Date of publication SEP-30, 2024, date of current version JUL-24, 2024.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.23.3.XXXX

An Algorithm Based on An Efficient Cost Model to Form Learning Groups

ALI BEN AMMAR[1], AMIR ABDALLA MINALLA[2]
[1] Higher Institute of Computer Science and Management of Kairouan, University of Kairouan, Tunisia.
[2] University College of Tayma, University of Tabuk, Kingdom of Saudi Arabia.

Corresponding author: Ali Ben Ammar (e-mail: ali.benammar@isigk.rnu.tn).

ABSTRACT The purpose of this research is to form learning groups that are intra-homogeneous (a high level of similarity across student GPAs inside a group), inter-homogeneous (similarity or balance in the degree of homogeneity between groups), and balanced in size. The algorithm proposed for this purpose treats the learning group formation as an assignment-type optimization problem where it seeks to find a feasible least-cost assignment of a given set of students to a given set of learning groups. It is referred to as GAGF (Generalized Assignment Strategy for Group Formation). It is based on an efficient cost model, which performs three tasks: measuring the cost of assigning students to a learning group, relating each improvement in assignment cost to increased intra-group homogeneity and group size balance, and bringing the intra-homogeneity of the groups to a reference value (a specific level of homogeneity), which improves inter-homogeneity. Experimental results have shown that the GAGF algorithm is effective at constructing intra- and inter-homogeneous learning groups with balanced sizes. It was found that using GAGF attained an improvement of more than 29% in intra-group homogeneity when compared to both related work and self-formation methods. It significantly improved inter-group homogeneity, outperforming related works by 79.75%.

KEYWORDS Group formation Algorithm; Learning group formation; Intra-group Homogeneity; Inter-group Homogeneity; Generalized Assignment problem; Cost model; Reference value.

I. INTRODUCTION

THE way learning groups are created affects the success of the educational process and the achievement of learning outcomes. The methods of forming learning groups vary between traditional and automated. Traditional methods include random formation and self-formation, in which students take the responsibility to choose themselves to be the group's members, and the instructor-based formation method involves manually selecting group members based on several criteria. Automatic methods rely on algorithms to accurately determine the members of each group. Each method aims to form homogeneous, heterogeneous, or mixed groups. In a homogeneous group, students have approximately similar characteristics such as academic performance, or GPAs (grade point average), learning styles, personality traits, and demographic information that can include age, gender, and racial, ethnic, or cultural background. On the other hand, in a heterogeneous group, students have different or diverse characteristics. A mixed group includes students with a mixture of homogeneous and heterogeneous characteristics. Homogeneous groups are more effective in in-person learning or in some types of learning activities that involve guided discovery, knowledge development, review of material already learned, or highly structured tasks to build proficiency, allowing students to progress at the same rate [1]. Cooperative learning, which has become widespread due to electronic networks, requires heterogeneous groups to enhance assistance among students, improve interaction between group members, and increase communication and cooperation skills.

Forming groups is a complex process because of the variety of student characteristics it uses and because of the large number of students. According to [2], the automatic formation of learning groups is an NP-hard problem. Therefore, many algorithms have been developed to solve this problem in an effective and fast manner. They used different heuristics and optimization methods, such as genetic algorithms, simulated annealing, ant-colony, and machine learning. Most of the group formation approaches examined in this work dealt with the creation of mixed or heterogeneous groups for collaborative purposes; these groups were typically small, with three to six members. As another option for forming learning groups, some studies have proposed optimizing intra-group homogeneity/heterogeneity and inter-group homogeneity/heterogeneity. However, although in-person learning is still needed and desirable, as stated in several recent

studies [3-5], there have been a few recent works studies [6, 7] that focused on groups' formation via this type of learning. In-person learning is characterized by larger group sizes, with instructor contributing more than the students. Therefore, in in-person learning, it is preferable to have intra-group homogeneity and inter-group homogeneity to achieve learning outcomes, make the instructor's effort balanced between the groups, and facilitate his task.

This paper proposes an algorithm for forming learning groups that are intra-homogeneous (a high level of similarity across student GPAs inside a group), inter-homogeneous (similarity or balance in the degree of homogeneity between groups), and balanced in size. The proposed algorithm considers the learning group formation as an assignment-type optimization problem where the goal is to find a feasible least-cost assignment of a given set of students to a given set of learning groups. It is based on a cost model to measure the assignment cost of a student or a small set of students to a learning group. For every improvement in assignment cost to be reflected in the improvement of intra-group homogeneity and group size balance, the proposed cost model combines several parameters, such as intra-group homogeneity, the size of the item to be assigned, and the size of the group to which the item will be assigned. It also employs a reference value, which is a specific level of homogeneity towards which the intra-homogeneity of the groups tends. The use of a reference value aims to bring the intra-homogeneity of the groups closer to each other and thus improve their inter-homogeneity. Hence, the questions that the research intends to answer in this regard are:

- Is the proposed cost model effective in improving the intra- and inter-homogeneity of learning groups and ensuring a balance between their sizes? If yes, what is the recommended reference value?
- Compared with related works, what is the advantage of the proposed algorithm in improving intra- and inter-homogeneity of learning groups and ensuring balance between their sizes?

II. LITERATURE

The way students are distributed into groups affects learning outcomes [8]. It is a complex process due to the variety of attributes and constraints used to form the groups [9]. Therefore, this issue has received great attention from researchers in the fields of education and computer science, especially in the past two decades. With the development of e-learning and collaborative learning platforms, the issue of automating the formation of learning groups has become very necessary and of great importance. This section presents some algorithms that have been used to automate the formation of learning ensembles and highlights their characteristics.

The objectives of learning group formation algorithms vary, but they often fall into two categories: enhancing homogeneity or improving heterogeneity among students in the group. Other objectives can be constructed by combining these two possibilities. In homogeneous groups, learners' attributes, such as their grades and personality traits, are similar or close to each other, while the opposite occurs in heterogeneous groups. According to [9], homogeneous groups are more suitable than heterogeneous groups to perform skill exercises and guided discovery learning activities. Heterogeneous groups are suitable for in-class problem solving (journal creation, project, and case analysis) and long-term problem-solving projects. [1] developed an algorithm for generating homogeneous groups based on personality traits. [10] proposed an algorithm for forming homogeneous learning groups based on learning styles. However, heterogeneous group formation algorithms have received more attention from researchers, especially with the spread of collaborative learning, learning platforms, and remote work. Some of these algorithms were presented in [11-19]. These algorithms use different grouping attributes but mostly represent academic performance, personality traits, learning styles, and demographic information such as age, gender, and cultural background.

