Smart Computing Review

A 'Mixed' Approach to Smart Group Formation in Collaborative Learning

Anal Acharya[1], Devadatta Sinha[2], Anurag Sarkar[3], Dibyabiva Seth[4], and Kaustav Basu[5]

[1] Computer Science Department, St. Xavier's College / Kolkata, India / anal_acharya@yahoo.com

[2] Department of Computer Science and Engineering, University of Calcutta / Kolkata, India / devadatta.sinha@gmail.com

[3] Computer Science Department, St. Xavier's College / Kolkata, India / andrewsmith_rusmag@yahoo.co.in

[4] Computer Science Department, St. Xavier's College / Kolkata, India / meetdseth@gmail.com

[5] Computer Science Department, St. Xavier's College / Kolkata, India / kaustavb20@gmail.com

* Corresponding Author: Anal Acharya

Received August 15, 2015; Revised August 29, 2015; Accepted October 16, 2015; Published October 31, 2015

Abstract: The most important aspect in collaborative learning is group formation. Over the years, researchers have proposed a variety of algorithms for collaborative group formation. Researchers have identified learner's characteristics for this purpose like learning style, subject knowledge, preferred time slot, and domain expertise. Most of the generated groups were heterogeneous in nature. This study uses two parameters, the personal style of the learner and a performance indicator for collaborative group formation. Similar personal styles within a group were thought to facilitate group work, whereas a diverse degree of knowledge was thought to enhance the quality of learning. This 'mixed' approach was initially used to form homogeneous groups using K-means clustering. Heterogeneity was introduced within these groups through Agenda-driven search. An Automated Group Decomposition Program was written in Java platform using the developed model ,which was used to generate groups of students doing post-graduate workin Computer Science for their project work. T-test and survey results indicate that the proposed method is more effective for group formation than traditional methods.

Keywords: Agenda Driven Search, Collaborative Learning, Group Formation, K-Means Algorithm, Personal Style, Performance Indicator.

DOI: 10.6029/smartcr.2015.05.010

Introduction

Collaborative learning is a method of learning where two or more people attempt to learn something together [1]. In this form of learning, learners capitalize on each other's resources and skills,thereby broadening their knowledge base. Interaction is considered to be an integral part of collaborative learning. Participants in a group with different levels of knowledge interact with each other and try to overcome their shortcomings. Students who engage in collaborative learning are exposed to others viewpoints, which may significantly vary from their own. This enables them to come up with ways to defend their viewpoint(s) and ensures that they have a deeper understanding of the subject. Several examples of such collaborative learning systems exist in contemporary literature [2][3].

However, an important aspect of collaborative work is group formation. Over the years, several methods for collaborative group formation have been proposed by researchers. These methods have been implemented from the learner's as well as instructor's point of view. The simplest way of doing this is to let the learners choose their own groups [4]. In this case, it was found that learners tended to choose groups based on friendship rather than educational background. These groups generally consisted of learners with a similar collaborative skill and knowledge level. An issue in forming groups in this manner is that it may create "orphan" students. These are students who have not been able to find any groups. This issue can be dealt in two ways: they may be added as a member of an existing group, or the "orphans" may be asked to form a group of themselves [5]. In either case, if the learners fail to assimilate themselves in a group due to a lack of interest and initiative, they may not be able to function properly in the group. Learners and instructors often use the "proximity" method in determining groups [6]. As an example, students with contiguous roll numbers or those residing in the same location may be assigned to form groups. The problem with this method is that these groups may end up having students with different backgrounds and learning interests, and the purpose of collaboration may be defeated. Instructors sometimes use common characteristics of learners to form groups. For example, groups may be formed on the basis of similar research interests or the same free-time slots. However, this would require the instructor to know his learners very well [7].

The aim of this research is to propose an automated method of group formation for collaborative work. However, two aspects must be considered before proceeding with this task. First, specific characteristics on which to group students need to be decided. This study proposes two types of learner attributes for group formation: personal style and performance indicator, the details of which are discussed in Section 3. The second aspect relates to the type of group: homogeneous and heterogeneous. It is argued that a similar personal style facilitates group work. On the other hand, a variation in domain knowledge enhances collaborative work within the group. This is because low-performing students within the group can learn from high-performing students, and also high-performing students can clarify concepts by teaching low-performing students. This study thus proposes collaborative group formation based on the homogeneity of personal style and heterogeneity based on learner performance. In other words, groups are neither totally homogeneous nor heterogeneous but are *mixed* in nature.

In light of the above discussion, this study has two main goals.The first one is to develop a model for automated group formation that ensures the required degree of homogeneity and heterogeneity using Data Mining and Artificial Intelligence algorithms. The second goal is to develop an Automated Group Decomposition Program (AGDP) on the Java platform to implement this model. To ensure the homogeneity of personal style within groups, this study uses a K-means algorithm for initial group decomposition. Agenda-driven search is used to ensure group heterogeneity on the basis of performance indicators. This process is described in detail in the third section. The fourth section gives an example of how this model works. The fifth section describes how the AGDP tool is used in the group decomposition of a set of post-graduate students in computer science for their project work. The sixth section presents a comparative assessmentof the current study along with work done by researchers in this area. The final section concludes this work by discussing the limitations of the model and indicating the areas where future work may be directed. The next section presents a detailed discussion of various methods of group formation proposed by various researchers over the years.

