

A Skill based Pairing Scheme for Effective Collaborations

Sushma Hans

Computer Engineering Department, Netaji Subhas
Institute of Technology, Sector 3, Dwarka, New Delhi
sushma.nsit@gmail.com

S. Chakraverty

Computer Engineering Department, Netaji Subhas
Institute of Technology, Sector 3, Dwarka, New Delhi
apmahs.nsit@gmail.com

Abstract: We propose an innovative approach of pairing learners in competitive and complementary way based on the learner's learning skills. Students are assessed to determine their inherent learning skills and after their assessment; students are paired up for collaborative learning. We employ two methods: (i) the Stable Marriages Problem and (ii) the recursive Kernighan-Lin partitioning algorithm to generate optimal collaborative pairs. Results reveal that both pairing algorithms generate high quality collaborative pairs. Further, it takes lesser time to discover all possible pairs.

Keywords: *Collaborative Learning, Learning Skills, Kernighan Lin Algorithm, Stable Marriages Problem.*

I. INTRODUCTION

With the widespread and proactive participation of users through various online applications, researchers are now exploring the various ways of collaborating and clustering them based on their profile information [1]. Education research and practices indicate that Collaborative Learning (CL) spurs discussion, facilitates mutual exchange of ideas, and generates a healthy learning atmosphere in which participants can measure up their relative strengths and weaknesses to help each other improve. Academic discussion such as presented in [2] indicates that CL results in higher achievement, better psychological health, social competence and confidence among students. The advent of e-learning as a new, technology-enabled paradigm in education has given a further fillip to CL. People across the globe, cutting across age, time and space constraints, can now cooperate to leverage the benefits of a diverse population covering a wider socio-cultural spectrum.

The core issue that must be addressed in CL is how to form learning groups in the best possible manner so that each student learns significantly more in a group than he or she would have learnt individually. An obvious requirement is that all collaborators must be able to communicate with each other and interact smoothly. In fact, the bulk of research efforts in CL have focused on identifying features that facilitate higher levels of interaction to create homogeneous or heterogeneous groups [3-9]. Many of these schemes form collaborative groups on the basis of the intrinsic qualities of learners such as their personality traits [3], learning styles or cognitive styles [3, 4], interests [5] and learning abilities [6-8].

Mentors distinguish that each learner possesses unique learning skills that impel her to learn and apply knowledge in distinctive ways [9]. However, students do learn new skills and polish existing skills with interaction, guidance, and practice. Indeed, there is evidence of a shift in research direction towards skills analysis and skill-based grouping [10-12]. In [10, 11], the authors group learners based on their programming skills and show that skills-based grouping develops learners' confidence and their retention power. The research work in [12] forms evenly skilled heterogeneous groups in which different skills such as technical writing, engineering, psychology etc. are distributed evenly in a group.

In this paper, we create fruitful collaborations by clubbing together students based on their Learning Skills (LS) repertoire and their skills gap. The primary pedagogical goal of our study is to improve the students' caliber through the mutual transfer of skills. We propose competitive and complementary collaborative pairs among students using (i) the Stable Marriages Problem (SMP) algorithm and (ii) the recursive Kernighan-Lin (KLin) partitioning algorithm.

The rest of the paper is organized in the following manner. Section 2 expounds upon our proposed scheme for competitive and complementary CL. In section 3, we discuss experimental results and we conclude our work in Section 4.

II. LEARNING SKILLS BASED COLLABORATIVE LEARNING SCHEME

The proposed system is designed to form learning skills based competitive and complementary collaborative pairs of co-learners regularly as a course proceeds. The system initially conducts a general pre-test for all registered learners to get an assessment of their inherent learning skills such as their analytical skill, mental skill, numerical skill or linguistic skill and generates initial collaborative pairs accordingly. Once the course is launched, regular examinations are conducted to gauge the performance of students on concepts imparted. The system dynamically re-groups students according to their most recent skills assessment.

A) Forming Collaborative Pairs

Let us consider a set of n students to be organized into $n/2$ collaborative pairs. The system forms competitive and complementary pairs in three steps:

- (i) Create an $n \times n$ matrix of pairing weights by following rules that assign weights to different kinds of pairings.
- (ii) Form optimal collaborative pairs by invoking SMP or recursive KLin pairing algorithms.
- (iii) Calculate the overall quality of collaboration.