The combination of the previous two types, i.e., homogeneity and heterogeneity, has been applied in some works as a third option to form mixed groups. Mixed groups consist of members with a mixture of homogeneous and heterogeneous learners' attributes. For example, the algorithm proposed in [11] forms learning groups based on a set of heterogeneous criteria such as student grades, knowledge levels, and learning roles and homogeneous criteria such as social interactions. In [9], sensing/intuitive learning styles and topical interests are used as criteria for homogeneity within groups, while active/reflective learning styles and prior knowledge are used as criteria for heterogeneity. The method proposed in [13], forms groups with different levels of knowledge, similar interests, and distributed leadership. As another option for forming learning groups, some works have proposed optimizing intra-group homogeneity/heterogeneity and inter-group homogeneity/heterogeneity. [20-23] propose methods to enhance inter-group homogeneity and intra-group heterogeneity. The purpose of this option is to create different groups that are as similar as possible and, on the other hand, to enhance the complementary role of learners within each learning group by enabling their differences in their qualifying characteristics (students with distinct features and skills). While [6] suggested the formation of intra- and inter-homogeneous learning, that is, the similarity of the students' performance within the group and the similarity of the groups' performance. Another goal adopted in some works is balance in group size. It was stated in [9] that group size has an impact on the learning process. Large groups contribute to increasing the exchange of knowledge and skills, but they also represent a burden in managing and evaluating their members and monitoring their behavior. Therefore, [6] and [20] set group balance as a goal to form learning groups. While works focused at improving cooperative learning, such as [1, 7, 11, 12, 14, 19, 21, 23-26], limit group size, groups should be small, and with no more than 6 members. This last option is suitable for collaborative environments but is not possible in in-person learning environments.

To optimize homogeneity and heterogeneity within and between groups, some related works have formulated the group formation problem as an optimization problem with a single objective or multiple objectives in a combinatorial scenario that integrates all the criteria used. According to [9], a single objective function converts all attribute values to a single value, making no attribute optimal, while multiple objective functions make optimal values for all attributes simultaneously. The works developed by [1, 2, 7, 15, 19], 23, 26-28] used a single objective function to evaluate solutions (groups) formed by the algorithm. They used either a single criterion or multiple criteria. Other works have used multiple objective functions, each consisting of different criteria. [22], proposes a multi-objective optimization of group formation that consists of three objective functions: maximizing mutual homogeneity, maximizing heterogeneity within each group, and maximizing

empathy so that group members have affinity for each other. [21] proposes a multi-objective heuristic to achieve multiple predetermined targets of learning group formation simultaneously, especially the inter-homogeneity and intra-heterogeneity of each learning group. In [9], a multi-objective ant colony system for group formation is developed in which sensing/intuitive learning methods and interests in subjects are used to improve group homogeneity, while active/reflective learning and previous knowledge are used to establish group heterogeneity.

Learner attributes used by related works as criteria in group formation are mainly knowledge level, learning styles, communication skills, leadership skills, gender, age, and self-confidence. [15] 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. Dynamic criteria are especially used in collaborative environments. However, when and how to define its value has been a shortcoming in most related works, because failure to define it properly leads to undesirable collaborative outcomes. Therefore, to solve the problem of the unavailability of student characteristics at the starting point, dynamic grouping was used, in which groups are created and then modified by dynamic swapping [12, 15, 29]. But dynamic grouping causes an expensive runtime. Also, with dynamic grouping, it takes a long time to form and stabilize groups, and it takes a long time for students to work regularly. Because of the multiplicity and dynamism of these criteria and the multiplicity of students to be divided into groups, the issue of forming learning groups becomes more complex, and is therefore considered NP-hard, as stated in [2]. Hence, it is necessary to use optimization techniques to address them, such as genetic algorithms [1, 2, 10, 11, 13, 15, 17, 25, 27, 28, 30], ant-colony [9], and machine learning [24, 31, 32].

In summary, most related works were interested in forming groups for the purpose of collaborative tasks, which spread rapidly thanks to technological development. However, collaborative activities are only a complement to in-person learning, which is still needed and desirable, as reported in several recent studies [3-5]. In in-person learning, the teacher's contribution is greater than that of the student, and the number of students is large. Therefore, in order to achieve learning outcomes and facilitate the teacher’s task, it is necessary to form homogeneous learning groups that are balanced in terms of size and homogeneity. This combination of intra- and inter-homogeneity of groups is treated in [6], where a self-balancing BST was used as a data structure to improve the homogeneity of groups, but achieving both objectives (intra- and inter-homogeneity) at the same level was not possible. In this research, to overcome the drawbacks of the algorithm proposed in [6], the same data structure will be reused, a cost model will be proposed and used to distribute students, and group formation will be formulated as a generalized assignment problem.

III. METHOD

This section introduces the proposed method for automating the formation of learning groups. This method seeks to improve the homogeneity of students' performance within learning groups for the same course and to achieve a balance between those groups in terms of size and degree of homogeneity. Student grouping is based on GPA (grade point average), which means that students with homogeneous GPAs are likely to be in the same group. According to [33-35], GPA is positively related to subsequent academic performance. The contributions of this method are:

- Formulate the learning group formation problem as a generalized assignment problem (GAP). The aim of this formulation is to minimize the cost of assigning students into groups while respecting the carrying capacity of each group. It should be noted here that every reduction in the assignment cost must be matched by an improvement in the intra- and inter- homogeneity of groups and in the balance of their sizes, which is the main goal of this work.
- A cost model that the objective function of GAP will use to minimize the assignment cost and to perform the matching between minimizing the assignment cost and improving both homogeneity and size balance. The proposed cost model combines several parameters, such as the intra-homogeneity of the groups, the size of the item to be assigned, and the size of the group to which the item will be assigned. It is also characterized by the use of a reference value to which the intra-homogeneity of the groups tends. The use of a reference value aims to bring the intra-homogeneity of the groups closer to each other and thus improve their inter-homogeneity.

This method uses self-balancing binary search trees (self-balancing BST) to form learning groups. Self-balancing BST can classify and sort data. They were used in [6] to classify students according to their GPA. The result was branches representing small groups of students who were nearly homogeneous in performance, as shown in Figure 1. For example, the branches B1 = {2.78, 1.88, 1.62, 1.00} and B2 = {2.78, 3.75, 3.3, 3.29} in figure 1.a are two blocks of students whose GPAs are approximately homogeneous. These branches are then used as blocks to form learning groups. Therefore, in this section, some concepts for using self-balancing BST will be summarized. More details on how to use these trees are presented in [6]. Then the focus will be on explaining the two contributions of this work and how to implement them. This section will conclude with the development of an algorithm that summarizes all the steps involved in the formation of learning groups.

