ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 8, February 2016

Ability based Collaborative Pairing and Grouping among Learners

Sushma Hans[1], S.Chakraverty[2]
Division of Computer Engineering, Netaji Subhas Institute of Technology, Delhi, India[1,2]

Abstract: ***It is well recognized that collaborative learning enhances academic performance by pooling the diverse strengths of co-participants and by providing a healthy competitive learning environment. Existing techniques for forming collaborative groups emphasize on effective transfer of knowledge between learners. However, mastery of any subject requires not just knowledge, but a distinct set of learning abilities. While forming collaborations, one must examine the scope for mutual transfer of abilities that different members possess so as to fulfill any skills gap. In this paper, we illustrate two well-known algorithms: the Stable Marriages Problem and the Kernighan-Lin algorithm to discover optimal competitive and complementary groupings between students. Experiments reveal that competitive (complementary) collaborative pairs generated by either approach yield the same pairs when the participants have matching (contrasting) performance in 50% or more parameters. The Stable Marriages approach takes lesser time to discover all possible pairs while K-Lin has the distinct advantage of generating multi-member groups.***

Index Terms: **Collaborative E-Learning, Formal Concept Analysis, Kernighan Lin Algorithm, Learning Ability, Stable Marriages Problem.**

I. INTRODUCTION

Education research and practices indicate that collaborative learning spurs discussion, facilitates mutual exchange of ideas and generates a healthy competitive atmosphere in which learners can measure up their relative strengths and weaknesses to help each other improve. The advent of e-learning as a new technology-enabled paradigm in education has given a further fillip to collaborative learning. People across the globe - cutting across age, time and space constraints- can now cooperate to leverage the benefits of diverse socio-cultural strengths.

Prior works refined techniques for knowledge acquisition and for creating collaborative groups to enable transfer of knowledge between members of a group [1] [4], [6], [7], [10], [11], [13], [16]-[18]. However, merely acquiring raw knowledge is not enough to learn and master a subject. In fact, Information and Communication Technologies (ICT) has made it possible to obtain knowledge on almost any topic with a few key strokes. What is truly required is a set of certain skills or learning abilities that enable a student to internalize the knowledge gained and apply it practically. Learners really need to acquire the necessary skills to understand, discern, apply, and extend the knowledge intelligently. For example, for a student learning the C language, it would not suffice to acquire knowledge of various programming constructs. She needs to sharpen her programming skills through logical reasoning, analytical thinking, and mathematical abilities. It is apparent that she must hone these skills in order to become a proficient coder. Traditional methods of assessment simply examine to what extent a student has reproduced whatever she has learnt previously rather than evaluating her learning abilities and thereby identifying the skills gap. It makes better sense to first identify the specific skills or learning abilities that are required to learn and apply the various concepts of a subject. Then, students can be assessed in a more rational manner in terms of these qualitative skills. Thereafter, one can create fruitful collaborations by clubbing together students based on their skills repertoire. Our aim is not merely to transfer knowledge but to build up the core learning abilities in each pupil and improve their caliber through the mutual transfer of skills.

The rest of the paper is structured in the following manner. In section 2, we review prior works that address the problem of creating learner groups. Section 3 gives the theory behind the approaches and techniques utilized in this paper. Section 4 expounds upon our proposed scheme. We demonstrate our experimental results in Section 5. We conclude with our work in Section 6.

II. RELATED WORK

Three types of grouping strategies are generally recognized: homogeneous, heterogeneous, and mixed. Let us take a tour of the various approaches discussed in the literature to generate collaborative groups.

(i) ***Homogeneous Grouping:*** In this scheme, students with the same interests, abilities, or experience form a collaborative group. It is useful when students want to apply their joint capabilities to accomplish a specific aim or to create a healthy competition environment for their academic development. Homogeneous grouping can be easily cast as a clustering problem. The authors have applied various techniques such as K-means clustering method [13], Naive Bayesian method [17], fuzzy rough approximation approach [7] and meta-heuristic algorithms [10] - [11] to collaborate learners in a homogeneous manner.

Jin et al. [13] create learner groups based on learners' personality and learning behavior. The authors consider sixteen factors to judge each learner's personality, such as dominance, reasoning, emotional stability, sensitivity, *etc*. They propose a behavior model from learners' activities under different learning modes such as during courseware learning, doing homework and while answering test questions. However, Jin et al. have not described how behavioral information, which is large and changes continuously, can be captured dynamically.

Yang et al. [17] apply a Naive Bayesian classifier to group learners. They identify four static attributes of a learner- learning period, the region where the student lives, age group and personality values such as social and political attributes, to establish effective groups. The main snag in this paper is the use of static attributes including the number of members of a group which is fixed. Whereas we envisage dynamic attributes and flexible group formations.