We now explain these three steps for competitive and complementary pairing.

• Competitive Pairing

(i) *Pairing Rules*: Competitive collaboration is based on the premise that students whose performance levels in a particular learning skill are comparable join together in a healthy competitive spirit to pit themselves against each other and improve. Therefore, students who have similar performance levels in various skills are paired together. Table 1 gives the rules for pairing students on the basis of a given LS_x . Rules 1 to 3 favor competitive pairing and are therefore assigned a higher pairing weight $W(x)=1$. For example, pairing rule 1 pairs two students who are both *weak* in LS_x . This rule is assigned a high pairing weight $W(x) = 1$. Rules 4 to 6 are assigned $W(x) = 0$ as the collaborators have dissimilar performance.

Table 1: Competitive Pairing Rules for a given LS_x

Rule No.	Student-pair Performance Combination	Pairing Weight $W(x)$
1	Weak-Weak	1
2	Good-Good	1
3	Average-Average	1
4	Average-Good	0
5	Average-Weak	0
6	Good-Weak	0

With a total of n_{LS} LSs, the overall Preference Weight $PW_{i,j}$ between s_i and s_j is given by the sum of the pairing weights for all LSs:

$$PW_{i,j} = \sum_{k=1}^{LS} W_{i,j}(k) \quad (1)$$

(ii) *Optimal collaborative pairs*: Given a class of n students, the system forms an $n \times n$ upper triangular matrix $PW[n][n]$ of preference weights for all possible pairs between them. This matrix is accepted as input by one of the two pairing algorithms, namely, SMP or recursive KLin to yield an optimal set P of $n/2$ collaborative pairs. The functioning of SMP and recursive KLin is described in pseudo code 1 and 2.

(iii) *Quality of competitive collaboration*: Assume an optimal set P of $n/2$ competitively collaborating pairs of students. The competitive collaboration quality $QC(i,j)$ of a

pair of students (s_i, s_j) is defined as their Preference Weight per LS, i.e.

$$QC(i,j) = \frac{PW_{i,j}}{LS} \quad (2)$$

In order to differentiate between competitive and complementary pairing, we denote the overall pairing quality for the former as QC_α and for the later as QC_β . The overall quality of competitive pairing QC_α is given by the average of the quality of competitive pairing of each pair in P . Thus, with $n/2$ pairs we have:

$$QC_\alpha = \frac{\sum_{(i,j) \in P} QC(i,j)}{n/2} \quad (3)$$

• Complementary Pairing

(i) *Pairing Rules*: Complementary pairs are geared to fulfill the skills gap. If a student is *good* at one performance parameter say LS_x but lacks in another say LS_y , he or she can be paired with another student with complementary learning skill that is, one who is lacking in LS_x but good at LS_y . In this way both get an opportunity to benefit from each other and grow. This pairing scheme allows the transfer of related skills between collaborators so as to enhance their skills to solve complex problems requiring multiple skills.

Table 2 gives the rules for this pairing scheme. The scheme applies a distance measure d_x that measures the difference in the three performance levels: *good*, *average* and *weak*, of the two collaborators for a given LS_x . The distance $d_x=1$ for (*good*, *average*) as well as (*average*, *weak*) combinations and $d_x= 2$ for the (*good*,*weak*) combination. The value of d_x for (*average*,*average*), (*good*,*good*) and (*weak*,*weak*) combinations are all *zero*. The pairing weight for a rule is computed by adding up the values of d_x and d_y for LS_x and LS_y respectively.

From Table 2, we can see that the first rule presents the maximum contrast between performance levels. For this rule, the values of d_x and d_y are both 2. 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. Similarly, the second rule has $d_x=2$ and $d_y=1$, giving $W(x,y)=3$.