A. USING SELF-BALANCING BINARY SEARCH TREES TO FORM LEARNING GROUPS.

Binary search trees (BST) are a data structure used to sort and classify data [36]. Self-balancing BSTs are a class of binary search trees whose branches are balanced, and therefore approximately equal in size or number of elements. For these reasons, it has been used in [6] to form learning groups. The two trees in Figure 1 are two implementations of self-balancing BSTs. The 2-3 tree (Figure 1.a) allows each node to have one or two data elements and two or three children. The 2-3-4 tree (figure 1.b) allows each node to contain one to three data elements and two, three, or four children. If 2-3 and 2-3-4 trees are used for the same sample of data, the 2-3 will produce fewer branches than the 2-3-4 tree, but its branch size will still be larger than the 2-3-4 tree. For additional information about balanced trees and their implementation, see [36]. For example, the two trees in Figure 1 represent the structure of the GPA of 17 students. They were used in [6] in two stages. In the first stage, the tree was built to sort the students and arrange them

according to their GPA. The tree branches were then used as blocks to build the learning groups. They represented small groups of students who were roughly homogeneous in terms of GPA or performance. For example, in figure 1.a, the branches B1 = {2.78, 1.88, 1.62, 1.00} and B2 = {2.78, 3.75, 3.3, 3.29} represent two small sets of students whose GPAs are not very different. But some of its components (mostly the top nodes of the branch) are outliers, and this is useful because it is not recommended that groups be completely homogeneous in order to maintain social integration and exchange of experiences among students [37, 38].

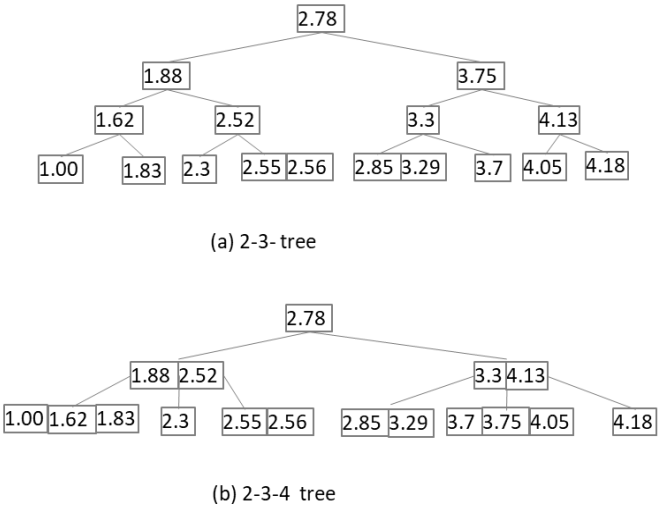

Figure 1. Examples of self-balancing BSTs recording the GPAs of 17 students.

The inclusion of a small group of students (a tree branch) in a learning group is determined by the improvement in group homogeneity (intra-homogeneity). In [6], the following formulas were used to measure intra-homogeneity and inter-homogeneity (homogeneity between groups). Intra-homogeneity (formula 1) and inter-homogeneity (formula 2) are used in the form of percentage values:

$$H_{intra}(\mathrm{g}_i) = \frac{S(\mathrm{g}_i)}{\mu(\mathrm{g}_i)} \times 100, \quad (1)$$

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

where:

- g_i: a learning group.
- $H_{intra}(\mathrm{g}_i)$: intra-homogeneity of g_i
- $\mu(\mathrm{g}_i)$: The mean of the students' GPAs within a group g_i
- $S(\mathrm{g}_i)$: The standard deviation of the students' GPAs within a group g_i.
- $G = \{g_1, \dots, g_n\}$: set of learning groups
- $\mathrm{H} = \{\frac{S(\mathrm{g}_1)}{\mu(\mathrm{g}_1)}, \dots, \frac{S(\mathrm{g}_n)}{\mu(\mathrm{g}_n)}\}$: set of intra-homogeneities of groups.
- $H_{inter}(\mathrm{G})$: inter-homogeneity of learning groups in G

Formulas (1) and (2) measure what is known in statistics as the coefficient of variation (CV), which is the ratio of the dispersion of the data from its mean. In this work, the CV was used as an indicator of both homogeneity and heterogeneity within the group. A group with a CV ≤ 30% is considered homogeneous. Otherwise, it is considered heterogeneous.

The most important advantages of using self-balancing search tree branches to form learning sets are the following:

- Reducing the complexity of the algorithm by reducing the number of iterations, as more than one student is included in a learning group in each iteration rather than one student.
- Helping in the formation of homogeneous learning groups while limiting the achievement of complete homogeneity.

In this research, self-balancing BSTs will be used in the same way and with the same homogeneity measurement formulas as was done in [6].

B. FORMULATE THE PROBLEM OF FORMING LEARNING GROUPS AS A GENERALIZED ASSIGNMENT PROBLEM.

This subsection presents the formulation of the learning group formation problem as a Generalized Assignment Problem (GAP). [39] defines the GAP as follow: "The generalized assignment problem (GAP) seeks the minimum cost assignment of *m* tasks to *n* agents such that each task is assigned to precisely one agent subject to capacity restrictions on the agents.". The aim of formulating the learning group formation problem as a GAP is to minimize the assignment costs of students to learning groups. The proposed cost model, as will be explained later, will reflect this improvement in the assignment cost in achieving the goal of this work, which is to improve intra-group homogeneity, inter-group homogeneity, and group balance.

The formulation of the GAP as presented in [39] is:

$$min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} x_{i,j}$$

$$Subject\ to \sum_{j=1}^{m} a_{i,j} x_{i,j} \leq p_i \quad i = 1, \dots, n$$

$$\sum_{i=1}^{n} x_{i,j} = 1 \quad j = 1, \dots, m$$

$$x_{i,j} \in \{0,1\}\ i = 1, \dots, n;\ j = 1, \dots, m$$

where:

- $c_{i,j}$ is the cost of assigning task j to agent i.
- $a_{i,j}$ is the capacity used when task j is assigned to agent i.
- p_i is the available capacity of agent i.
- $x_{i,j}$ is equals 1 if task j is assigned to agent i, and 0 otherwise.

In this work, the following matching will be applied to formulate the problem of constructing learning groups as a GAP:

- The branches of self-balancing binary search trees are the tasks to be assigned. After each branch b_j is assigned to a group g_i, it is necessary to restructure the branches that contain common elements with b_j. For example, if the assignment algorithm processes branches B1 = {2.78, 1.88, 1.62, 1.00} and B2 = {2.78, 3.75, 3.3, 3.29} (as shown in Figure 1.a here above), and if B1 is assigned during the first stage, the GPA with value 2.78 must be removed from B2 because the student with this GPA has been assigned to a group.
- The available capacity of each group, represented by p_i in the formulation above, corresponds to the maximum

number of students the learning group can contain. Since the goal was to create groups of balanced size, it was necessary to divide the total number of students by the number of groups to be formed to obtain the p_i value. Euclid division is what is applied here, where the result (group size) must be an integer. But if the total number of students is not divisible by the number of groups to be formed, the resulting groups will not accommodate all students. For example, if the total number of students is 65 and the number of groups is 3, the maximum number of each group will be 21 (63/3=21, remainder=2), which means that two students will not be included in any of the three groups. To avoid this problem, the value 1 is added to the division result. Thus, the formula for calculating p_i becomes as follows:

$$p_i = \frac{N}{n} + 1 \qquad (3)$$

Where N is the total number of students and n is the number of groups to be formed. Adding 1 to the quotient, in formula 3, will allow for a small difference between group sizes of no more than n in both cases, whether N is divisible by n or not. However, as long as this size difference does not exceed n, it will not significantly upset the balance between the groups because n is very small compared to the sizes of the groups. For example, if $N = 96$ and $n = 4$ (i.e. N is divisible by n), the maximum group size will be 25 students (96/4 = 24 and 24+1=25). So, the difference between groups will not exceed a maximum of 4 (i.e. n value) students. This means that in extreme cases, the composition of the four groups will be as follows: 25, 25, 25 and 21. If N = 96 and n = 5 (i.e. N is not divisible by n), the maximum group size will be 20 students (96/5 = 19 and 19+1=20). Therefore, the difference between the groups will not exceed a maximum of 5 students (i.e. n value). This means that in extreme cases, the composition of the five groups will be as follows: 20, 20, 20, 20 and 16.