In [7], the author applies a fuzzy rough approximation approach to cluster learners based on their web access patterns. The work in [10] proposes a fuzzy Particle Swarm Optimization (PSO) based clustering method to build groups considering the learners' cognitive styles as inferred from their activity logs. They consider two major criteria for better cluster performance (i) compactness of each cluster and (ii) the separation between them. Their results show that learners in the same cluster are more alike through this method as compared with k-means, c-means, and evolutionary fuzzy clustering algorithms.

(ii) Heterogeneous Grouping: The downside of homogeneous grouping is that it focuses only on the similarity between learners, who thereby have less scope to learn from each other. Heterogeneous grouping helps learners to improve academically by clubbing learners with different behavior and interests. This is desirable when there is a broader range of tasks to be carried out.

In [4], [11], the authors propose a mathematical model that maps both personality and performance attributes into a single learner vector space. In order to maximize the heterogeneity of groups generated, they use a fitness function based on three factors:. Goodness of heterogeneity (GH) values, coefficient of variation based on all GH values and Euclidean distance between groups. An ant colony optimization based approach is applied to maximize the fitness function and widen the heterogeneity of the groups. Gogoulou et al. [11] offer a tool called OmadoGenesis, that works on learner-attribute vector space to achieve homogeneous and heterogeneous grouping.

Paredes et al. [16] introduce a heterogeneous grouping technique in which a student's learning style is represented as a tuple with four dimensions given in the Felder-Silverman Learning Style Model. This supervised approach is aided by visualization tool to generate good heterogeneous groups.

***(iii) Mixed Grouping*:** This strategy allows pairing of students homogeneously on some attributes and heterogeneously on another set of attributes or students with different ability level in the same attribute collaborate to form a larger group of students. In [18], the authors demonstrate a technique for group formation based on scholars' programming skills using a genetic algorithm. This paper forms balanced groups of learners where each group consists of students' with good, moderate and poor programming skills.

The work in [6] proposes a strategy to form groups using online social networks. The system starts with the choice of a small circle of possible learners from an underlying social network using breadth first, random walk and best connected search strategies. The authors use genetic algorithm to distinguish the best group constellations using a group fitness criteria that include (i) a common learning style, (ii) a high score in the knowledge ranking and (iii) a low distance in the social web. In [1], the authors use an evolutionary algorithm to form mixed groups using learners' learning styles and ranking in programming skills. A validation performed by students and instructors of a class shows 30% better results as compared with random groups.

An analysis of above-mentioned methods reveal that prior works consider the inherent attributes of learners such as their learning style, their personality, online behavior and test performance as the basis of grouping. However, none of these approaches consider learners' academic abilities such as linguistic ability, analytical ability *etc.* as factors for forming collaborations. We endeavor to put together learners with diverse learning abilities so as to initiate a process of skills transfer.

The proposed approach has several spin-off benefits. Firstly, many basic skills are commonly required to grasp different topics of a subject. Therefore, when students improve their proficiency in a particular skill, they actually gain strength on all those concepts. Secondly, the basic skills may span different subjects. Hence, skills enhancement can enable a student to grasp and master different subjects and facilitate interdisciplinary academics. Lastly, we define performance parameters as a combination of various learning abilities so that students can apply them jointly to solve many complex problems.

III. THEORETICAL BACKGROUND

We now describe the central concepts that have been utilized in our work.

A. Kernighan-Lin (K-LIN) Algorithm

The K-Lin algorithm is an iterative partitioning algorithm. It follows a greedy approach to partition a graph of $2n$ vertices into two disjoint arbitrary subsets of n vertices each that are connected together in an optimal way such that the external cost S of the weightiness of the edges between nodes in X and Y is minimized [15]. Let I_x be the internal cost of a vertex x *i.e.* sum of the cost of edges between x and other nodes in X and E_x be the external cost of x *i.e.* sum of the cost of edges between x and nodes in Y. The internal and external costs are computed using equation 1 and 2 respectively.

$$I_x = \sum_{a \in X} C_{xa}$$

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 8, February 2016

(1)

$$E_x = \sum_{b \in Y} C_{xb}$$
(2)

The difference between internal and external cost of x is given below in *eq.* 3.

$$D_x = E_x - I_x$$
(3)

Similarly, we can compute E_y, I_y and D_y for another vertex y. Now if $x \in X$ and $y \in Y$ are interchanged, the reduction in cost, *i.e.* g_{xy} is [15]:

$$g_{xy} = S_{old} - S_{new} = D_x + D_y - 2C_{xy}$$
(4)

The algorithm executes a series of interchange operations between elements of X and Y greedily to maximize the overall gain, finally producing a re-partitioned X and Y. The time complexity for K-Lin algorithm is $O(n^{2.4})$ [14].

Note that a single invocation of K-Lin creates two almost equal partitions. However, in our proposed scheme, we aim to pair up students. Therefore, we make recursive calls to K-Lin to execute a divide and conquer strategy until we get the pairs of collaborating students.