Table 2: Complementary Pairing Rules for a given pair LS_x and LS_y

Rule No.	Student-pair Performance Combination In LS_x and LS_y	Pairing weight $W(x,y)$
1	LS_x : Good-Weak LS_y : Weak-Good	4
2	LS_x : Good-Weak LS_y : Average-Good	3
3	LS_x : Good-Weak LS_y : Weak-Average	3
4	LS_x : Good-Average LS_y : Weak-Good	3
5	LS_x : Average-Weak LS_y : Weak-Good	3

6	LS _x : Good-Weak LS _y : Average-Average	2
7	LS _x : Good-Average LS _y : Average-Good	2
8	LS _x : Average-Weak LS _y : Weak-Average	2
9	LS _x : Average-Average LS _y : Weak-Good	2
10	LS _x : Good-Good or Weak-Weak Any other combination	0

When students pair up in complementary mode, they are free to learn from each other on any pair of skills in a mutually beneficial manner. With LS total LSs, we get a maximum of $({}^L_2C)$ distinct pairs of learning skills. The appropriate rule that is triggered by a given LS pair, LS_x and LS_y is found by matching the performance levels of s_i and s_j in LS_x and LS_y with the corresponding rule in Table 2. The Preference Weight PW between s_i and s_j is their cumulative pairing weights of all LSs:

$$PW_{i,j} = \sum_{k=1}^{LS-1} \sum_{l=k+1}^{LS} W_{i,j}(k, l) \quad (4)$$

(ii) *Optimal pairs*: As in the case of competitive pairing, the system forms an $n \times n$ upper triangular matrix $PW[n][n]$ of overall pairing weights between all n students which is input by one of the two algorithms, SMP or recursive KLin to yield an optimal set P of $n/2$ collaborative pairs (refer pseudocode 1 and 2).

(iii) *Quality of complementary collaboration*: With a set P of $n/2$ optimally paired students, the quality $QC(i, j)$ of complementary collaboration between a pair of students (s_i, s_j) is defined as their average pairing weight per LS combination. With (LS_{C_2}) Possible LS combination, we have:

$$QC(i, j) = \frac{PW_{i,j}}{(LS_{C_2})} \quad (5)$$

The overall quality of complementary pairing QC_{β} is given by the average quality of complementary pairing considering all pairs in P . With $n/2$ such pairs, we have:

$$QC_{\beta} = \frac{\sum_{i,j} QC(i, j)}{n/2} \quad (6)$$

• Pairing Algorithms

(i) *SMP Approach*: The Stable Marriages Problem (SMP) determines a stable matching between members of two groups. In [13], the authors have explained the SMP algorithm in detail. The readers can refer these papers for complete understanding of the algorithm. The pseudo-code 1 given below describes how SMP is applied to yield collaborative pairs.

SMP_Pairing (.)

Input: Set of Students S ; Preference Lists $PW[n][n]$

Output: Collaborative pairs

Begin

1. Replicate $PW[n][n]$ as $PW_A[n][n]$ and $PW_B[n][n]$, S as A and B .
2. Initialize all $s_i \in A$ and $s_j \in B$ as *unpaired*
3. **While** \exists *unpaired* student $s_i \in S$ {
4. **If** $((s_i \neq s_j) \text{ and } (s_j \text{ is the most preferred student for pairing with } s_i))$ **then**
5. **If** $(s_j \text{ is unpaired})$ **then**
6. pair (s_i, s_j)
7. **Else If** $(s_j \text{ is paired with } s_k)$ **then**
8. **If** $(s_j \text{ has higher preference for } s_i \text{ than for } s_k)$ **then** {
9. pair (s_i, s_j)
10. Set s_k as *unpaired* }
11. } //end while

End

Pseudo-code 1: Forming collaborative pairs using SMP approach

The traditional SMP starts with two groups of n men and n women, each having a list of ordered preferences for the other group. In our case, however, a single set of students register for a course and collaborations need to be forged within that group. A matrix of Preference Weights $PW[n][n]$ between all pairs of students is input to the system. This is replicated to represent the two preference lists (line 1). After initializing all students of both groups as *unpaired*, SMP initiates an iterative process of pairing (line 2). Taking each unpaired student s_i in turn, it finds the most preferred yet-unpaired student s_j for s_i (line 3-4). If s_j is unpaired then s_i and s_j are paired together (line 5-6). If s_j is already paired with another student s_k , the algorithm checks whether s_j has more preference for s_i than for s_k (lines 7-8). If s_i has more preference, s_i and s_j are paired together and s_k is marked as *unpaired* (lines 9-10).