- $a_{i,j}$ represents the capacity occupied by b_j in g_i. In the current problem, it is measured by the number of seats, which means that the value of $a_{i,j}$ will be the number of elements (students) in branch b_j and will be constant for all groups.
- $c_{i,j}$ is a measure of how much the assignment of the branch b_j to the group g_i affects the intra-homogeneity of g_i, the inter-homogeneity between groups, and the balance of their sizes. Its measurement formula will be explained in the next subsection. According to the above problem formulation, the objective would be to minimize the sum of $c_{i,j}$.

To form the learning groups, the proposed algorithm iterates to select, at each iteration, the assignment $x_{i,j}$ that gives the best assignment cost. This selection was based on the idea of heuristic proposed by [40] for GAP. The heuristic proposed by [40] was well suited to the problem of forming learning groups and it was also easy to apply. It states that the assignment of job j to machine i is measured by a weight function $f(i,j)$. For each job j, the difference (called minimum difference) between the second smallest and smallest values of $f(i,j)$ is computed, and the jobs are assigned in decreasing order of this difference. This minimum difference represents the advantage of assigning j to i over the other assignments, i.e., the minimal decrease in cost (or increase in profit) it provides over them. This heuristic assumes that the jobs are independent of each other and that the result of $f(i,j)$ for job j is independent of the prior contents of machine i, which is not the case in the learning group formation problem where the jobs (branches) are intersected (have common elements) and the weight function is calculated based on the prior content of the group. Therefore, the heuristic proposed by [40] will not be applied in all its details, but rather the idea of the minimum difference between the second smallest and smallest values of $f(i,j)$ will be used as a criterion for selecting the best assignment and will be applied in a different way. The proposal is that for each group g_i, the difference (or minimum difference MD_i) between the second smallest and smallest values of $c_{i,j}$ is calculated. MD_i represents the benefit that the best assignment in g_i can make compared to the rest of the assignments. Then, the group g_k with the maximum MD value, for example MD_k, will have priority to include the branch that creates MD_k. Next, the proposed algorithm updates the branches by deleting the common elements included in g_k and iterates again to select the best assignment between the remaining branches and the groups that have not yet reached the available capacity. The selection of the best assignment is formulated as follows:

Let:

- $\varphi_j = \{i : actualSize(g_i) + a_{i,j} \leq p_i\}$ for $j = 1, \dots, m$. This determines for each branch b_j which groups it can be a member of.
- $s_i = \arg min_{j\ /\ i \in \varphi_j}\{c_{i,j}\}$ for $j = 1, \dots, m$. This determines which branch has the minimum cost of assignment in the group g_i.
- $MD_i = \min_{j\ /\ i \in \varphi_j\ and\ j \neq s_i}\{c_{i,j} - c_{i,s_i}\}$ This determines the least minimum difference between the best cost and the other costs in each group g_i.
- $\hat{\imath} = \arg\max_i MD_i$ for $i = 1, \dots, n$. This determines which group has the best assignment.
- $\hat{\jmath} = s_{\hat{\imath}}$.

So, to apply the best assignment:

- $x_{\hat{\imath},\hat{\jmath}} = 1$
- $x_{i,j}$=0 for all $i = 1, \dots, n\ and\ i \neq \hat{\imath}$
- $p_{\hat{\imath}} = p_{\hat{\imath}} - a_{\hat{\imath},\hat{\jmath}}$

C. PROPOSED COST MODEL

This subsection presents the cost model that has been used, in formulating the learning group formation problem as a GAP, to minimize the cost of assigning branches to groups. This cost, denoted by $c_{i,j}$, is a measure of how much the assignment of the branch b_j to the group g_i affects the intra-homogeneity of g_i, the inter-homogeneity between groups, and the balance of their sizes. It is calculated as follow:

$$c_{i,j} = \frac{(h_{i,j} \times |h_{i,j} - \propto GH|)}{(1 + size(g_t)) \times a_{i,j}} \qquad (4)$$

where:

- $g_t = g_i \cup \{b_j\}$. This is a temporary learning group.
- $h_{i,j} = H_{intra}(g_t)$. The intra-homogeneity of g_t which is calculated using formula 1.
- $GH = H_{intra}(G)$: Called the general homogeneity of G. It is the intra-homogeneity of G that is calculated using formula 1, where G is the set of GPAs of all students.
- $\propto \in [0,1]$: a percentage.

As explained above in the problem formulation, the goal is to minimize the sum of cost $c_{i,j}$. To achieve this goal, the value of $h_{i,j}$ in formula (4) must be small, which means that the assignment priority will be to the branches that contribute most

to achieving homogeneity of the group. However, relying on $h_{i,j}$ alone may lead to the formation of highly homogeneous learning groups, which is undesirable and may also cause failure to achieve inter-homogeneity. Therefore, a percentage $\propto$ of GH (The general homogeneity value for all GPAs before distributing them), was determined to be a reference value towards which the homogeneity of the groups would tend. Then $h_{i,j}$ was multiplied by the value of the distance between it and $\propto GH$, so that minimizing this calculation ($h_{i,j} \times |h_{i,j} - \propto GH|$) requires a small value for $h_{i,j}$ and a small value for $|h_{i,j} - \propto GH|$. This means that the branches that contribute the most to improving intra-homogeneity without deviating from the homogeneity of the rest of the groups have priority in assignment. Then, in the first stage, the result of the calculation $h_{i,j} \times |h_{i,j} - \propto GH|$ was divided by the temporary size of the group i.e. $1 + size(g_t)$. The +1 here is added to avoid division by zero when the group is still empty. This division aims to improve inter-homogeneity between groups and achieve balance in their sizes. It forces the cost of assignment, i.e. $c_{i,j}$, to be proportional to the size of the group. This means that if only the numerator in Formula (4) is used as the assignment criterion, the algorithm will speed up the completion of the formation of groups whose size has increased and delay it for groups that are still empty or have few members. This is because adding branches to large groups often results in a significant improvement in the numerator in Formula (4) compared to small groups. What would happen in this case is high intra-homogeneity for the groups that formed quickly in the first iterations of the algorithm because they chose what was best for them, and low intra-homogeneity for the other groups because they had to contain the remaining elements that might be dispersed. It will also happen that the groups formed quickly in the first iterations of the algorithm will have larger sizes than those formed in the last iterations. Then the result of this calculation was divided by the size of the branch, i.e., $a_{i,j}$, so that there is a proportionality between the cost resulting from the branch and its size. Also, this division aims to make the assignment fair, meaning there is no absolute priority in assigning long branches, which may cause a weak balance in homogeneity and sizes between groups.

