Intelligent Group Formation in Computer Supported Collaborative Learning Scripts

Master Thesis

Ishari Amarasinghe

Master in Intelligent Interactive Systems
September 2016

Supervisors
Professor Davinia Hernández-Leo
Professor Anders Jonsson

Department of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona
Abstract

Well-structured collaborative learning groups scripted based on collaborative learning flow patterns (CLFP) pave the way to successful outcomes in collaborative learning environments. Formulation of such learner groups based on instructor defined criteria promises maximum potential performance of participating students. However, forming student groups manually based on multiple criteria often fails due to its complexity and time limitations. Moreover learner drop outs during collaborative learning session often creates hardships for instructors since it demands learner group reformulations on the fly. Considering all these difficulties associated with collaborative learning sessions, an intelligent assistance towards learner group formulation based on instructor defined criteria, while adhering to Jigsaw CLFP was presented during the thesis. Learner group formulation was modeled as a Constraint Optimization Problem and preliminary tests were carried out using simulated and real world data-sets. The results of the tests have indicated that the proposed approach formulates student groups while satisfying team formation criteria. Further it could be successfully utilized when reformulating learner groups while minimizing changes to the existing group structures. Finally conclusions and recommendations for further research have been provided.
Acknowledgement

I would like to express sincere gratitude to my advisors Prof. Anders Jonsson and Prof. Davinia Hernández-Leo for the continuous support extended towards my Master Thesis, for their patience, motivation, and immense knowledge provided. I acknowledge my dearest parents for showing faith in me and providing me liberty to choose what I desired. I salute them for their love, care and sacrifices made to shape my life. I owe a big thank to my husband, for his continuous support and understanding that made the completion of the thesis possible. Finally, all the members of GTI (Interactive Technologies Group) Group at Universitat Pompeu Fabra reminded here with gratefulness for their kind assistance and support provided to make this effort a success.
Contents

List of Figures v
List of Tables vii

1 INTRODUCTION 1
1.1 Motivation ................................................. 1
1.2 Formulation of the Research Problem ....................... 2
1.2.1 Collaborative Learning Flow Patterns & Constraints ... 2
1.3 Research Objectives ....................................... 4
1.4 Research Methodology .................................... 5

2 LITERATURE REVIEW 7
2.1 Limitations of existing methods ............................. 9

3 PROPOSED ALGORITHM 11
3.1 Problem Modeling ......................................... 12
3.2 Implementing Flexibility ................................... 14

4 TESTING & EVALUATION 16
4.1 Test Settings .............................................. 16
4.2 Group Formation Design Analysis .......................... 17
4.2.1 Tests conducted using simulated dataset ................ 17
4.2.2 Tests conducted using real world dataset ............... 25
4.3 Regrouping Design Analysis ................................ 31
4.4 Performance Analysis ..................................... 33
4.4.1 Discussion ............................................. 34

5 PERFORMANCE IMPROVEMENTS 35
5.1 Linearizing Quadratic terms ............................... 35
5.2 Obtaining results as approximations ....................... 36
5.2.1 Discussion ............................................. 37

6 CONCLUSIONS & FUTURE WORK 39

Appendix A ARTIFICIAL DATASET 43
Appendix B SOURCE CODES 44
List of Figures

1.1 Jigsaw CLFP-Learning Flow Diagram .......................... 3
4.1 System Design .................................................. 16
5.1 Execution time for expert group formation .................... 38
5.2 Execution time for Jigsaw group formation .................... 38
List of Tables