B. Stable Marriage Problem

The Stable Marriages Problem (SMP) determines a stable matching between members of two different groups [9]. The SMP algorithm works with a set of men and a set of equal number of women, in which each man passes on her preference for each woman and vice versa. A pairing is stable if there is no unmatched man-woman pair (*m, w*), where man *m* gives preference to woman *w* to his assigned partner, and woman *w* prefers man *m* to her assigned partner. After several rounds of SMP pairing, an ordered preference list is mapped to all members of the opposite gender.

In its essence, the algorithm conducts a series of proposals from the men to the women. In each iteration, an unassigned man *m* looks for the next preferred woman *w* determined from his preference list. Each woman goes for the best proposition for now she receives during the process. If the proposed woman *w* is free, then man *m* got engaged to woman *w*. But if *w* is already engaged with some other man *m'*, there can be two cases:

If *w* prefers *m* to *m'*, she replaces her current partner *m* to the man *m'* with a new proposal. After this, *m* becomes unengaged and joins the list of unmatched men. *m* again search for a stable pair further down the ordered preference list in succession in a further iterations.

If *w* gives high preference to her current partner *m'*, women *w* remains engaged with *m'*. Then *m* will look next woman *w'* from his preference list.

An analysis of the SMP algorithm shows that we get stable pairs for all the men and women at the end of its iterative operations. Although the SMP algorithm runs quite fast, often in almost linear time, it has a worst-case time complexity of θ (n^2), as the algorithm may have to process most of the inputs, *i.e.* $2n$ preference lists each of length n, for an instance involving n men and n women [12].

IV. ABILITIES BASED COLLABORATIVE E LEARNING SCHEME

We now describe our proposed e-learning system named Learning Ability driven Collaborative E-learning System (LACES). At the outset, subject experts identify different types of learning abilities or skills that are required for learning each topic or concept that is covered in the syllabus of a course. Using this as a guideline, a database of questions is prepared in a systematic manner. Each question is framed to test the understanding and application of one or a set of related concepts. Accordingly, the pre-determined set of abilities required for its correct resolution are derived.

Now, a student can answer a question correctly only when she has the knowledge of that concept as well as the learning abilities that are required to apply them well enough. The aim of LACES is to assess the basic learning abilities acquired by each student and form collaborative pairs of students based on their parity or complimentary abilities. The system takes care of the fact that answering correctly just one or a few questions requiring a particular skill does not indicate her full proficiency in applying that skill. She needs to answer multiple questions covering different concepts that require common skills in order to conclusively demonstrate her ability.

It would be a very slow procedure to examine each student's performance under these guidelines if we follow traditional methods of evaluation. We develop a smart evaluation procedure that assesses students along a two-dimensional concept-lattice of questions and their associated learning abilities. Let us now examine the steps involved in this procedure. In [19], we describe the complete process of determining the sets of skills named as Performance Parameters (PPs) required for the various concepts of a course. Now, we describe the two ways of collaborating learners to enhance their skills.

A. Forming Collaborative Pairs

Let us assume that there are n students S = {s_0, s_1……, s_{n-1}} who have registered for the course. We employ the SMP and the K-Lin algorithms to generate competitive and complementary pairings between students. When the course starts, the initial decision about the students' abilities is derived by conducting a pre-test with generic questions that test all the abilities required for the course. Subsequently, as the course progresses,

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 8, February 2016

tests are conducted at regular intervals based on the topics covered till that point of time. Students are assessed based on their latest test performance.

- ***Rules for Pairing***

Competitive and Complementary pairing schemes have different guidelines for pairing up students:

Rules for Competitive Pairing: In competitive collaborations, students who have similar performance levels in various PPs are paired together so that they can be pitted against each other to improve. We start by comparing the performance of each pair of students individually on each of the PPs derived from the question bank. The rules for forming combinations for this kind of pairing are given in Table 1.

Rules 1 to 3 are the favorable rules for competitive pairing and therefore assign a higher pairing weight *W*=3. For example, pairing rule 1 pairs two students who are both *weak* in the given PP. This rule is assigned a high pairing weight *W*=3. Rules 4 to 6 are assigned *W*=0 as the collaborators have dissimilar performance. Hence, they are assigned pairing weight *W= 0*.

Table 1: Competitve Pairing Rules for student pairs for a given PP_k

Rule No.	Student Pair Performance combination for PP_k	Pairing Weight $W(k)$
1	Weak-Weak	3
2	Good-Good	3
3	Average-Average	3
4	Average-Good	0
5	Average-Weak	0
6	Good-Weak	0

Rules for Complementary Pairing: Complementary pairs are geared to fulfill the skills gap. If a student is *good* at some PP, but lacks in another, she should be paired with a student with the complementary PPs so that both get an opportunity to benefit from each other and grow.

Table 2 elaborates the rules for this pairing scheme. In this pairing scheme, we consider two PPs, say PP_x and PP_y. With three performance levels: *good*, *weak*, and *average* that a student can achieve in a given PP, we apply distance measure d_k that measures the difference in the performance levels of the two collaborators. The distance d_k=1 for *good-average* and *average-weak* combinations and d_k =2 for *good-weak* combination. The value of d_k for *average-average*, *good-good* and *weak-weak* combination are all *zero*.