Related Works

Over the years, several researchers have proposed various methods for collaborative group formation. This section discusses some of the pioneering work done by researchers in this field. Redmond [8] did the first work in this area. He proposed group formation based on preferred time slots. This greedy method collects students' preferred time slots using a survey and assigns them to groups based on their preferred subjects. The heterogeneity within the group is achieved on the basis of students' computer proficiency. Christodoulopoulos and Papanikolaou [9] proposed a method that is based on the learning attitude and communication skills of the learner. Learners were first divided into homogeneous clusters using a Fuzzy C Means (FCM) algorithm. It was found that each cluster contained an unequal number of students. These clusters were shuffled to equalize the number of students by using an FCM probability matrix. The heterogeneity of the sample was obtained using random sampling without a replacement. Graf and Bekele [10] proposed a method of student grouping based on a learner's aptitude for group work and performance. Their method initially formed a group of two high-achieving students, two low-achieving students, and two average students in between. The optimality of these groups was determined by the Goodness of Heterogeneity (GH), Coefficient of Variation (CV), and Euclidian Distance(ED) between groups. The optimal values of these parameters were then determined using the Ant Colony Optimization (ACO) algorithm.

In a different vein, Wang et al. [11] used similarity of thinking style as the basis for collaborative group formation. Three different thinking styles were identified: legislative, executive, and judicial. Students' thinking style scores were first tabulated from a questionnaire. Groups were formed on the framework of a triangle generated by Random Mutation Hill Climbing (RMHC), the vertices of which denote three thinking styles. The greater the size of the triangle, the higher the degree of heterogeneity within groups. Ounnas et al. [12] performed a detailed survey of the algorithms proposed by various researchers for collaborative group formation. They found two problems with these methods: (i) groups were created on the basis of a fixed and predefined set of parameters (ii) one-dimensional data derived from questionnaires and surveys was used for model computation. Based on these, they designed a framework to assist teachers in forming groups based on their chosen set of constraints. The framework had two levels: modeling student features using semantic web ontology and negotiating group formation using the constraint satisfaction problem. A technique called "Opportunistic Group Formation" was proposed by Ikeda et al. [13] in which they used the educational significance of collaboration in group learning. A model for negotiation process is constructed which takes charge of collaboration when a learner shifts from individual learning mode to collaborative learning mode. Various system concepts like negotiation process, communication protocols etc. are used to characterize learning groups for Opportunistic Model Formation.

A novel approach to collaborative group formation using Intended Points of Cooperation (IPoC) was introduced by Wessner and Pfister [14] by integrating them into a web-based course. IPoCs define different points of learning where collaboration occurs according to the type and size of the learning group, the collaboration type, and additional material for each activity. This concept was used in development of "Lifelong Learning as Utility," a German project aimed at developing an integrated Internet-based learning infrastructure for continued training and education. De Faria [15] describes a way of group formation for collaborative learning in an"Introductory Computer Programming"course based on students' programming styles. These styles were categorized according to features such as the length of identifiers, size and number of modules, and number of indentations, comments and blank lines. These features were used to develop a tool that will automatically evaluate student programs. Experiments conducted by the authors showed that collaborative learning was very effective for improving the programming style of students working in heterogeneous groups. Tobar and de-Freitas [16] discuss the functioning of a software tool that divides a class into groups based on student characteristics and educational objectives. Groups have been created based on a set of rules that define the characteristics teachers want in group participation. The IMS data model was used to represent student characteristics.

Proposed Approach

As discussed in the introduction section, this study uses two types of parameters for collaborative group formation: personal style and performance indicator. Under personal style, attributes Communication skills, Fluency in using computers and Group work attitude are considered. Subject knowledge is used as a performance indicator to ensure heterogeneity. Each learner is represented by a 5-dimensional vector (si1,si2,si3,si4,si5), 1<=i<=n, where n is the number of learners. The first component represents a student id, whereas the next four components represent the attributes stated above. The model is built on the assumption that similar values within the group sij (1<=i<=n, 2<=j<=4) enhance collaborative work, whereas heterogeneity is required for si5 (1<=i<=n). In other words, groups should be formed in such a manner that it should contain high, medium,and low-achieving learners.The values for the parameters si2, si3, si4 (1<=i<=n) may be assigned by their respective instructors based on their observation. A maximum rating of p is assigned for each parameter. The value of the fifth parameter may be determined by conducting a pre-test on the subject. The full score of the test is q. Since similar values of si2, si3, si4 (1<=i<=n) are required within a group, initial groups are formed by ensuring the homogeneity of the sum of these values.The sum total of si2, si3, si4 (1<=i<=n) for each learner represents Vector A, whereas si5 (1<=i<=n) represents Vector B. The elements of these vectors are represented using Equations (1) and (2), respectively.

$$a_i=\sum_{j=2}^{4} s_{ij}, \quad 1<=i<=n \qquad (1),$$

$$b_i=s_{i5}\ , \quad 1<=i<=n \qquad (2),$$

$$\text{where} \quad A=\{a_i\}, \quad 1<=i<=n \qquad (3),$$

$$B=\{b_i\}, \quad 1<=i<=n \qquad (4) .$$

Learners should be grouped in a manner that allows the members of the group to have "very close" ai values and "very scattered" bi values.In other words,Vector A would be used to ensure homogeneity, whereas Vector B would be used to ensure heterogeneity. It can be achieved by the following three steps which is described below:

(i) Homogeneity within the group is achieved by clustering students using a K-means algorithm on Vector A.

(ii) Each group contains a different number of learners. Reshuffling is performed within groups based on Cluster Distance (CD) to make sure that each group contains an equal number of learners.

(iii) Heterogeneity in knowledge of the subject matter is required within a group to ensure the quality of group work. This is achieved using Agenda Driven Search on groups derived in Step (ii) by applying appropriate heuristics.

The process is explained in detail in the following subsections.