(ii) *Recursive KLin Approach*: The KLin algorithm is a well known graph partitioning algorithm [14]. It follows a greedy approach to partition a graph of $2n$ vertices into two disjoint arbitrary subsets X and Y of n vertices each, in an optimal way such that the sum of costs of the external edges between nodes in X and Y is minimized.

The basic partitioning logic can be applied repeatedly, resulting in smaller groups. Starting with n students, for l partitioning steps, KLin yields $\left\lceil \frac{n}{2^l} \right\rceil$ groups of roughly the same size. The advantage of recursive KLin is that the process of partitioning can be stopped at any step to get collaborative groups of different sizes. The group size is set to 2 for collaborative pairing.

Pseudocode 2 given below describes the recursive clean approach to generate collaborative pairs. The problem of bipartitioning a set of n students is modeled as a weighted graph $\tilde{G}(V, E)$ whose vertices represent students S and edges denote their mutual Preferences Weights $PW_S[n][n]$ (line 1). The weight of the edge $(E_{i,j})$ from

student vertex v_i to v_j is computed by adding the preference weight PW_{ij} of v_i for v_j and preference weight PW_{ji} of student vertex v_j to v_i :

$$E_{i,j} = PW_{i,j} + PW_{j,i} \quad (7)$$

The basic KLin algorithm is invoked to partition the graph (line 2) and the partition size is halved (line 3). If the size of each partition is greater than the preset group size, then the KLin_pairing algorithm is recursively applied to both partitions (lines 5,6). Otherwise, the groups of the prescribed size have been achieved and are output (line 7). The time complexity of KLin to form a single optimal partitioning is $O(n^2 \log n)$ [14]. The recurrence relation for the proposed KLin_pairing(.) algorithm is given by:

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2 \log n \quad (8)$$

By applying the third rule of master's theorem, we get the time complexity of eq. 8 as $\theta(n^2 \log n)$.

KLin_pairing (.)

Input: No_of_Students n , $PW_S[n][n]$, set of students S , Group_size $f=2$,

Output: Collaborative pairs

1. Create $\tilde{G}(V, E)$, where $|V|=|S|$ and assign edge weights using $PW_S[n][n]$
2. Call **KLIn** ($\tilde{G}(V, E)$) to generate A and B
3. Set $n=n/2$
4. **If** ($n > f$) {
5. Call **KLIn_pairing**($n, PW_A[n][n], A, f$);
6. Call **KLIn_pairing**($n, PW_B[n][n], B, f$); }
7. Output all partitions obtained as collaborative pairs.

Pseudo-code 2: Recursive KLin to generate collaborative pairs

III. EXPERIMENTAL RESULTS

The proposed framework was coded in C using Dev C++ version 5.9.2. We performed our experiments on an Intel core i5 machine with 2.40 GHz processor running Windows 7. We experimented with student group sizes of 32, 64 and 128 students.

A) Competitive Pairing

The first experiment aims to check the quality of the competitive pairs generated using the SMP and the KLin.

We conducted the experiment with a group size set to 32 students. Table 3 lists the set of competitive pairs. Part A shows the common competitive pairs generated by both SMP and KLin. Part B segregates the pairs that were different for SMP and KLin.

1. There are 5 high quality competitive pairs that are commonly generated by both SMP and KLin with QC_a varying from 0.72 to 0.89. There are 11 competitive pairs that are distinct for SMP and KLin.
2. We find that the average quality of the common competitive pairs is 0.82 which is 12.3% higher than

QC_a of distinct SMP pairs (average $QC_a = 0.73$) and 18.8 % higher than QC_a of distinct KLin pairs (average $QC_a = 0.69$). This shows that most of the high quality pairs are common for both the algorithms. The same observations are identified with set of 64 and 128 students as given in Table 4. Hence, we can use either method to generate high quality pairs.

3. The average QC_a for SMP is 0.76 and KLin is 0.73. This intricate that given a dataset with random distribution of students' performance, SMP produces more competitive pairs in the higher range, whereas KLin produces more competitive pairs in the middle range. It indicates that SMP generate more high quality pairs as compared to KLin.