The pairing weight for each pairing rule is computed by adding up the values of d_x and d_y for PP_x and PP_y respectively. From Table 2, we can see that the first rule presents the maximum contrast between matched performance levels. The value of d_x as well as d_y is 2 in this rule. Therefore, its Pairing weight is assigned the value $W(x,y) = d_x + d_y = 4$. All other combinations have lesser differentiation between the matched performance levels.

Table 2:Complementary Pairing Rules for ability based student pairs for a given pair PP_x and PP_y

Rule No.	**Student-pair Performance Combination In PP_x and PP_y**	Pairing Weight $W(k)$
1	PP_x: Good-Weak PP_y: Weak-Good	4
2	PP_x: Good-Weak PP_y: Average-Good	3
3	PP_x: Good-Weak PP_y: Weak-Average	3
4	PP_x: Good-Average PP_y:Weak-Good	3
5	PP_x: Average-Weak PP_y: Weak-Good	3
6	PP_x: Good-Weak PP_y: Average-Average	2
7	PP_x: Good-Average PP_y: Average-Good	2
8	PP_x: Average-Weak PP_y: Weak-Average	2
9	PP_x: Average-Average PP_y: Weak-Good	2
10	PP_x: Good-Good or Weak-Weak Any other combination	0

Generating Preference Weights

The pairing rules give a basis for finding out the overall preference any student will have for pairing up with any other student in a collaborative scheme.

Competitive pairing

The aim of the competitive pairing scheme is to maximize the similarity between members of a collaborating pair considering all PPs. Given two students si and sj, the system calculates the overall Preference Weight PWi,j as the cumulative pairing weights of individual PPs. Thus, if nPP is the total number of PPs, we get:

$$PW_{i,j} = \sum_{k=1}^{n_{pp}} W_{i,j}(k) \qquad (5)$$

Complementary pairing

The preference weight computation for complementary pairing is more involved. We may note that students in a pair are free to learn from each other on any performance parameter in a mutually beneficial manner so as to hone the necessary skills.

The system checks the performance levels of s_i and s_j for each pair of PPs. With n_{pp} total PPs, we

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 8, February 2016

get a maximum of ${}^{n_{pp}}C_2$ possible pair of PPs. The appropriate rule that is triggered by a given PP pair is found from their performance levels and Table 2. The corresponding pairing weight cumulatively added to the Preference

Weight *PW*:

$$PW_{i,j} = \sum_{k=1}^{n_{pp}-1} \sum_{l=k+1}^{n_{pp}} W_{i,j}(k,l) \tag{6}$$

Forming Collaborations

After generating all preference weights, the system starts forming collaborative pairs. The system takes different routes to generate pairs depending upon the approach.

Collaborative Pairing with SMP

The conventional SMP algorithm has been described with two distinct groups of n men and n women. However, in our case a single set of students register for a course and collaborations need to be forged within that group. Therefore, the same set of students is replicated.

Given the preference weights calculated in previous sub-section using *eq*. 5 and 6, LACES generates an $n \times n$ Ordered Preference Table (OPT) for n students. The SMP algorithm utilizes this OPT for determining $n/2$ collaborative pairs.

To allow pairing within the same group of registered students, the OPT is replicated as OPT_A and OPT_B. Initially, all the students of both the groups as *unpaired*. The SMP algorithm then initiates an iterative process of pairing. Taking each student s_i in turn, it finds the most preferred student s_j for s_i from the OPT that has not yet been tried for pairing provided it is not the same student. If s_i happens to have the same preference weight with two or more students, then the system assigns the first student encountered.

If s_j is unpaired so, then a collaborative pair is made with s_i and s_j. However, it is possible that during the course of forming earlier pairs, s_j was already paired with some other student s_k. The algorithm then checks whether s_j has more preference for s_i than for its currently paired student s_k. If s_i has more preference, s_i and s_j are paired together and s_k is set as *unpaired*. If this criterion is not satisfied, s_j remains paired with s_k and the system restarts to find another student for pairing with s_i.

In its current implementation, the SMP algorithm cannot generate collaborative groups of size greater than 2 students.

Collaborative Grouping with K-Lin

The algorithm for creating collaborative pairs through iterative K-Lin is given in Fig. 1. Let us first assume the group_size f=2 in order to create pair-wise collaborations. The collaborative grouping problem is modeled as a weighted graph Ḡ (V, E) whose vertices represent students and edges denote their mutual preferences (step 3). The weight of the edge ($E_{i,j}$) from student vertex v_i to v_j is computed by adding the preference weight $PW_{i,j}$ of v_i for v_j and preference weight $PW_{j,i}$ of student s_j for s_i :

$$E_{i,j} = PW_{i,j} + PW_{j,i} \tag{7}$$

The K-Lin algorithm initially divides n students into any two arbitrary partitions S_A and S_B of size $n/2$ (step 4). The execution of steps from 6 to 15 gives us two groups of roughly size $n/2$ in which students with high preference weights for each other are tried to put together. The algorithm is called recursively for each sub-group so formed, until pairs are generated. The above process can be used to generate group sizes > 2 by inputting the required group size (steps 16 and 17). Thus K-Lin algorithm has the advantage that we can terminate the partitioning process at any level to get the desired group size.

