120557

[7] Hasan, M., “Optimal Group Formulation Using Machine Learning”. arXiv preprint arXiv:2105.07858. 2021. https://doi.org/10.48550/arXiv.2105.07858

[8] Ho, T. F., Shyu, S. J., Wang, F. H., & Li, C. T.-J., “Composing high-heterogeneous and high-interaction groups in collaborative learning with particle swarm optimization”. Proc. World Congr. Comput. Sci. Inf. Eng. (WRI), vol. 4, (pp. 607-611). 2009. DOI: 10.1109/CSIE.2009.876

[9] Hodara, M., & Lewis, K., “How well does high school grade point average predict college performance by student urbanicity and timing of college entry?” US Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance, Regional Educational Laboratory Northwest. https://ies.ed.gov/ncee/ edlabs/projects/project.asp?projectID=4546. 2017.

[10] Jong, B., Wu, Y., & Chan, T., “Dynamic grouping strategies based on a conceptual graph for cooperative learning”. IEEE Transactions on Knowledge and Data Engineering, 18(6), 738–747. 2006.

[11] Krouska, A., & Virvou, M. “An enhanced genetic algorithm for heterogeneous group formation based on multi-characteristics in social networking-based learning”. IEEE Transactions on Learning Technologies, 13(3), 465–476. 2020. DOI: 10.1109/TLT.2019.2927914

[12] Lambić, D., Lazović, B., Djenić, A., & Marić, M., “A novel metaheuristic approach for collaborative learning group formation”. Journal of Computer Assisted Learning, 34(6), 907–916. 2018.://doi.org/10.1111/jcal.12299

[13] Li, X., Ouyang, F., & Chen, W., “Examining the effect of a genetic algorithm-enabled grouping method on collaborative performances, processes, and perceptions”. J Comput High Educ 34, 790–819. 2022. DOI: 10.1007/s12528-022-09321-6

[14] Lin, Y., Chang, Y., & Chu, C. “Novel approach to facilitating tradeoff multi-objective grouping optimization”. IEEE Transactions on Learning Technologies, 9(2) , 107–119. 2016. DOI: 10.1109/TLT.2015.2471995

[15] Masri, H. L., & Kalid, K. S. , “Group-formation system to facilitate heterogeneous grouping in collaborative learning for non-technical courses”. Platform A J. Sci. Technol., vol. 3, no. 1, 48-62. 2020.

[16] [16] Moreno, J., Ovalle, D. A., & Vicari, R. M., “A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics”. Computers & Education, 58(1), 560–569. 2012. DOI:10.1016/j.compedu.2011.09.011

[17] Ounnas, A., Davis, H., & Millard, D., “A framework for semantic group formation”. Proc. 8th IEEE Int. Conf. Adv. Learn. Technol. , (pp. 34-38). 2008.. DOI: 10.1109/ICALT.2008.226

[18] Reis, R. C., Isotani , S., Rodriguez, C. L., Lyra, K. T., Jaques, P. A., & Bittencourt, I. I. , “Affective states in computer-supported collaborative learning: Studying the past to drive the future”. Computers & Education, 120, 29–50. 2018. DOI:10.1016/j.compedu.2018.01.015

[19] Revelo-Sánchez, O., Collazos, C. A., & Redondo., M. A. , “Group formation in collaborative learning contexts based on personality traits: An empirical study in initial Programming courses”. Interaction Design and Architecture(s) Journal - IxD&A, N.49, 2. 2021. DOI:10.1007/978-3-030-66919-5_8

[20] Revelosanchez, O., Collazos, C. A., Redondo, M. A., & Bittencourt, I. I. , “Homogeneous group formation in collaborative learning scenarios: An approach based on personality traits and genetic algorithms”. IEEE Trans. Learn. Technol. 2021. DOI:10.1109/TLT.2021.3105008

[21] Sarode , N., & Bakal , J. , “Toward effectual group formation method for collaborative learning environment”. Sustainable Communication Networks and Application, (pp. 351-361). Chennai, India:Springer. 2021. DOI:10.1007/978-981-15-8677-4_29

[22] Singh, K., & Maloney, T. “Using validated measures of high school academic achievement to predict university success”. New Zealand Economic Papers, 53(1), 89–106. 2019. DOI:10.1080/00779954.2017.1419502

[23] Stephens, R. Essential Algorithms: A Practical Approach to Computer Algorithms. Wiley, ISBN: 9781119575993. 2019, DOI:10.1002/9781119575955

[24] Sulphey, M. M., Al-Kahtani, N. S., & Syed, A. M., “Relationship between admission grades and academic achievement”. The International Journal of Entrepreneurship and Sustainability Issues, 5(3), 648–658. 2018

[25] Sun, Z., & Chiarandini, M. “An exact algorithm for group formation to promote collaborative learning”. Proc. 11th Int. Learn. Anal. Knowl. Conf., (pp. 546-552). 2021. DOI:10.1145/3448139.3448196

[26] Takači, Đ., Marić , M., Stankov , G., & Djenić, A. “Efficiency of using VNS algorithm for forming heterogeneous groups for CSCL learning”. Computers & Education, 109, 98–108. 2017. DOI:10.1016/j.compedu.2017.02.014

[27] Tien, H.-W., Lin, Y.-S., Chang, Y.-C., & Chu, C.-P., “A genetic algorithm-based multiple characteristics grouping strategy for collaborative learning”. Proc. Int. Conf. Web Learn., (pp. 11-22). 2013. DOI:10.1007/978-3-662-46315-4_2

[28] Wang, D. -Y., Lin, S. S., & Sun , C. -T., “DIANA: A computer-supported heterogeneous grouping system for teachers to conduct successful small learning groups”. Computers in Human Behavior, 23(4), 1997–2010. 2007. DOI:10.1016/j.chb.2006.02.008