Table 3: Same and distinct Competitive pairs generated through SMP and KLin

Part A			Part B				
N	Pairs	QC_a	N	SMP Pairs	QC_a	KLIn Pairs	QC_a
1	$S_0 - S_{24}$	0.83	1	$S_1 - S_{18}$	0.72	$S_1 - S_{27}$	0.83
2	$S_3 - S_{21}$	0.72	2	$S_2 - S_{13}$	0.94	$S_2 - S_{17}$	0.72
3	$S_6 - S_{15}$	0.89	3	$S_4 - S_{11}$	0.94	$S_4 - S_5$	0.89
4	$S_{10} - S_{25}$	0.78	4	$S_5 - S_{17}$	0.77	$S_7 - S_{28}$	0.83
5	$S_{16} - S_{22}$	0.89	5	$S_7 - S_{12}$	0.72	$S_8 - S_{31}$	0.67
			6	$S_8 - S_{14}$	0.72	$S_9 - S_{12}$	0.78
			7	$S_9 - S_{28}$	0.89	$S_{11} - S_{13}$	0.78
			8	$S_{19} - S_{26}$	0.5	$S_{19} - S_{29}$	0.11
			9	$S_{20} - S_{27}$	0.89	$S_{14} - S_{20}$	0.78
			10	$S_{23} - S_{31}$	0.78	$S_{18} - S_{30}$	0.72
			11	$S_{29} - S_{30}$	0.17	$S_{23} - S_{26}$	0.5
Average		0.82	Average		0.73	Average	0.69

Table 4: Average QC_a for different student group sizes

Student Group Size	Average QC_a (common pairs)	Average QC_a (distinct SMP pairs)	Average QC_a (distinct KLin pairs)	Average Overall QC_a of SMP pairs	Overall QC_a of KLin pairs
32	0.82	0.73	0.69	0.76	0.73
64	0.81	0.79	0.80	0.80	0.81

128	0.87	0.83	0.82	0.85	0.84
-----	------	------	------	------	------

Figure 1 shows the frequency distribution of the competitive pairs along varying quality ranges. This shows that in case of competitive pairs generated by SMP and KLin, maximum pairs have QC_{α} in range from 0.8 to 1.0. If we see, on an average, SMP has generated more pairs in the highest quality range from all the group sizes.

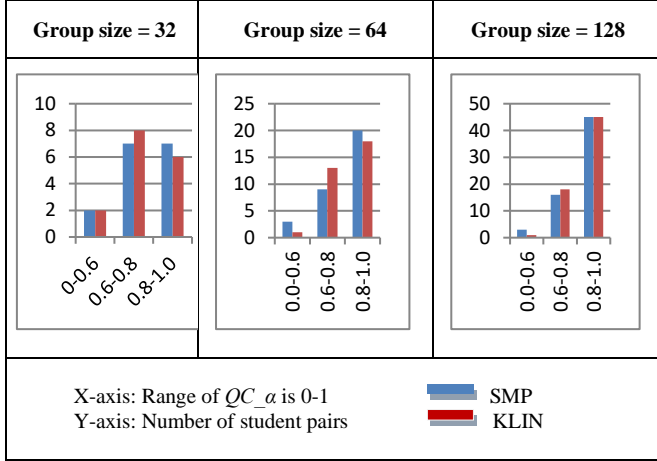


Fig. 1: Number of student pairs v/s Quality range QC_{α} for SMP and KLin

B) Complementary Pairing

The second experiment is performed to quantify the nature of complementary pairs generated by both the algorithms. With a group of 32 students, part A of Table 5 lists the complementary pairs that are common for SMP and KLin. Part B recorded distinct pairs generated by the algorithms.