The time complexity of K-Lin to form an optimal partitioning is $0(n^{2.4})$. However, collaborative groups are formed by repeatedly invoking K-Lin for any prescribed size $\geq$ 2. Therefore, the overall complexity is a summation of individual iterations. Starting with n students segregated into two groups, for l partitioning steps, K-Lin yields $\left\lceil \frac{n}{2^l} \right\rceil$ groups of roughly the same size.

KLin_grouping (.)
Input: OPT, set of students S, number of students n, Group_size *f*
Output: Collaborative groups
Steps:

1. Initialize: partition_size = n/2
2. $\forall s \in S$, flag[s] = 0,
3. Create Ḡ(V, E), where |V|=|S| and assign edge weights using Eq. 7.
4. Divide the students set S into two equal sets SA and SB such that $SA \cap SB = \varphi$ and $SA \cup SB = S$
5. **Repeat {**
6. $\forall v \in V$ Calculate D_V using Eq. 3.
7. **For** (i = 1 to partition_size) {
8. Calculate the gain $g_{a,b}$ for each pair (v_a, v_b), $v_a \in VA$ and $v_b \in VB$ using Eq. 4
9. Find the maximum gain $g_{l,k} = Max\{g_{a,b}\}_{a,b}$ and Set flag[v_l]=1, flag[v_k]=1.
10. $\forall v \in V$ such that flag[v] = 0, compute new values of D_S
11. Find j, such that $G_j = \sum_{i=1} g_i$ is maximized
12. **If** $G_j > 0$ **then** Shift $s_{a1}, s_{a2}, \ldots\ldots, s_{aj}$ from SA to SB and $s_{b1}, s_{b2}, \ldots\ldots, s_{bj}$ from SB to SA
13. Set flag[S] = 0, $\forall s \in S$
    }
    }
14. **Until** $G_j \leq 0$
15. **If** (partition_size > f) {
16. Call *KLin_Pairing(OPT, SA, partition_size)*
17. Call *KLin_Pairing(OPT, SB, partition_size)* }
18. Output all partitions obtained as collaborative groups.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 8, February 2016

B. *Quality Metrics*

Quality Metric for Competitive Groups

The quality of a competitive group G is given by the parity in performance of its members in various PPs. In order to determine the maximum similarity in performance for a given PP_k. We need to identify the most frequently occurring performance level in PP_k among students in a group. Thus, forming sets of students with same performance level *l*, in PP_k:

$$Z_l(k) = \{(s_a, s_b) | \ PT(a,k) = PT(b,k)\} \qquad (8)$$

The set with the maximum cardinality is given by:

$$Z_I(k) = Max_l |Z_l(k)| \qquad (9)$$

This represents the largest group of students having the same performance level for PP_k. Let QC denotes the Quality of Collaboration of a group. The factor *QCCompetitivePP*(*k*) represents the quality of collaboration of the competitive group on PP_k computed as given below:

$$QCCompetitivePP\ (k) = \frac{Z_I(k)}{\sum_l Z_l(k)} \qquad (10)$$

The QC for the competitive group G is given by:

$$QCCompetitive\ (G) = \sum_{x=1}^{n_{pp}} \frac{QCCompetitivePP\ (x)}{n_{pp}} \times 100 \qquad (11)$$

A group will be totally competitive if *QCCompetitive*(*G*) = 1, when all students have the same performance level in each of the PPs, *i.e.* they are at par in every aspect. The relatively high value of *QCCompetitive* indicates a correspondingly high degree of parity in the performance levels of the collaborators in the group.

Quality Metric for Complementary Groups

Let us take a group of students G = $\{s_1, s_2, s_3 \ldots\}$. Since any student can interact with any other student in the group, therefore, we consider all possible pair of students in a group. The total number of pairings that are possible among this group is $|G|C_2$.

In addition, student in a pair can share their knowledge on any performance parameter in a mutually beneficial manner so as to enhance their abilities. There are ρ = $n_{pp}C_2$ pairs possible between each pair of students on which they can collaborate with each other. Each such PP pair triggers one of the rules in Table 2. Let *rule*(*STpair*$_j$, *PPpair*$_k$) is triggered by *PPpair*$_k$ in a student pair *STpair*$_j$. Let W be the pairing weight assigned to this rule that is determined by referring Table 2. We take percentage difference on all possible PP pairs for each pair of students to determine the quality of collaboration of a complementary group (QCComplementary) as given in *eq*.12.

$$QCComplementary\ (G) = \frac{\sum_{j=1}^{|G|_{C_2}} \sum_{k=1}^{\rho} W(rule(STpair_j, PPpair_k))}{|G|_{C_2} \times \rho \times 2} \times 100 \qquad (12)$$

The value 2 in the denominator is the largest integer that is less than the average of all the preference weights assigned to various rules. As we have assigned 4, 3, 2, and 0 preference weight for various rules, therefore the average is 2.25. As 2 is the largest integer that is less than the average, therefore we have taken 2.