Thus, the proposed cost model for assignment provides all necessary conditions to ensure homogeneity within and between groups and a balance of their sizes. It also has a mechanism to prioritize groups in branch inclusion.

D. THE ALGORITHM FOR FORMING LEARNING GROUPS.

To form a predetermined number n of learning groups that are intra- and inter-homogeneous and of balanced size, an algorithm is developed, denoted for simplicity as GAGF (Generalized Assignment strategy for Group Formation), and shown in figures 2, 3, and 4. GAGF considers the formation of learning groups as a general assignment problem (described here above) and determines the best assignment for each branch such that the intra- and inter-homogeneity of the groups is optimized. It iterates (from line 10 to line 19 in Figure 2) to assign each branch to the most appropriate group, until all branches are assigned. At each iteration, for each group, the branches whose addition does not overflow the group are selected (line 5 in Figure 3). The cost that each of those branches would achieve if it were added to the group is then calculated using the formula 4 (lines 6 and 7 in Figure 3). The branch with the minimum cost is then kept with the minimum difference (called MD) between its cost and the cost of the second-best branch (Figure 4). At the end of each iteration, the group with the best MD is selected, and the branch that achieved the best cost is assigned to it (from lines 12 to 13 in Figure 2). Also, at the end of each iteration, the algorithm reconstructs the candidate branches by removing elements in common with the selected branch. 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 which 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 T tree
- b: A branch in S
- n: The predetermined number of learning groups
- G: The set of learning groups
- g: A learning group.
- $H_{intra}(g)$: Intra-homogeneity of the learning group g
- maxSize: The allowed size for groups.
- c: The cost of assigning a branch b to group g. It is calculated according to formula (4).
- MD: The minimum difference (MD) in group g is the difference between the best cost resulting from assigning a branch b to g and the cost of the second-best branch.

GAGF- 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
1. T← ConstructTree(GPAs, TT) *// Construct the T tree of type TT // from the list of GPAs*
2. S← generateBranches(T) *// Extract all the branches of T*
3. $G \leftarrow \emptyset$ *// Initialize the Learning groups list to be empty*
4. GH← $H_{intra}(GPAs)$ *// calculate the general homogeneity (GH) // of all GPAs*
5. maxSize← $\frac{size(GPAs)}{n} + 1$ *// calculate the allowed size for groups.*
6. For $i \leftarrow 1$ *to* n do *//Initialize all groups to an empty set and add // them to G*
7. $g_i \leftarrow \emptyset$
8. $G \leftarrow G \cup \{g_i\}$
9. End for

While (S not empty) do *// iterate to fill in the groups of G from S*
11. A ←searchBestLocalAssign(S, G) *// subfunction to search the // best assignment for each group (see figure 3)*
12. $(g_{best}, b_{best}, MD_{best}) \leftarrow \arg_{(g,b,MD)\in A} \max(MD)$ *// Find the //best assignment i.e. the triplet (group, branch, minimum //difference)* that *has the maximum MD in A*
13. $g_{best} \leftarrow g_{best} \cup \{b_{best}\}$ *// add the branch* b_{best} *to the group* g_{best}
14. For each:$b \in S$ *// delete from any branch in S the elements in //common with* b_{best}
15. $b \leftarrow b - \{b \cap b_{best}\}$
16. End for
17. Refresh S *// Remove from S any branch that has become empty. //after deleting its elements in common with* b_{best}
18. End while
19. Return G *// return the set of created learning groups.*

STOP

Figure 2. The Algorithm GAGF (Generalized Assignment strategy for Group Formation).

searchBestLocalAssign(S, G)
INPUT:
- *S: list of branches*
- *G: set of learning groups*

OUTPUT:
- *A: a set containing the best local assignment for each learning group*

BEGIN
1. For each:$g \in G$
2. P ← ∅ // *declare an empty set of branch assignment costs to groups.*
3. For each:$b \in S$
4. $gt \leftarrow g \cup \{b\}$ // *declare gt as a temporary group.*
5. if (size(gt)<=maxSize)
6. h← $H_{intra}(gt)$ // *calculate the homogeneity of the group* gt
7. $c \leftarrow (h \times |h - \propto GH|)/((1 + size(gt)) \times size(b))$ // *calculate the cost of assigning the branch b to g*
8. P ← $P \cup \{(b, c)\}$ // *add the b and its cost c as pair //to the cost list*
9. End if
10. End for
11. (b_m, MD) ← BestForAGroup(P) // *subfunction to find //the branch* $b_m \in P$ *that has the best cost (best local //assignment) for group g and the difference //(MD = minimum difference in P) between the cost of* b_m *//and the cost of the second-best branch. (see figure 4)*
12. A ← $A \cup \{(g, b_m, MD)\}$ // *add to A the best local //assignment* (g, b_m) *with its minimum advantage // (MD) that it provides over other possible assignments.*
13. End for
14. Return A // *return the set of best local assignments.*

STOP

Figure 3. The *searchBestLocalAssign* function to search the best local assignment.

BestForAGroup (P)
INPUT:
- *P: set of branch assignment costs to groups.*

OUTPUT:
- (b_m, MD): *the best branch in P that achieved the best cost plus the difference (MD = minimum difference in P) between the cost of that branch and the cost of the second-best branch.*

BEGIN
1. $(b_m, c_m) \leftarrow \arg\min_{(b,c) \in P}(c)$ // *Find the pair (b, c) that has // the minimum cost in P*
2. MD← $\arg\max_{(b,c) \in P}(c)$ // *Initialize the minimum difference //(MD) between the cost of the branch selected for //assignment and the costs of other possible branches to the //maximum cost in P*
3. For each:$(b, c) \in P - \{(b_m, c_m)\}$
4. diff ← $c - c_m$
5. if (diff< MD)
6. MD ← diff
7. End if
8. End for
9. Return $(b_m$, MD) // *return the set of best local assignment //in the group.*

STOP

Figure 4. The *BestForAGroup* function to selecting the best assignment in a group.

IV. RESULTS

Two experiments were conducted to examine the effectiveness of the proposed method in improving the intra- and inter-homogeneity of groups and achieving balance in their sizes. They were carried out on a sample of 82 students who self-enrolled in four learning groups in the computer skills course at the University of Tabuk. The GPAs of students in this sample were heterogeneous, as the general homogeneity reached 37.14%. During these experiments, the proposed method, which is referred to as the GAGF algorithm for simplicity, was applied to form four learning groups. Its results were then compared to the results of two other formation methods: (i) the self-formation method (the student registers himself and chooses the group) applied at the University of Tabuk; (ii) the related algorithm, presented in [6] and referred to for simplicity as the GF-SBT algorithm. GAGF and GF-SBT use 2-3 and 2-3-4 self-balancing BSTs to generate GPA branches (student blocks). The average intra-homogeneity and inter-homogeneity of the four generated groups are determined for each formation method. The total number of students in each group was calculated as well.