■ Homogeneity using k means clustering

This section forms a cluster of the learners on the basis of the Vector A value, thus ensuring their homogeneity on the basis of personal style. The K-means algorithm is used for this purpose. Two reasons may be cited for using this algorithm in this context: (i) the number of clusters is known in advance (K), and (ii) each group consists of at least one learner [17]. Once an initial set of clusters are formed, the number of learners in these clusters are equalized, as is described in the next section.

The K-means algorithm works by arbitrarily selecting K elements as cluster centers corresponding to each cluster. Elements of each cluster are computed based on similarity. The method used for computing similarity is the Square Error criterion defined by the equation

$$E=\sum_{i=1}^{k}\sum_{a\in C_i} |a-m_i|^2 \quad (5),$$

where E is the sum of the square errors for all objects in a data set, a represents the object, and mi is the mean of cluster Ci[18]. Based on these values, new cluster centers are computed. The procedure applied in this study is enumerated in Algorithm 1.

Algorithm 1: Initial Groups

Input: number of groups g, $\{a_i\}$, 1<=i<=n

Output: n learners clustered into g groups of unequal length

Step 1: Choose arbitrary g elements as cluster centers

Step 2: Repeat until no changes are made to the clusters

Step 2.1 : Assign each $\{a_i\}$, 1<i<=n to the cluster based on Equation (5).

Step 2.2 : Update cluster means

Step 3: Return $\{G_j\}$, 1<=j<=g

■ Group equalization based on cluster distances

The previous section derives a set of student groups using a K-means algorithm based on cluster centers. The problem, however, is that each cluster contains a different number of students. The aim of this section is to construct groups from these clusters so that each group contains an equal number of students while maintaining homogeneity in Vector A within the group. This process is as follows.Let the desired (fixed) number of learners in each group be r. If the number of learners in the group G_i is less than r (say k), then r-k learners are identified in group G_{i+1},and their s_{i1} (1<=i<=n) values are stored in the vector T such that

$$T=\text{Max } CD_{(i+1)} \{G_{i+1}\} \quad (6).$$

The learners identified by Vector T are transferred to group G_i.

On the other hand, if the number of learners in group G_i is greater than r (again say k), then k-r learners are identified in Group G_i and the s_{i1} (1<=i<=n) values are stored in Vector T such that

$$T=\text{Max } CD_i \{G_i\} \quad (7).$$

The learners identified by Vector T are transferred to group G_{i+1}.

This process is summarized in the following algorithm. It is to be noted that this algorithm is greedy [20] in nature, because a deficient number of learners, a_i (1<=i<=n) with maximum cluster distances are chosen to fill the groups at each step without considering future consequences.

Algorithm 2: Group Reshuffle

Input: n learners clustered into g Groups of unequal length
Output: s Groups of fixed length r
Step 1: Sort the groups in ascending order based on cluster centres
Step 2: Repeat the following steps for each group G_i (1<=i<=g)
  Step 2.1: Find the number of learners in each group (k)
  Step 2.2: if (k<r) then
    Step 1.2.1 : Compute t=r-k
    Step 1.2.2 : Compute vector T using formula (6)
    Step 1.2.3 : Move these learners to group G_i.
  Step 2.3: if (k>r) then
    Step 1.3.1: Compute t=k-r
    Step 1.3.2: Compute vector T using formula (7)
    Step 1.3.3: Move these learners to group G_{i+1}.
Step 3: Return $\{G_j\}$, each G_j containing exactly r elements, 1<=j<=s, n=r*s

After this step, a test is performed to verify whether group homogeneity is preserved after equalization. This is done by computing the Group Decomposition Factor (GDF). GDF is the ratio of the number of students that remain in the original group (n1) after equalization divided by the total number of students(n). In other words,

$$GDF=n1/n \qquad (8).$$

This value of GDF is computed after equalization. If the value of GDF is greater than 0.6,the computation proceeds to the next step.

■ Heterogeneity using Agenda driven Search

This section first defines a parameter to check the degree of heterogeneity of Group G_i. This parameter is then used as a heuristic for adjusting the group members using Agenda Driven Search such that each group satisfies a predefined value of the degree of heterogeneity. Let the values of Vector B for Group G_i be divided into a set of intervals as $[I_1,I_2]$, $[I_2+1,I_3],[I_3+1,I_4],\ldots,[I_k,I_k+1]$. If full marks of the test for determining Vector B is q, these intervals may be rewritten as $[0,I_2]$, $[I_2+1,I_3],[I_3+1,I_4],\ldots,[I_k,q]$.Of all of these intervals, suppose x interval contains non-zero members. Then the heterogeneity of Group G_i (denoted by $H(G_i)$) is defined as

$$H(G_i)=x/r \qquad (9).$$

For example, suppose a test was conducted on Group$G_1$of 5 students. The full score of the test is 50. This range is divided into intervals [0-10], [11-20], [21,30], [31-40], and [41-50]. Let's suppose the number of students scoring marks in this range are 1,0,3,0, and 1, respectively. Then, H(G1)=3/5.

This study proposes that the value of $H(G_i)$ corresponding to Group G_i should be greater than 0.5. This is achieved by applying Agenda Driven Search [19] in all groups. Agenda Driven search has an agenda in the form of a list of tasks that a system can perform. Associated with each task is a justification and overall rating for this task. In this case, the agenda is to determine a set of heterogeneous groups of fixed size r. This is justified by the fact that proper learning requires group heterogeneity determined by $H(G_i)$. The algorithm works by computing the sum total of the degree of heterogeneity(SH) of all groups for a current state. This defines the rating of the current state. This state is obtained due to improved $H(G_i)$ of a particular group, achieved by swapping s_{i1} (1<=i<=n) based on Vector B. If this value of SH is greater than the value achieved in the previous state, this state is accepted as the new state. The algorithm works until all the groups achieve the desired degree of heterogeneity. The algorithm is shown below.