V. EXPERIMENTAL RESULTS

The LACES framework was coded in C language using Dev C++ version 5.3.0.4. We performed our experiments on an Intel core i5 machine with 2.40 GHz processor running Windows 7. We used the running example taken up in section 4 as a synthetic dataset for conducting our experiments.

A. Quality of Pairings

The first experiment aims to check the quality of the competitive (eq.11) and complementary (eq. 12) pairs generated by LACES. We prescribed group sizes of 32, 64 and 128 students. Table 3 lists the competitive and complementary pairs generated with the group size set equal to 32. The main observations are summarized below.

For the competitive pairing scheme:

1. K-Lin and SMP generate the same pairings for 9 students: $S_0, S_1, S_2, S_4, S_6, S_{12}, S_{13}, S_{16}, S_{23}$.
2. The pairs generated by both the algorithms coincide exactly in case where the *QCCompetitive* value of the pairs lies in the range of 72.2 to 100.
3. We consider those as *good* quality pairs where value of *QCCompetitive* of the pair ≥ 50.
4. In addition, the results of competitive pairing in Table 3 confirm that both the schemes generate good quality pairs for all the students except S_{15} - S_{17} and S_{17} - S_{29} pair.

In case of complementary pairing scheme:

1. There are 10 common pairs generated by both the pairing algorithms. These are pairs for students: $S_0, S_3, S_4, S_5, S_6, S_9, S_{10}, S_{11}, S_{12}$, and S_{20}.
2. The pairs generated by both the algorithms coincide exactly in case where the *QCComplementary* value of the pairs lies in the range of 67.97 to 100.
3. We prescribe those pairs as *good* quality pairs where its value of the factor

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 8, February 2016

QCComplementary ≥ 50.

4. Both SMP and K-Lin algorithms generated 7 such good quality pairs from the available set of students.

We tested the pairing obtained with group sizes of 32, 64 and 128 students and confirmed that both these algorithms give comparable results for either pairing scheme. As we can see that both the algorithms give 15 good quality competitive pairs, whereas only 7 good quality complementary pairs. We can say that the quality of pairs depends on the available students set. The results show that this students data set supports competitive pairing as compared to complementary pairing. We will get results that are more refined for competitive pairing if there are more similar performance students in the group. The quality of the complementary pairs is augmented if there are students in the group with contrasting abilities.

Table 3: Competitive and Complementary pairs generated through SMP and K-Lin

S.No	Competitively Paired students using				Complementarily Paired students using			
	SMP	QCCompetitive	K-Lin	QCCompetitive	SMP	QCComplementary	K-Lin	QCComplementary
1	S0 - S28	100	S0 - S28	100	S0 - S16	81.04	S0 - S16	81.04
2	S1 - S9	72.2	S1 - S9	72.2	S1 - S17	31.04	S1 - S14	1.31
3	S2 - S27	94.4	S2 - S27	94.4	S2 - S30	22.22	S2 - S17	29.41
4	S3 - S30	94.4	S3 - S21	88.9	S3 - S7	62.09	S3 - S7	62.09
5	S4 - S31	83.3	S4 - S31	83.3	S4 - S8	33.01	S4 - S8	33.01
6	S5 - S25	77.8	S5 - S8	66.6	S5 - S19	80.72	S5 - S19	80.72
7	S6 - S19	100	S6 - S19	100	S6 - S25	91.83	S6 - S25	91.83
8	S7 - S29	55.5	S7 - S18	50	S9 - S15	21.24	S9 - S15	21.24
9	S8 - S22	61.1	S22 - S25	55.5	S10 - S13	67.97	S10 - S13	67.97
10	S10 - S21	66.6	S10 - S30	72.2	S11 - S29	29.41	S11 - S29	29.41
11	S11 - S18	94.4	S11 - S15	77.8	S12 - S22	41.83	S12 - S22	41.83
12	S12 - S14	77.8	S12 - S14	77.8	S14 - S18	0	S18 - S21	9.80
13	S13 - S20	77.8	S13 - S20	77.8	S20 - S23	70.59	S20 - S23	70.59
14	S15 - S17	33.3	S17 - S29	22.2	S21 - S27	10.46	S27 - S28	47.71
15	S16 - S24	94.4	S16 - S24	94.4	S24 - S28	67.97	S24 - S26	66.99
16	S23 - S26	100	S23 - S26	100	S26 - S31	41.50	S30 - S31	41.18