The first experiment tests the effectiveness of the proposed cost model in improving intra- and inter-homogeneity and balancing group sizes. In particular, this experiment focuses on the role of the reference value in improving the homogeneity of groups and the balance of their sizes. Therefore, the GAGF algorithm was applied first without using the reference value $\propto$, and then other times using the reference value $\propto$ that was moved from 30% to 100%. If the reference value is not used in the cost model, it means that the numerator of the proposed cost model (formula 4) consists of $h_{i,j}$ only without multiplying it by the distance between it and a reference point ($|h_{i,j} - \propto GH|$). Small values of $\propto$ mean that the reference value ($\propto GH$) to which the homogeneity of the groups is pulled will be very small compared to the general homogeneity (GH) value. A value of 100% means that the reference value is the same as the general homogeneity (GH) value. The results of using the reference value were then compared with the results of not using it to determine whether the cost model had a role in improving the homogeneity of the groups and the balance of their sizes. In this experiment, a 2-3-4 tree was used. The results of this experiment are presented in Table 1.

The results in Table 1 show that using the proposed cost model with its reference value contributed to an improvement in intra-homogeneity, especially for $\propto$ < 80%, where this improvement peaked in case $\propto$ = 30% when there was a ten-percentage point difference with the case of not using the reference value. However, in cases where the $\propto$ value was less than 60%, this improvement resulted in poor inter-homogeneity. Therefore, applying the cost model with an $\propto$ value ranging between 60% and 80% ($60\% \leq \propto < 80\%$) gave acceptable results for both intra- and inter-homogeneity. The best result was for the case $\propto$=70%, where the intra-homogeneity was 24.37% and the inter-homogeneity was 7.83%, which means an advantage over the results of not using the reference value of 25.34% ($\frac{32.64-24.37}{32.64}$%) for intra-homogeneity and 48.14% ($\frac{15.10-7.83}{15.10}$%) for inter-homogeneity. The sizes of the groups formed by the algorithm were approximately balanced, as one group included 19 students while the rest of the groups included 21 students. This slight difference is due to the maximum group size, which is defined in the algorithm as $\frac{N}{n} + 1$. To summarize this experiment, the proposed cost model was effective in improving both intra- and inter-homogeneities for $\propto$ values between 60% and 80%. It was also able to generate learning groups with near-balanced sizes.

Table 1. Results of applying the proposed cost model with different refence values.

Reference value	Average intra-homogeneity	Inter-homogeneity	Group sizes
Without reference value	32.64%	15.10%	21, 21, 21, 19
α=30%	21.84%	26.44%	21, 21, 21, 19
α=40%	26.70%	34.94%	21, 21, 19, 21
α=50%	29.89%	44.87%	21, 21, 19, 21
α=60%	24.52%	19.83%	21, 19, 21, 21
α=70%	24.37%	7.83%	21, 19, 21, 21
α=80%	32.67%	15.84%	21, 19, 21, 21
α=100%	35.43%	8.70%	21, 19, 21, 21

The second experiment was concerned with comparing the results of applying the proposed method with the results of the related work, which is the GF-BST algorithm, and the results of the self-formation method. The goal is to study the effectiveness of the proposed method, with its two contributions, in improving the intra- and inter-homogeneity of groups and achieving balance in their sizes. For this purpose, the GAGF algorithm was applied with a reference value $\propto$ = 70%. In order to determine the type of self-balancing BST that enhances the effectiveness of the proposed method, the GAGF and GF-BST algorithms were applied twice, first using 2-3 tree and then using 2-3-4 tree.

The results of this experiment are shown in Table 2. It was found that the GAGF algorithm was more effective than the GF-BST algorithm and self-formation in improving intra-homogeneity in both uses of 2-3 tree and 2-3-4 tree. The average intra-homogeneity difference between the GAGF algorithm and the other two methods ranged between 10 and 12 percentage points, giving an improvement rate between 29% and 32%. It was also found that the type of self-balancing BST used did not have a significant impact on the intra-homogeneity of the groups formed. However, the type of self-balancing BST used had a significant impact on improving inter-homogeneity, as using 2-3-4 tree produced better inter-homogeneity than 2-3 tree. In both uses of 2-3 tree and 2-3-4 tree, the GAGF algorithm formed learning groups with more balanced homogeneity than those formed by the GF-BST algorithm. However, the best difference between the two methods was with the use of the 2-3-4 tree, where the inter-homogeneity value of the GAGF algorithm represents 20.25% of the inter-homogeneity value produced by the GF-BST algorithm, i.e. an improvement of 79.75% ($\frac{38.67-7.83}{38.67}$%). The self-formation method was better than the GF-BST and GAGF algorithms in improving inter-homogeneity. In this regard, the difference between its results and the results of the proposed method was not significant, especially when using tree 234. The groups formed using the GAGF algorithm were approximately balanced in size, unlike the groups formed using the self-formation method, whose sizes were unbalanced. The algorithm GF-BST was better in this regard because it was more stringent in balancing the size of groups.

Table 2. Comparison between the results of GAGF, GF-SBT, and self-formation method.

Method	Self-balanced BST	Average intra-homogeneity	Inter-homogeneity	Group sizes
GF-SBT	2-3Tree	34.88%	20.95%	21, 21, 20, 20
	2-3-4 Tree	34.65%	38.67%	21, 21, 20, 20
GAGF	2-3 Tree	24.14%	11.99%	21, 19, 21, 21
	2-3-4 Tree	24.37%	7.83%	21, 21, 19, 21
Self-formation	-	35.82%	5.08%	21, 23, 16, 22

V. DISCUSSION

The results of this study are discussed based on the two research questions, as follows:

Question1: Is the proposed cost model and its reference value effective in improving the intra- and inter-homogeneity of learning groups and ensuring a balance between their sizes? If yes, what is the recommended reference value?

The proposed cost model is used to measure the cost of including a student or a small set of students in a group. It is based on the idea of approximating the homogeneity of the groups around a reference value that represents $\propto$% of the general homogeneity of the students. The experiment's results have shown that the proposed cost model was a significant contribution. Its effect was most evident in achieving an excellent balance between the homogeneity of the groups. To achieve good levels of intra- and inter-group homogeneity, it is recommended to apply the proposed method with an $\propto$ value between 60% and 80% (60% $\leq \propto <$ 80%). With a reference value $\propto$=70%, the proposed method outperforms the related work, presented in [6], in improving the homogeneity between groups by more than 79%. The proposed method outperformed the same work in improving the intra-homogeneity value by about 30%.

Question2: Compared with related works, what is the advantage of the proposed algorithm in improving intra- and inter-homogeneity of learning groups and ensuring balance between their sizes?

Combining the two contributions of this work had a positive impact on improving the intra- and inter-homogeneity of learning groups and ensuring their balance. This made the GAGF algorithm 79.75% better than the GF-BST algorithm, presented in [6], in improving inter-homogeneity and 29.66% better in improving intra-homogeneity. It also enabled it to outperform the self-formation method by 31.96% in improving intra-homogeneity.