Algorithm 3: Achieve Heterogeneity

Input: s Groups of fixed length r satisfying (8) with vector B.
Output: s Groups of fixed length r satisfying (9).

Step 1: Repeat the following steps until $H(G_i)$ >0.6 for all G_i, 1<=i<r.
Step 1.1: Compute $C==\sum_i H(G_i)$. This is current task rating. Call this step CURR.
Step 1.2: for each Group G_j, 1<=j<=r
If $H(G_j)$ <=0.5
Step 1.2.1 : Find Min {k} in [1,r], k<>j such that suitable swap between (s_{j1},s_{k1}) based on B vector yields $H(G_j)$>0.5.
Step 1.2.2 : Compute new $H(G_j)$
Step 1.3: Compute $N==\sum_j H(G_j)$. This is new task rating. Call this step NEW. If N>C then set NEW to CURR.
Step 2: Return {G_j} with desired levels of {$H(G_j)$} 1<=j<r.

The value of GDF corresponding to this group decomposition is computed again. Due to the swaps performed within s_{i1}, 1<=1<=n, this value is bound to be reduced when compared to the value computed in Section 3.3. An acceptable value of GDF computed after this step is thus proposed to be greater than 0.4.

■ Summing it up

This section discusses how the modules proposed in the previous section can be used in collaborative group formation. In a nutshell, a K-means algorithm is used to forman initial group of learners based on the values of Vector A. It is found that each group contains an unequal number of learners. Based on cluster distances, groups are reshuffled for group equalization. The value of GDF corresponding to this group decomposition is computed. Once the GDF is greater than 0.6, the computation proceeds to the next stage. In the third stage, Agenda Driven Search is performed to achieve the required degree of heterogeneity with Vector B values. After computing, if the GDF value is found to be greater than 0.4, the groups are accepted. These steps are enumerated in Algorithm 4.

Algorithm 4: Collaborative Group Formation

Input: n learners with their respective A and B scores
Output: s groups each containing r elements ensuring group wise homogeneity in vector A and heterogeneity in vector B

Step 1: Apply *Initial Groups*
Step 2: Apply *Group Reshuffle*
Step 3: Compute GDF. If GDF<0.6
Step 3.1 Print "Group decomposition not possible"
Step 3.2 Exit
Step 4: Apply *Achieve Heterogeneity*.If GDF<0.4
Step 4.1 Print "Group decomposition not possible"
Step 3.2 Exit
Step 5: Return {G_j}, 1<=j<=s

Example

The aim of this section is to illustrate the working of the algorithms proposed in the previous section by using a suitable example. A data set corresponding to 15 students is used for this purpose. Each group consists of 3 learners. The algorithm works in 3 steps. The first step uses a K-means algorithm to compute Vectors A and B corresponding to the student set, their cluster numbers, and the Euclidian distance of Vector A from the center of the cluster. The corresponding results are shown in Table 1. These clusters are taken as the basis for initial group formation. The Statistical Package for Social Sciences (SPSS) was used for ease of computation. It is to be noted that marks corresponding to each of the three parameters related to personal style was allocated out of 10, whereas marks corresponding to subject knowledge were allocated out of 30.

Table 1. Initial clustering of students into groups using K-means algorithm.

Student id	*Vector A*	*Vector B*	*Cluster Number*	*Distance*
1	17	12	1	0.1428
2	18	06	1	1.1428
3	12	28	3	0.6667
4	23	09	2	1.3333
5	13	07	3	0.3333
6	17	13	1	0.1427
7	21	6	2	0.6667
8	25	27	5	0.0000
9	16	28	1	0.8572
10	21	28	2	0.6667
11	18	25	1	1.1428
12	28	05	4	0.0000
13	17	14	1	0.1428
14	13	16	3	0.3333
15	21	11	2	0.6667

The clusters are sorted according to the ascending order of cluster centers, which is found to be {3,1,2,5,4}}. However, each of these clusters are found to contain a variable number of learners, which is {3,6,4,1,1}. The resulting cluster is shown in Table 2.

Table 2. Clusters obtained after sorting on cluster centers

Cluster Number	*Student id*	*Cluster Centre*	*Cluster Size*
3	3,5,14	12	3
1	1,2,6,9,11,13	17	6
2	4,7,10,15	22	4
5	8	25	1
4	12	28	1

Cluster equalization is performed next. For example, Group 1 contains 6 learners instead of 3. For the purpose of equalization, the 3 learners farthest away from Group 1 are selected and transferred to Group 2. This is shown in Table 3.

Table 3. Equalizing group 1

Group Number	3	1	2	5	4
Student id	3,5,14	1,6,13	2,9,11,4,7,10,15	8	12

This process is continued for all the groups until they are all finalized. The final groups obtained after equalization are shown in Table 4.

Table 4. Final groups obtained after equalization

Group Number	3	1	2	5	4
Student id	3,5,14	1,6,13	7,10,15	8,9,11	2,4,12

The value of the GDF factor is computed to check the preserved degree of homogeneity. The numbers of learners that remain in their original groups after decomposition are

3(Group 3) + 3(Group 1) + 3(Group 2) + 1(Group 5) + 1(Group 4) = 11. Thus GDF=11/15=0.73 and, hence, group decomposition proceeds to the next step.

The next task is to compute the degree of heterogeneity $H(G_i)$ (1<=i<=6) with respect to Vector B for each group and determine the rating of this state. The intervals are defined as [0,9], [10,19], [20,30]. This is shown in Table 5.