1. In this scenario, SMP and KLin generated eight complementary pairs that are common. Their QC_{β} varies from 10.4 to 68.9. Eight complementary pairs are distinct for SMP and KLin.
2. The average quality of the common complementary pairs is 0.76 which is 46.1% higher than QC_{β} of distinct SMP pairs (average QC_{β} = 0.52) and 76.7 % higher than QC_{β} of distinct KLin pairs (average QC_{β} = 0.43). This demonstrates that most of the high quality pairs are common for both the algorithms. The same pattern of various QC_{β} values are observed with set of 64 and 128 students as given in Table 6. This reassures the idea of using either of the method to generate high quality pairs.
3. The overall average QC_{β} for SMP is 0.64 and KLin is 0.59. These findings show that SMP produces higher quality complementary pairs as contrasted with KLin.

Table 5: Same and distinct Complementary pairs generated through SMP and KLin

Part A			Part B				
N	Pairs	QC_{β}	N	SMP Pairs	QC_{β}	KLin Pairs	QC_{β}
1	$S_2 - S_{27}$	0.60	1	$S_0 - S_{12}$	0.10	$S_0 - S_4$	0.29
2	$S_5 - S_{25}$	0.91	2	$S_1 - S_3$	1.29	$S_1 - S_{17}$	0.81
3	$S_7 - S_{19}$	0.88	3	$S_4 - S_{30}$	0.64	$S_3 - S_{30}$	1.11
4	$S_8 - S_{11}$	0.62	4	$S_6 - S_{15}$	0.15	$S_6 - S_{29}$	0.15
5	$S_9 - S_{31}$	0.21	5	$S_{13} - S_{24}$	0.18	$S_{15} - S_{24}$	0.18
6	$S_{10} - S_{21}$	1.38	6	$S_{14} - S_{22}$	0.82	$S_{14} - S_{26}$	0.19
7	$S_{16} - S_{20}$	0.79	7	$S_{17} - S_{18}$	0.66	$S_{18} - S_{22}$	0.54
8	$S_{23} - S_{28}$	0.71	8	$S_{26} - S_{29}$	0.29	$S_{12} - S_{13}$	0.16
Average		0.76	Average		0.52	Average	0.43

Table 6: Average QC_{β} for different student group sizes

Student Group Size	Average QC_{β} (common pairs)	Average QC_{β} (distinct SMP pairs)	Average QC_{β} (distinct KLin pairs)	Overall Average QC_{β} of SMP pairs	Overall Average QC_{β} of KLin pairs
32	0.76	0.52	0.43	0.64	0.59
64	0.89	0.54	0.5	0.69	0.66
128	0.88	0.68	0.68	0.72	0.71

The frequency distribution of the complementary pairs along varying quality ranges is shown in Figure 2. This shows that complementary pairs generated by SMP and KLin mostly lies in the QC_{β} range from 0.4 to 1.2.

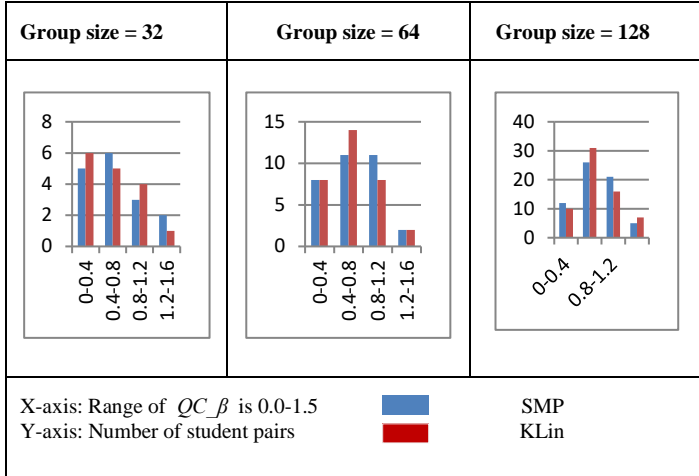


Fig. 2: Number of student pairs v/s Quality range $QC \beta$ for SMP and KLin

C) Execution of SMP and KLin

In order to measure the time taken for execution of both algorithms, we increased the group size in powers of 2. Figure 3 shows the execution times taken by the SMP and KLin based pairing methods with increment in group size. We find that the time taken by KLin algorithm is always higher as compared with SMP.