Using the 2-3-4 trees was better than using the 2-3 trees, because it provided the optimum balance of enhancing both intra- and inter-homogeneity. This is because the 2-3-4 trees, with their short and homogeneous branches, contributed to the formation of groups with improved intra-homogeneity, which was also confirmed in [6]. The use of the reference value in the cost model contributed to balancing the homogeneity of the groups, and this is an advantage compared to the algorithm GF-BST. Therefore, it is recommended to use a self-balancing BST with short branches to form learning groups. Since the branch length is the height of the tree + 1 and the total number of students N does not exceed a few hundred in most cases, 2-3-4 trees are very suitable for achieving excellent results because they produce short branches with length in the range of $\log_2 N$

(when there is only one member at each node) and $\log_4(N/3)$ (when each node has 4 children).

The proposed algorithm groups students based on one static characteristic, which is the students' GPAs. It would be interesting to include other group characteristics, preferably dynamic data such as interaction or emotional state, which would allow the instructor to adjust the composition of the groups after a few lectures.

I. CONCLUSION

In this paper, an algorithm, called GAGF (Generalized Assignment strategy for Group Formation), has been proposed and tested for forming intra-homogeneous (student performance similarity within the group) and inter-homogeneous (group performance similarity between groups) learning groups with a balanced size. GAGF considers the learning group formation as an assignment-type optimization problem where the goal is to find a feasible least-cost assignment of a given set of students to a given set of learning groups. It is based on a cost model that is used to minimize the assignment cost and perform the matching between minimizing the assignment cost on the one hand and improving intra- and inter-homogeneity and size balance on the other hand. The specificity of this cost model is the use of a reference value towards which the homogeneity of the groups tends and thus improves the inter-group homogeneity.

The results of the experiments have shown the efficiency of GAGF in balancing the size of the groups, balancing the homogeneity between them (inter-homogeneity), and improving their intra-homogeneity. It was found that GAGF was 79.75% better than the GF-BST algorithm, presented in [6], in improving inter-homogeneity and 29.66% better in improving intra-homogeneity. It was also found that GAGF outperforms the self-formation method by 31.96% in improving intra-homogeneity. Therefore, the GAGF algorithm is recommended for in-person learning where the groups are large and the teacher's contribution is greater than the students' contribution, which requires balance between groups to achieve learning outcomes and make the teacher's effort balanced between groups.

The algorithm GAGF provides a mechanism for grouping students according to a static characteristic, which is the students' GPAs. Future work could include other dynamic characteristics that make the formation of learning groups dynamic and responsive to the teacher's desires. For example, it will be important to incorporate dynamic student data such as interaction and adjust student distribution when there is an imbalance between groups.

References

[1] O. Revelo-Sánchez, C. A. Collazos & M. A. Redondo, "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*, no. 49, 2, 2021. https://doi.org/10.55612/s-5002-049-002.

[2] J. Moreno, D. A. Ovalle & R. M. Vicari, "A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics," *Computers & Education*, vol. 58, issue 1, pp. 560–569, 2012. https://doi.org/10.1016/j.compedu.2011.09.011.

[3] V. Gherheș, C. E. Stoian, M. A. Fărcașiu, M. Stanici, "E-Learning vs face-to-face learning: analyzing students' preferences and behaviors," *Sustainability*, vol. 13, no. 8, pp. 4381, 2021. https://doi.org/10.3390/su13084381.

[4] R. J. Petillion, W. S. McNeil, "Student experiences of emergency remote teaching: impacts of instructor practice on student learning, engagement, and well-being," *J Chem Educ.*, vol. 97, pp. 2486–2493, 2020. https://doi.org/10.1021/acs.jchemed.0c00733.

[5] P. Photooulos, C. Tsonos, I. Stavrakas, D. Triantis, "Remote and in-person learning: Utility versus social experience," *SN Comput. Sci.*, vol. 4, no. 116, pp. 1–13, 2023. https://doi.org/10.1007/s42979-022-01539-6.

[6] A. Ben Ammar, A. Minalla, "An algorithm based on self-balancing binary search tree to generate balanced, intra-homogeneous and inter-homogeneous learning groups," *International Journal of Advanced Computer Science and Applications*, vol. 14, issue 6, 2023. https://doi.org/10.14569/IJACSA.2023.0140622.

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

[8] C. T. Krouska, M. Virvou, "Applying genetic algorithms for student grouping in collaborative learning: A synthetic literature review," *Intelligent Decision Technologies*, vol. 13, pp. 395-406, 2020. Doi: 10.3233/IDT-190184. https://doi.org/10.3233/IDT-190184.

[9] F. Fahmi & D. Nurjanah, "Group formation using multi objectives ant colony system for collaborative learning," *Proceedings of the 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Malang, Indonesia, October 2018, pp. 696-702, https://doi.org/10.1109/EECSI.2018.8752690.

[10] V. R. Garcia, B. Vega, A. Ruiz-Ichazu, D. Rivera, E. Rosero-Perez, "Automating the generation of study teams through genetic algorithms based on learning styles in higher education," *Advances in Artificial Intelligence, Software and Systems Engineering*, pp. 270-277, 2021. https://doi.org/10.1007/978-3-030-51328-3_38.

[11] C.-M. Chen, C.-H. Kuo, "An optimized group formation scheme to promote collaborative problem-based learning," *Computers & Education*, vol. 133, pp. 94–115, 2019. https://doi.org/10.1016/j.compedu.2019.01.011.

[12] U. Haq, A. Anwar, I. U. Rehman, W. Asif, D. Sobnath, H. H. Sherazi, et al., "Dynamic group formation with intelligent tutor collaborative learning: A novel approach for next generation collaboration," *IEEE Access*, vol. 9 , 2021, 143406-143422. https://doi.org/10.1109/ACCESS.2021.3120557.

[13] P. K. Imbrie, J. Agarwal, G. Raju, "Genetic algorithm optimization of teams for heterogeneity," *Proceedings of the IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden, October 2020, pp. 1–5. https://doi.org/10.1109/FIE44824.2020.9274243.

[14] D. Lambić, B. Lazović, A. Djenić & M. Marić, "A novel metaheuristic approach for collaborative learning group formation," *Journal of Computer Assisted Learning*, vol. 34, issue 6, pp. 907–916, 2018. https://doi.org/10.1111/jcal.12299.

[15] X. Li, F. Ouyang, W. Chen, "Examining the effect of a genetic algorithm-enabled grouping method on collaborative performances, processes, and perceptions," *J Comput High Educ*, vol. 34, pp. 790–819, 2022. https://doi.org/10.1007/s12528-022-09321-6.

[16] Y. Lin, Y. Chang, C. Chu, "Novel approach to facilitating tradeoff multi-objective grouping optimization," *IEEE Transactions on Learning Technologies*, vol. 9, issue 2, pp. 107–119, 2016. https://doi.org/10.1109/TLT.2015.2471995.