Table 5. Groups with respective $H(G_i)$ values

Group Number	*Student id*	*Vector B*	*H(Gi)*
1	1,6,13	12,13,14	0.33
2	7,10,15	6,28,11	1.0
3	3,5,14	28,07,16	1.0
4	2,4,12	06,09,05	0.33
5	8,9,11	27,28,25	0.33

The value of C corresponding to this state is 2.99. One set of learners of G1 and G4, are now swapped to improve H(G1) and H(G4). The resulting state is shown in Table 6. The swapped si1 (i=1,2) are indicated using *.

Table 6. State obtained after swapping members of group 1 and 2

Group Number	*Student id*	*Vector B*	*$H(G_i)$*
1	2*,6,13	06,13,14	0.66
2	7,10,15	6,28,11	1.0
3	3,5,14	28,07,16	1.0
4	1*,4,12	12,09,05	0.67
5	8,9,11	27,28,25	0.33

The obtained value of N corresponding to this new state is 3.66. Thus, this state is accepted as the new state and search continues until all H(Gi) are greater than 0.5 (1<=i<5). The final groups obtained in this manner are shown in Table 7.

Table 7. Final state obtained with highest rating

Group Number	*Student id*	*Vector B*	*H(Gi)*
1	8,6,13	27,13,14	0.66
2	7,10,15	6,28,11	1.0
3	3,5,14	28,07,16	1.0
4	1,4,12	12,09,05	0.67
5	2,9,11	06,28,25	0.67

Finally a check is performed to see if homogeneity is preserved using GDF. The number of learners that remain in their original groups after decomposition are

2(Group 1) + 3(Group 2) + 3(Group 3) + 1(Group 4) + 0(Group 5) = 10

Thus GDF is 9/15=0.60. Since this value is greater than 0.4, this state preserves homogeneity along the proposed degree of heterogeneity within the groups.

Implementation

The steps described in the previous section were applied to a class of post graduate students taking an "Introduction to Java Programming" course in a certain college in Kolkata, India. The class of 48 students was first taught the subject in a conventional manner. They were then given project assignments where working in groups was necessary. They were divided into two sets, the Control Set and Experimental Set, with 24 students each. The students in the Control Set were assigned projects in a traditional manner, i.e., they were asked to choose their group according to their choices. The students in the Experimental Set were assigned to projects using the methods proposed in this study. The aim of this section is to illustrate the process of collaborative group assignment to an Experimental Set using the Automated Group Development Program (AGDP) and compare its effectiveness with Control set group assignment. Each project group consisted of 4 students. Evaluation was done by the supervisor of the group corresponding to the parameters s_{i2},s_{i3},s_{i4} (1<=i<=24) out of 10 where s_{i5} (1<=i<=n) was derived using a pretest conducted on the subject. The full marks for this test was 40.

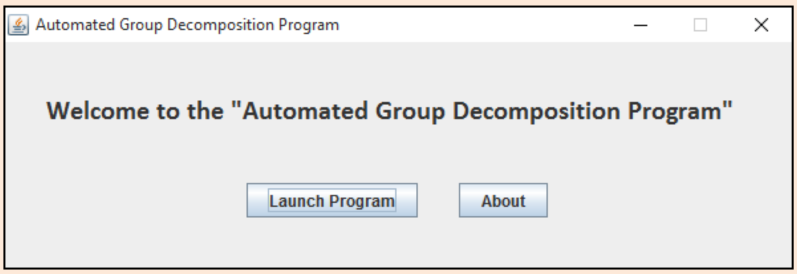

Figure 1. The welcome screen

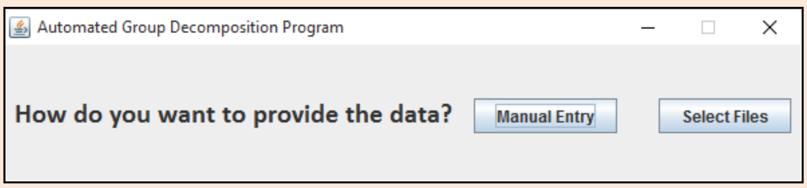

Figure 2. Data-entry option

AGDP was implemented using the Java programming language and the GUI was implemented using the light-weight Java Swing GUI widget toolkit. Fig. 1 is the first screen, which is displayed when the instructor runs this software.Two modes of data entry are used (Fig. 2).In the "Manual Entry" option, the user manually enters the data (Fig.3). Fig. 4 shows groups containing Vectors A and B along with their degree of homogeneity. In the case of the "Select Files" option, the software automatically reads data from the selected files and displays the output. This screenshot is not shown here for the purpose of brevity. The "About" button in Fig.1 illustrates the working of the software in detail.