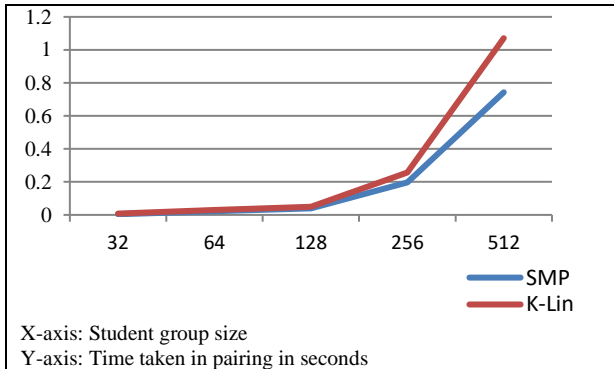


Fig. 3: Time taken in pairing v/s group size

IV. CONCLUSION

In this paper, we demonstrated how the SMP matching algorithm and recursive KLin partitioning algorithm can be utilized to create high quality competitive or complementary collaborative pairs by applying rules to match students based on their achievement in various LSs. We defined quality metrics to measure the quality of pair-wise collaboration. Using randomly generated dataset, we experimentally showed that both approaches yielded high quality pairs though on an average, SMP generates slightly

higher quality pairs. It also consumes lesser execution time, scaling well with problem size.

For future work, we will try to identify the impact of this new way of analyzing a subject on the basis of the required mental skills for that subject on human psychology and their learning. In addition, we will try to identify a realistic method to determine skill information of students that can give us more insight about the learner's mental skills to solve a problem.

REFERENCES

- [1] Chakraverty, S., Rani, G., and Bardhan, A., 2013. Profiling E-Governance Users using Biclustering. *International Journal of Engineering and Innovative Technology (IJEIT)*, 3(1), 521-528.
- [2] Laal, M., and Ghodsi, S. M., 2012. Benefits of Collaborative Learning. *Procedia- Social and Behavioral Sciences* 31, Elsevier, 486-490.
- [3] Jin, D., Z. Qinghua, Z., Jiao, D., and 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.
- [4] Ghorbani, F., and Montazer, G.A., 2012. Learners Grouping Improvement in E-learning Environment using Fuzzy Inspired PSO Method. *6th National and 3rd International conference of e-Learning and e-Teaching ICELET2012*, Tehran, 65-70.
- [5] Kardan, A.; Sadeghi, H., 2014. Modeling the Learner Group Formation Problem in Computer-Supported Collaborative Learning using Mathematical Programming. *5th International Conference on e-Learning and e-Teaching (ICeLeT)*, IEEE, 1-5.
- [6] Betts, J. R., and Shkolnik, J. K., 2000. The Effects of Ability Grouping on Student Achievement and Resource Allocation in Secondary Schools. *Economics of Education Review* 19, Elsevier, 1-15.
- [7] Adodo, S. O., and Agbayewa, J. O., 2011. Effect of Homogeneous and Heterogeneous Ability Grouping Class Teaching on Student's Interest, Attitude, and Achievement in Integrated Science. *International Journal of Psychology and Counselling*, 3(3), 48-54.
- [8] Sushma Hans, S. Chakraverty, Ridam Batra, Radhika Subhangi, 2016. Ability Based Collaborative Pairing and Grouping Among Learners. *International Journal of Engineering and Innovative Technology*, 5(8).
- [9] Chen, J-Q, Moran, S., and Gardner, H.(Eds) 2009. Multiple intelligences around the world. *John Wiley and Sons*.
- [10] Zhamri C. A., and Yasin A., 2010. A Method for Group Formation Using Genetic Algorithm. *International Journal on Computer Science and Engineering (IJCSSE)*, 2(9), 3060-3064.
- [11] McDowell, C., Werner, L., Bullock, H. E., and Fernald, J. 2006. Pair Programming Improves Student Retention, Confidence, and Program Quality. *Magazine Communications of the ACM, NY, USA*, 49(8), 90-95.
- [12] Hübscher, R. 2010. Assigning Students to Groups Using General and Context-Specific Criteria. *IEEE Transactions on Learning Technologies*, 3(3), 178-189.
- [13] Gale, D., and Shapley, L., 1962. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69 (1), 9-15.
- [14] Kernighan, B. W., and Lin, S., 1970. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal*(49), 291-307.