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 8, February 2016

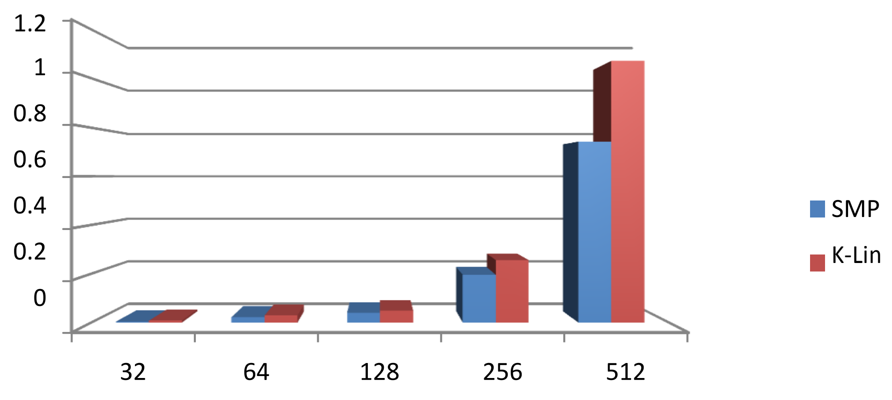
Time Complexity

The time complexity for K-Lin algorithm is O ($n^{2.4}$) [14]. The SMP algorithm works often in almost linear time, but it has a worst-case time complexity of θ (n^2), as the algorithm may have to process most of the inputs, *i.e.* 2*n* preference lists each of length *n*, for an instance involving n men and *n* women [12]. In order to measure the time taken for execution exactly for both the algorithms, we increased the group size in powers of 2.

Table 4 shows the execution times taken by the SMP and K-Lin based pairing methods with increment in group size. We can see from Fig. 2 that for 32, 64 and 128 students, both algorithms takes almost same time for pairing them. As we increase the number of students beyond that, the time taken by K-Lin algorithm increases drastically as compared with SMP.

Table 4: Time taken in pairing v/s group size by SMP and K-Lin

Number of students / Methods	SMP (time in seconds)	K-Lin
32	0.004	0.008
64	0.022	0.029
128	0.04	0.049
256	0.196	0.256
512	0.743	1.071



X-axis: Number of students in the quality group

Y-axis: Time taken in pairing in seconds

Fig. 2: Time taken in pairing v/s group size

K-Lin Groupings

The K-Lin algorithm is called recursively in our proposed K-Lin based collaborative scheme to generate groups which can be stopped after reaching a pre-set group size. In this experiment, we generate competitive and complementary groups of size 4. We can get groups of size 8 and 16 also. However, these group sizes will be very large and such a big group may not contain students who have similarity in their performance levels in various PPs to form a competitive scheme or contrasting performance levels to form complementary scheme. Therefore, we illustrate this experiment of generation of bigger groups using K-Lin with group size 4.The findings in Table 5 show that

– In competitive groups, all the groups generated have their *QCCompetitive* value greater than 70.8 that assures the good quality of competitive groups.

– Considering the case of complementary groups, all the groups except the first one are not satisfying the criteria of good quality complementary pairs. That again clears this data does not support complementary grouping as proved in experiment 1 also.

In case of complementary grouping, we determine that each group is comprised of students in such a way that if a student in the group is weak in a PP, then the group always contain a student who is good or average in that PP. This means that a student who is weak in a PP always has the scope to get help and improve in that PP that is not always possible in case of pairing students.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 8, February 2016

Table 5: Groups generated using K-Lin

S. No	Competitive group				*QCCompetitive*	Complementary group				*QCCompleme ntary*
1	S6	S16	S19	S24	94.4	S5	S6	S19	S25	57.51
2	S2	S11	S15	S27	90.3	S0	S16	S24	S26	47.55
3	S3	S10	S21	S30	87.5	S10	S13	S20	S23	44.39
4	S4	S7	S18	S31	79.2	S3	S7	S30	S31	35.57
5	S22	S23	S25	S26	79.2	S2	S12	S17	S22	25.05
6	S0	S17	S28	S29	76.4	S9	S15	S27	S28	17.76
7	S12	S13	S14	S20	72.2	S11	S18	S21	S29	13.29
8	S1	S5	S8	S9	70.8	S1	S4	S8	S14	14.16

VI. CONCLUSION

In this paper, we set out with a novel idea of doing a formal analysis of a course to see the important performance parameters of the course. These sets of abilities are considered as performance parameters that are necessitated by each student to learn various concepts of the course. The proposed system focused to inculcate all these performance parameters in a student who was registered for the course instead of just providing the conceptual knowledge of the course. These efforts of the system made the students ready for any kind of problem or in any field where these sets of abilities are required. We had applied Formal Concept Analysis technique to generate these performance parameters of the course. The system had taken these performance parameters as pairing attributes and had worked out on two types of pairing schemes, i.e. competitive and complementary pairing scheme for e-learners based on their demands. We had implemented SMP and K-Lin algorithm to generate these pairs of scholars. It is worthwhile mentioning that both the algorithms give comparable results in determining suitable pairs of the students. Besides, the time consumed by both the algorithm is almost same for smaller group size. Nevertheless, the time complexity of K-Lin algorithm increases drastically with bigger group size. SMP algorithm overpowers K-Lin in terms of time complexity. A big advantage of K-Lin algorithm is that we can get bigger collaborative groups through K-Lin algorithm that is not possible through SMP algorithm.