[17] H. L. Masri, K. S. Kalid, "Group-formation system to facilitate heterogeneous grouping in collaborative learning for non-technical courses," *Platform A J. Sci. Technol.*, vol. 3, no. 1, pp. 48-62, 2020. https://doi.org/10.61762/pjstvol3iss1art7130.

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

[19] Đ. Takači, M. Marić, G. Stankov, A. Djenić, "Efficiency of using VNS algorithm for forming heterogeneous groups for CSCL learning," *Computers & Education*, no. 109, pp. 98–108, 2017. https://doi.org/10.1016/j.compedu.2017.02.014.

[20] E. Andrejczuk, F. Bistaffa, C. Blum, J.A. Rodriguez-Aguilar, C. Sierra, "Heterogeneous teams for homogeneous performance," *Proceedings of the Conference on Principles and Practice of Multi- Agent Systems PRIMA 2018, Lecture Notes in Computer Science*, Springer, Cham, Switzerland, 2018, pp. 89–105. https://doi.org/10.1007/978-3-030-03098-8_6.

[21] S. Garshasbi, Y. Mohammadi, S. Graf, S. Garshasbi, J. Shen, "Optimal learning group formation: A multi-objective heuristic search strategy for enhancing inter-group homogeneity and intra-group heterogeneity," *Expert Systems with Applications*, vol. 118, pp. 506–521, 2019. https://doi.org/10.1016/j.eswa.2018.10.034.

[22] P. B. C. Miranda, R. F. Mello, C.A. Nascimento, "A multi-objective optimization approach for the group formation problem," *Expert Systems with Applications*, vol. 162, pp. 113828, 2020. https://doi.org/10.1016/j.eswa.2020.113828.

[23] Z. Sun, M. Chiarandini, "An exact algorithm for group formation to promote collaborative learning," *Proceedings of the 11th Int. Learn. Anal. Knowl. Conf.*, 2021, pp. 546-552. https://doi.org/10.1145/3448139.3448196.

[24] M. Hasan, "Optimal Group Formulation Using Machine Learning," *arXiv preprint arXiv:2105.07858*. 2021.

[25] C. T. Krouska & M. Virvou, "An enhanced genetic algorithm for heterogeneous group formation based on multi-characteristics in social networking-based learning," *IEEE Transactions on Learning Technologies*, vol. 13, issue 3, pp. 465–476, 2020. https://doi.org/10.1109/TLT.2019.2927914.

[26] N. Sarode & J. Bakal, "Toward effectual group formation method for collaborative learning environment," *Sustainable Communication Networks and Application*, Chennai, India:Springer, 2021, pp. 351-361. https://doi.org/10.1007/978-981-15-8677-4_29.

[27] J. M. A. Pinninghoff, A. R. Contreras, L. P. Salcedo et al., "Genetic algorithms as a tool for structuring collaborative groups," *Nat Comput*, vol. 16, pp. 231–239, 2017. https://doi.org/10.1007/s11047-016-9574-1.

[28] Z. Yaqian, L. Chunrong, L. Shiyu, L. Weigang. "An improved genetic approach for composing optimal collaborative learning groups". *Knowledge-Based Systems*, vol. 139, pp. 214-225, 2018, https://doi.org/10.1016/j.knosys.2017.10.022.

[29] B. Jong, Y. Wu, T. Chan. "Dynamic grouping strategies based on a conceptual graph for cooperative learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, issue 6, pp. 738–747, 2006. https://doi.org/10.1109/TKDE.2006.93.

[30] H.-W. Tien, Y.-S. Lin, Y.-C. Chang, & C.-P. Chu, "A genetic algorithm-based multiple characteristics grouping strategy for collaborative learning," *Proc. Int. Conf. Web Learn.*, 2013, pp. 11-22. https://doi.org/10.1007/978-3-662-46315-4_2.

[31] R. Costaguta, "Algorithms and machine learning techniques in collaborative group formation," In: Pichardo Lagunas, O., Herrera Alcántara, O., Arroyo Figueroa, G. (eds) *Advances in Artificial Intelligence and its Applications. MICAI 2015. Lecture Notes in Computer Science*, Springer, Cham, vol. 9414, 2015, https://doi.org/10.1007/978-3-319-27101-9_18.

[32] R. Vankayalapati, K. Ghutugade, R. Vannapuram, and B. Prasanna, "K-means algorithm for clustering of learners performance levels using machine learning techniques," *Revue d'Intelligence Artificielle*, vol. 35, pp. 99-104, 2021. https://doi.org/10.18280/ria.350112.

[33] M. Hodara, K. Lewis, *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, 2017. [Online]. Available at: https://ies.ed.gov/ncee/edlabs/projects/project.asp?projectID=4546.

[34] K. Singh, & T. Maloney, "Using validated measures of high school academic achievement to predict university success," *New Zealand Economic Papers*, vol. 53, issue 1, pp. 89–106. https://doi.org/10.1080/00779954.2017.1419502.

[35] M. M. Sulphey, N. S. Al-Kahtani, A. M. Syed, "Relationship between admission grades and academic achievement," *The International Journal of Entrepreneurship and Sustainability Issues*, vol. 5, issue 3, pp. 648–658, 2018. https://doi.org/10.9770/jesi.2018.5.3(17).

[36] R. Stephens, *Essential Algorithms*, 2nd edition, Wiley, 2019. ISBN: 9781119575993. https://doi.org/10.1002/9781119575955.

[37] S. O. Adodo, J. O. Agbayewa, "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*, vol. 3, issue 3, pp. 48-54, 2011.

[38] A. S. Booij, E. Leuven, H. Oosterbeek, "Ability peer effects in university: Evidence from a randomized experiment," *Rev. Econ. Stud.*, vol. 84, pp. 547–578, 2017. https://doi.org/10.1093/restud/rdw045.

[39] O. E. Kundakcioglu, S. Alizamir, "Generalized assignment problem," In: Floudas, C., Pardalos, P. (eds) *Encyclopedia of Optimization*. Springer, Boston, MA., 2009, pp. 1153-1162. ISBN 978-0-387-74759-0, https://doi.org/10.1007/978-0-387-74759-0_200.

[40] S. Martello, P. Toth, "An algorithm for the generalized assignment problem," *Proceedings of the 9th IFORS Conference*, Hamburg, Germany, 1981.

ALI BEN AMMAR obtained his PhD in computer science from Manouba University in the Republic of Tunisia. He has more than 20 years of academic experience as an assistant professor in Tunisian and Saudi universities. His current research interests include data science and e-learning. He can be contacted via e-mail at: ali.benammar@isigk.rnu.tn.

AMIR A. MINALLA obtained his PhD in English Language Teaching from Sudan University of Science and Technology, Sudan in 2016. He is currently associate professor at Department of Languages and Translation, University of Tabuk, Saudi Arabia. He has several publications in Indexed Magazines. His main areas of interest are applied linguistics, teaching and learning, and problem-based learning. He can be contacted via e-mail at: a-alameen@ut.edu.sa.