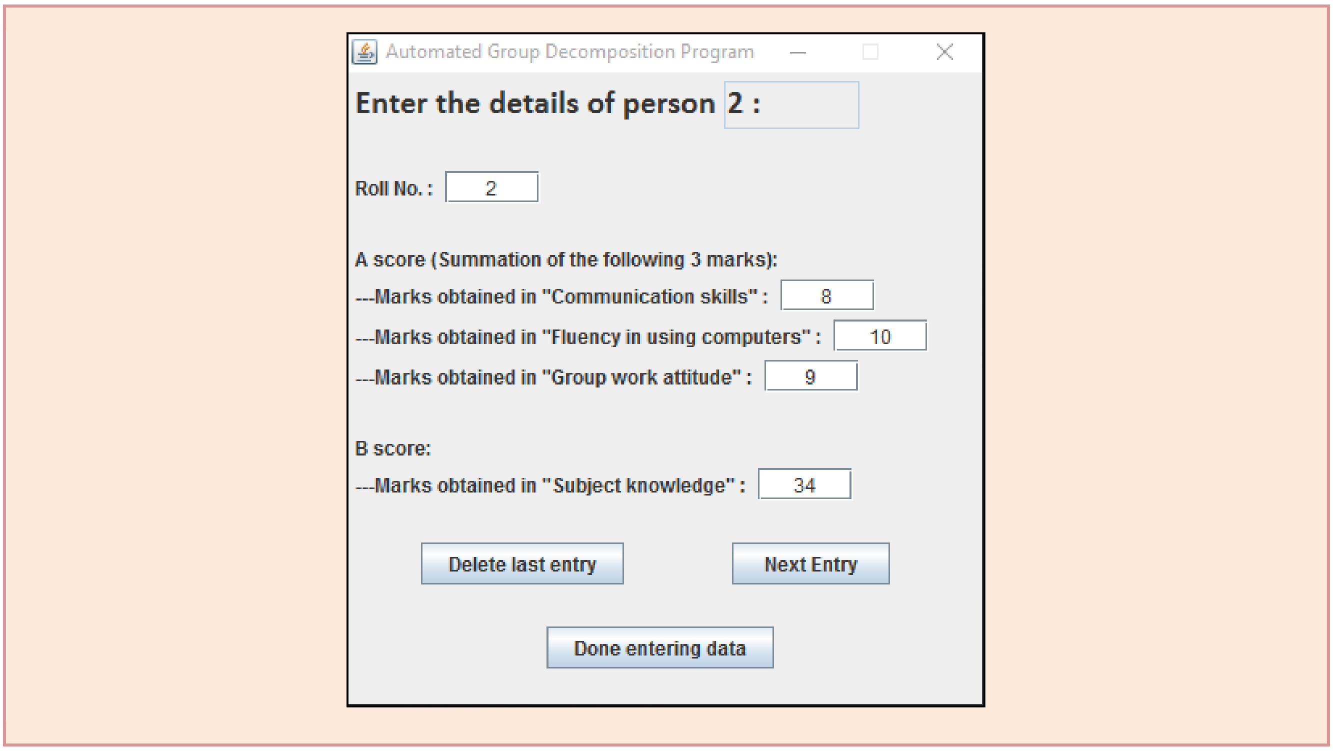

Figure 3. Manual data-entry option

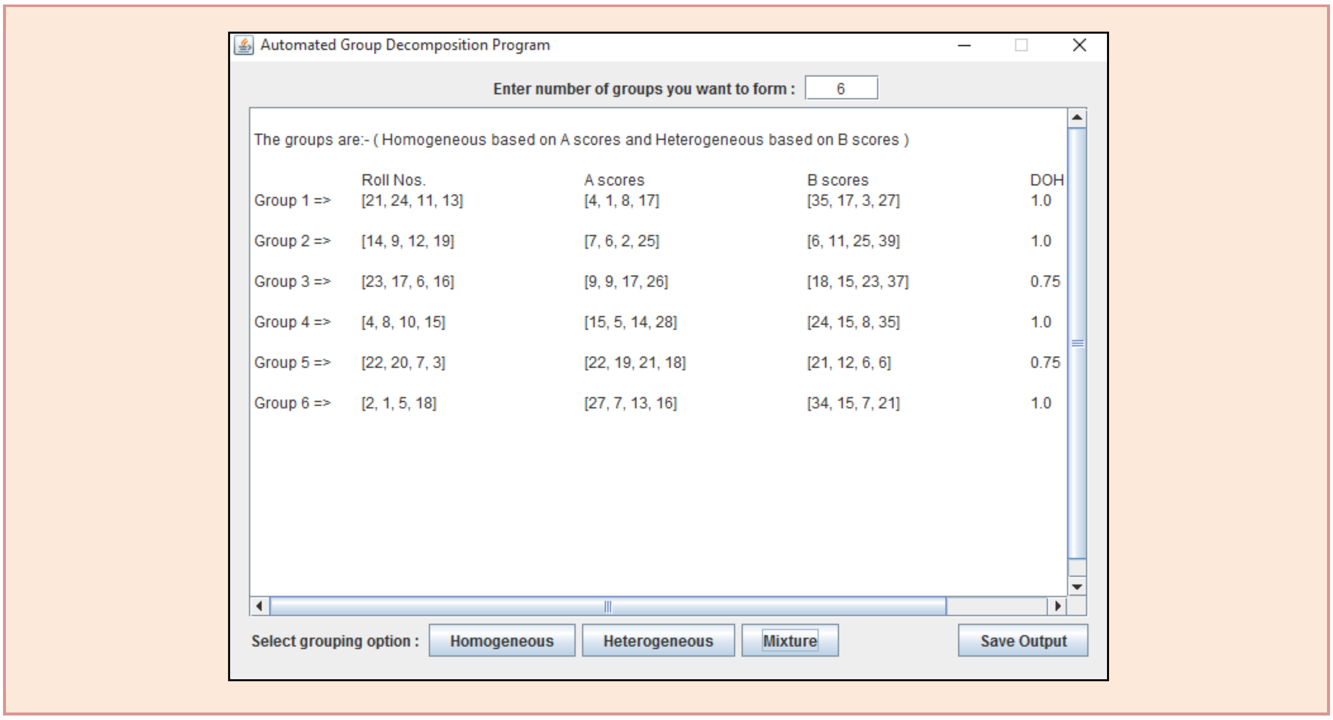

Figure 4. Group decomposition after manual data entry

The result of group decomposition obtained using the Automated Group Decomposition Program (AGDP) is now compared with results of group decomposition performed using conventional methods where students were asked to choose their own groups. After all students completed project work under their respective supervisors, they were asked to appear for a posttest. The pretest and posttest results were compared for the Control and Experimental Set using an unpaired t-test (Table 8). It was found that students whose groups were assigned byAGDP performed much better than the students who were assigned groups in conventional method.