For future work, we will try to test the algorithms on a large and varied dataset. We will also try to implement it on a real e-learning system to determine the feasibility of the evaluation and collaborative technique. In addition, we will work on implementing a realistic method to determine ability information of students that can give us more insight about the learner's skills in using a set of abilities to solve a problem. We identify the feasibility and practicality of our technique by comparing this with other upcoming skills based grouping techniques.

REFERENCES

[1] Abnar S., Orooji F., & Taghiyareh F. (2012). An Evolutionary Algorithm for Forming Mixed Groups of Learners in Web based Collaborative Learning Environments. International Conference on Technology Enhanced education (ICTEE), IEEE, 1-6.

[2] Analytical skill, http://en.wikipedia.org/wiki/Analytical_skill, Last accessed: 15 October 2014.

[3] Andrews, S. (2009). In-Close, a fast algorithm for computing formal concepts. Seventeenth International Conference on Conceptual Structures (ICCS). Moscow.

[4] Bekele, R., & Sabine, G. (2006). Forming Heterogeneous Groups for Intelligent Collaborative Learning Systems with Ant Colony Optimization. Intelligent Tutoring Systems Lecture Notes in Computer Science (4053), 217-226.

[5] Belohlavek, R. (2008). Introduction to Formal Concept Analysis. Springer, Olomouc.

[6] Brauer S., & Schmidt T. C. (2012). Group Formation in eLearning-enabled Online Social Networks. 15th International Conference on Interactive Collaborative Learning (ICL), IEEE, 1-8.

[7] Chen, C. (2009). A Fuzzy Rough Approximation Approach for Clustering User Access Patterns. WRI World Congress on Software Engineering WCSE '09(1), IEEE, Xiamen, 276-280.

[8] Cimiano, P., A. Hotho, A., & Staab, S. (2005). Learning Concept Hierarchies fromText Corpora using Formal Concept Analysis. Journal of Artificial Intelligence Research (24), 305-339.

[9] Gale, D., & Shapley, L. (1962). College Admissions and the Stability of Marriage. The American Mathematical Monthly , 69 (1), 9-15.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 8, February 2016
[10] Ghorbani, F., & Montazer, G.A. (2012). Learners Grouping Improvement in E-learning Environment using Fuzzy Iinspired PSO Method. 6th National and 3rd International conference of e-Learning and e-Teaching ICELET2012, Tehran, 65-70.

[11] Gogoulou, A., Gouli, E., Boas, G., Liakou, E., & Grigoriadou, M. (2007). Forming Homogeneous, Heterogeneous and Mixed Groups of Learners. Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level, 11th International Conference on User Modeling, 33-40.

[12] Gusfield, D. & Irving, R. W. (1989). The Stable Marriage Problem: Structure and Algorithms. P. 54, MIT Press.

[13] Jin, D., Z. Qinghua, Z., Jiao, D., & Zhiyong, G. (2006). A Method for Learner Grouping Based on Personality Clustering. Proc. of the 10th International Conference on Computer Supported Cooperative Work in Design. IEEE, Nanjing, 1-6.

[14] Kernighan, B. W., & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. Bell Systems Technical Journal (49), 291-307.

[15] Kernighan-Lin Algorithm http://en.wikipedia.org/wiki/Kernighan%E2%80%93Lin_algorithm, Last accessed:15-10-2014.

[16] Paredes, P., Ortigosa, A., & Rodriguez, P. (2010). A Method for Supporting Heterogeneous-Group Formation through Heuristics and Visualization. Journal of Universal Computer Science , 16 (19), 2882-2901.

[17] Yang, Q., Zheng, S., Huang, J., & Li, J. (2008). A Design to Promote Group Learning in e-learning by Naïve Bayesian. International Symposium on Computational Intelligence and Design, ISCID '08 (2), Wuhan, 379-382.

[18] Zhamri C. A., & Yasin A. (2010). A Method for Group Formation Using Genetic Algorithm. International Journal on Computer Science and Engineering (IJCSE), 2(9), 3060-3064.

[19] Hans, S., and Chakraverty, S. A Novel Evaluation Scheme using Formal Concept Analysis. International Journal of Applied Engineering Research, Vol 11(1), pp. 439-443.

AUTHOR BIOGRAPHY

Sushma Hans has done Masters in Technology from Jawaharlal Nehru University, Delhi and currently pursuing PhD in Quality Enhancement in E-learning Systems from Netaji Subhas Institute of Technology, New Delhi. She has various researches in the field of dynamic e-learning, e-learning recommender systems and collaborative e-learning in domestic and international conferences and Journals.

Shampa Chakraverty is a Professor and Head of COE Department in Netaji Subhas Institute of Technology, New Delhi. She is an active researcher in various fields such as E-learning, E-governance, semantic web and natural language processing.