Table 8. T-test results of pretest and post test marks for Experimental and Control Set

	Difference in mean	t-value	Two tailed p-value	Significance
Control Set	1.40	0.3859	0.7096	Statistically Not Significant
Experimental Set	8.80	2.9268	0.0191	Statistically Significant

A survey was conducted to find the attitude of the learners towards both methods of group formation. Their level of satisfaction was assessed by asking 5 questions, the answers to which were divided into a 5 point scale. The mean rating for each question is presented in Table 9. The results in Table 9 clearly show that learners in the Experimental Set were more satisfied than learners in the Control set.

Table 9. Survey results of the learner for Experimental and Control Set.

Serial #	Question	Control Set	Experimental Set
1	This method of learning enhanced collaborative activity within the group	3.2	3.8
2	The members of the group helped in identification of learning problems	3.3	3.8
3	The members of the group helped in learning the parts I have not learnt well earlier	3.6	4.1
4	The learners are assigned a group of their choice	4.1	3.7
5	This method of learning using group formation provided a new provided to learning	3.7	4.2
	Mean Rating	3.58	3.92

Comparative Study

This section compares the result of the current study with past research done in this field. The group decomposition methods for collaborative learning, which were used to compare with the present study, are the greedy algorithm proposed by Redmond [8], fuzzy C means proposed by Christodoulopoulos and Papanikolaou [9], ant colony optimization proposed by Graf and Bekele [10], and genetic algorithm proposed by Wang et al.[11]. These papers are chosen to introduce a certain degree of variation among collaborative algorithms in this study. A set of parameters are proposed in this section for comparison. These parameters also outline the step-by-step buildup of the research methodology used in developing such model. A brief description of the parameters used for comparison is presented here:

Table 10. Comparative study of proposed methodology with related works

Parameters	***Redmond [8]***	***Graf and Bekele[10]***	***Christodoulopoulos and Papanikolaou [9]***	***Wang et al. [11]***	***Present Study***
Parameters used for modeling student behavior	Preferred time slots	Group work aptitude and subject performance	Learning aptitude and communication skills	Learner thinking styles	Personal Style and Performance Indicator
Methodology used for group formation	Greedy Algorithm	Ant Colony Optimization	Fuzzy C Means	Random Mutation Hill climbing	Agenda Driven Search
Basis of group heterogeneity	Computer related job background	Total value of student personal traits	Random selection of records without replacement	Psychological parameters which affect group learning	Learner subject knowledge
"Orphan" student handling	No orphans due to fixed and predetermined group size	Not reported	No orphans due to equalization	No students ignored during group formation	No orphans due to reshuffle
Experimental data set	6 undergraduate classes each containing 30-35 records	5 sets of 100 student records constructed at random	18 undergraduate students for a course in "Digital Systems'	66 undergraduate computer science students	48 students doing post graduation in computer science
Strengths	Simple greedy approach	Using personality traits in group formation	Low computational complexity	considering psychological factors in group formation	Constructing groups which are homogeneous as well as heterogeneous
Weakness	Domain knowledge not taken into account	High number of iterations for large data sets	No theoretical justification behind achieving heterogeneity	Small sample size used for testing	Not tied up with any learning system

(i) *Parameters used for modeling student behavior*:A certain numerical basis is required to divide the learners into a set of heterogeneous groups. The numerical values of these parameters may be derived through a survey or assigned by the learner or instructor.

(ii) *Methodology used for group formation*: A certain class of algorithms is applied on parameters defined in the previous step to propose an outline of the group development method. Most of these algorithms are derived from the domain of Artificial Intelligence.

(iii) *Basis of group heterogeneity*: As discussed earlier, efficacy of group work increases many fold if the group contains learners who are heterogeneous in nature. Various researchers have proposed diverse parameters to ensure group heterogeneity. A heuristic is often applied to these parameters to determine the degree of heterogeneity of the group.

(iv) *"Orphan" student handling*: When students are asked to form groups by themselves, often certain students fail to find themselves a group. The group formation algorithm should be able to deal with such cases in an automated fashion.

(v) *Experimental data set*: Real world data as well as simulated data has been used to show the working of these developed models.

(vi) *Strengths and weakness*: Each study is based on a certain distinctive aspect of the collaborative process. However the study may not fulfill all the requirements of collaborative group formation. Each evaluated model may have some strengths and weakness.

The comparative estimate of the present study with the papers discussed is presented in Table 10.

Conclusion

After analyzing the work done by researchers in this domain, it becomes clear that most of them concentrated on creating groups which are heterogeneous in nature. This study adopts a different approach for group formation. First, homogeneous groups are created based on personal style and heterogeneity is engendered within these groups based on a performance indicator. This ensures a mixed group that is neither fully homogeneous nor heterogeneous. This process prevents any “orphan” students, which is an advantage. This is because the group size is a multiple of the total number of learners, and the reshuffle algorithm distributes the learners evenly over all groups. The distribution takes effect even if group size is not a multiple of class size. Simulations done using the Automated Group Decomposition Program (AGDP) tool indicate that the results of group decomposition are satisfactory. This conclusion was further supported by the survey results.

A major limitation of this study was the small sample size used in the simulation. The collaborative effect of groups formed from a large sample size could be initiated as done in Graf and Bekele [10]. Another limitation of the model is the inability to create groups when no initial information about personal style and performance indicators are available. This could occur when instructors have no prior information about the learners. Future work could involve considering the learners’ choices for group formation and use it as a basis for group decomposition. Only three attributes corresponding to personal style have been considered here. This set could be modified and expanded based on the needs of the instructor and the application. Finally, the AGDP application could be integrated as a part of a collaborative learning application to form groups prior to collaborative learning.

References

[1] P. Dillenbourg, “Collaborative Learning: Cognitive and Computational Approaches,” *Advances in Learning and Instruction Series*. New York, NY: Elsevier Science, Inc,1999.

[2] D. Suthers., D. Jones, “An Architecture for Intelligent Collaborative Educational Systems,” *8th World Conference on Artificial Intelligence in Education (AI&ED ’97)*, 20-22, Kobe Japan, pp. 55-62, Aug. 1997,

[3] G. J. Hwang, P. Wu, H. R. Ku, “An interacting Concept Map approach to supporting mobile learning activities for natural science courses,” *Computers and Education*, Vol. 57, pp. 2272-2280, 2011. Article (CrossRef Link)

[4] P. C. Abrami, B. Chambers, C. Poulsen, C. De Simone, S. d’Apollonia, J. Howden, “Classroom connections: understanding and using cooperative learning,” *Toronto: Harcourt Brace & Company*, 1995.

[5] R. E. Slavin, “When Does Cooperative Learning Increase Achievement? Psychological Bulletin,” Vol. 94, pp. 429–445, 1983. Article (CrossRef Link)

[6] C. Ames, R. Ames, “Research on Motivation in Education,” *Academic Press Inc.*, Orlando, USA, 1985.

[7] A. Collins, J.S. Brown, “The Computer As A Tool For Learning Through Reflection,” In: Mandl, H., Lesgold, A. (eds.): *Learning Issues For Intelligent Tutoring Systems, Springer Verlag, New York*, pp. 1–18,1988. Article (CrossRef Link)

[8] M. A. Redmond , “A Computer Program to Aid Assignment of Student Project Groups,” *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education*, Charlotte, NC, USA, 2001. Article (CrossRef Link)

[9] 9. C. E. Christodoulopoulos, K. A. Papanikolaou , “A Group Formation Tool in a E-Learning Context,” *Proceedings of the19th IEEE ICTAI’07*, pp. 117-123, 2007. Article (CrossRef Link)

[10] S. Graf, R. Bekele, “Forming Heterogeneous Groups for Intelligent Collaborative Learning Systems with Ant Colony Optimization,” *Lecture Notes in Computer Science*, 4053, pp. 217-226, 2006. Article (CrossRef Link)

[11] D. Wang, S. J. Lin, C. Sun, “DIANA: A computer-supported heterogeneous grouping system for teachers to conduct successful small learning groups,” *Computers in Human Behavior*, Vol 23, pp. 1997–2010, 2007. Article (CrossRef Link)

[12] A. Ounnas, H. C. Davis, D. E. Millard, "A Framework for Semantic Group Formation in Education," *Educational Technology & Society*, Vol. 12 (4), pp. 43–55,2009.

[13] M. Ikeda, Go, S., R. Mizoguchi, "Opportunistic Group Formation - A Theory For Intelligent Support In Collaborative Learning," *Proceedings of AI-ED 97 Artificial Intelligence in Education: Knowledge and Media in Learning Systems*, Amsterdam: IOS, pp. 166-174, 1997.

[14] M. Wessner and H. Pfister, "Group formation in computer supported collaborative learning," *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, New York: ACM, pp. 24–31, 2001. Article (CrossRef Link)

[15] J. E. De Faria, J. M. Adán-Coello, K Yamanaka, "Forming Groups for Collaborative Learning in Introductory Computer Programming Courses Based on Students' Programming Styles:An Empirical Study," *36th ASEE/IEEE Frontiers in Education Conference*, October 28 – 31, 2006, San Diego, CA.

[16] C. M. Tobar, C. De Freitas, "A Support Tool for Student Group Definition," *37th ASEE/IEEE Frontiers in Education Conference*, October 10 – 13, 2007, Milwaukee, WI.

[17] X. Wu, V. Kumar V, "The Top Ten algorithms in Data Mining," *CRC Press*, Taylor and Francis Group,2009.

[18] J. Haan, M. Kamber, "Data Mining: Concepts and Techniques," *Third Edition*, Elsevier,2011.

[19] E. Rich, K. Knight, "Artificial Intelligence, *Second Edition*," Tata Mcgraw-Hill,2005.

[20] G. Brassard, P. Bratley, "Fundamentals of algorithmics," *Pearson Education*, 1995.

Anal Acharya is Assistant Professor in the Department of Computer Science in St Xavier's College, Kolkata. He was the Head of the Department of Computer Science from the period of 01.01.2009 to 30.06.2013. His present research interests include Data Mining and Intelligent Learning Systems. He has above 17 years of experience in undergraduate and post graduate teaching & supervised several post graduate dissertations. He has several accepted papers in International Conferences and Journals.

Prof. Devadatta Sinha is currently a Professor in the Computer Science and Engineering Department of theUniversity of Calcutta, Kolkata. He was the Head of the department on several occasions. His research interests include Distributed Systems. He has over 30 years of experience and has supervised several Ph. D. students.

Anurag Sarkar is currently a student of the Post-Graduate Department of Computer Science of St. Xavier's College, Kolkata. He is currently doing research in Machine Learning and Data Mining.

Dibyabiva Seth is currently a student of the Post-Graduation Department of Computer Science of St. Xavier's College, Kolkata. He has done research on recognizing hand-gestures and also on a traffic rule violation information system and has published his findings in International Journals.

Kaustav Basu is currently a student of the Post-Graduate Department of Computer Science of St. Xavier's College, Kolkata. He has previously done research on wearable devices and traffic rule systems. He is currently doing research in the field of Artificial Intelligence and Machine Learning.