2.1 A Summary of related literature ........................................ 9
3.1 Intrinsic/Hard constraints applied on Jigsaw CLFP ................. 11
3.2 Extrinsic/Soft Constraints applicable for Jigsaw CLFP ............ 12
4.1 Description of simulated dataset ...................................... 18
4.2 Input Parameters & Execution Time - Test 01 (Simulated Dataset) 18
4.3 Test 01 Results: Expert Groups (Simulated Dataset) ............. 19
4.4 Test 01 Results: Jigsaw Groups (Simulated Dataset) .............. 19
4.5 Input Parameters & Execution Time - Test 02 (Simulated Dataset) 20
4.6 Test 02 Results: Expert Groups (Simulated Dataset) ............. 20
4.7 Test 02 Results: Jigsaw Groups (Simulated Dataset) .............. 20
4.8 Input Parameters & Execution Time - Test 03 (Simulated Dataset) 21
4.9 Test 03 Results: Expert Groups (Simulated Dataset) ............. 21
4.10 Test 03 Results: Jigsaw Groups (Simulated Dataset) ............ 21
4.11 Input Parameters & Execution Time - Test 04 (Simulated Dataset) 22
4.12 Test 04 Results: Expert Groups (Simulated Dataset) ............. 22
4.13 Test 04 Results: Jigsaw Groups (Simulated Dataset) .............. 22
4.14 Input Parameters & Execution Time - Test 05 (Simulated Dataset) 23
4.15 Test 05 Results: Expert Groups (Simulated Dataset) ............. 23
4.16 Test 05 Results: Jigsaw Groups (Simulated Dataset) ............ 24
4.17 Description of real world dataset ................................... 25
4.18 Input Parameters & Execution Time - Test 01 (Real World Dataset) 26
4.19 Test 01 Results: Expert Groups (Real World Dataset) ............ 26
4.20 Test 01 Results: Jigsaw Groups (Real World Dataset) ............ 26
4.21 Input Parameters & Execution Time - Test 02 (Real World Dataset) 27
4.22 Test 02 Results: Expert Groups (Real World Dataset) ............ 27
4.23 Test 02 Results: Jigsaw Groups (Real World Dataset) ............ 27
4.24 Input Parameters & Execution Time - Test 03 (Real World Dataset) 28
4.25 Test 03 Results: Expert Groups (Real World Dataset) ............ 28
4.26 Test 03 Results: Jigsaw Groups (Real World Dataset) ............ 28
4.27 Input Parameters & Execution Time - Test 04 (Real World Dataset) 29
4.28 Test 04 Results: Expert Groups (Real World Dataset) ............ 29
4.29 Test 04 Results: Jigsaw Groups (Real World Dataset) ............ 29
4.30 Input Parameters & Execution Time - Test 05 (Real World Dataset) 30
4.31 Test 05 Results: Expert Groups (Real World Dataset) ............ 30
4.32 Test 05 Results: Jigsaw Groups (Real World Dataset) ............ 30
4.33 Task allocation before regrouping .................................. 31
4.34 Task allocation after regrouping .................................... 32
4.35 Jigsaw groups before regrouping ................................... 32
4.36 Jigsaw groups after regrouping ...................................... 33
4.37 Execution time in seconds (Real World Dataset) . . . . . . . . . . 34
5.1 Approximation time in seconds (Real World Dataset) . . . . . . . 37
A.1 Simulated Dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
Chapter 1

INTRODUCTION

Over the past few years research conducted in different disciplines [2, 6] have confirmed that active collaborative learning is an effective means of instruction which can be utilized in both traditional and online educational environments [11] that would result in long-term effects in education [15]. Learning within groups expose students to peers enabling them to learn via interactions among each other in addition to learning from instructor [14, 15]. Further, group work conducted under proper conditions provides an opportunity for students to clarify and refine their understanding of concepts through discussions and rehearsals with peers [5].

In 1990 O’Malley & Scanlon [11] first introduced the term *Computer Supported Collaborative Learning (CSCL)*. Since then it has emerged as a dynamic, interdisciplinary and an international field of research focused on how technology can provide explicit or implicit support to facilitate the sharing and creation of knowledge through peer interactions and group learning processes [7, 11]. However, learning via interactions does not occur in any situation [7]. Careful consideration on the design of collaboration is the key to achieving desired learning goals.

1.1 Motivation

Many studies have pointed out that formation of well-structured collaborative learning groups as the starting point of CSCL [7, 11]. As mentioned by [3, 14] students can be either allocated to groups randomly self-select each other or be appointed to a group by the instructor based on specific criteria. Problems associated with the random grouping includes the tendency to avoid heterogeneous groups, choosing of groups based on friendship rather than on functional or learning reasons and the trouble of finding groups for some students resulting *orphans* [15]. Further some groups might not achieve learning goals because group members do not have the required skills for learning and some groups may easily accomplish learning goals without significant learning because every group member has skills required to achieve learning goals [12].

According to [8] groups formed without careful consideration often causes problems such as disproportionate participation of individuals, demotivation and resistance to group work in future activities [7]. On the other hand, instructor could select grouping criteria based on different factors related to student profiles [7, 11] such as cultural background, knowledge, interests, abilities, skills, learning styles and roles which can be then expressed as a set of conditions typically refers to as constraints [14] when forming groups. Different grouping structures could foster
different types of interactions and outcomes [5].

Yet, due to the possibility of using several learner’s characteristics and combine them in different ways, forming groups manually could become a difficult and time-consuming process [3, 4, 7, 14]. Hence researchers have been investigating several techniques to automate the process of group formation via Computer Supported Group Formation (CSGF) [3, 14] which provides computational support to complete group formation task successfully.

1.2 Formulation of the Research Problem

One major approach of forming student groups is based on homogeneity and heterogeneity information available on student profiles. Heterogeneous groups provide opportunities for students to learn how to interact with classmates who have different learning profiles [19]. In Homogeneous groups, members are similar to one another with respect to a particular criterion [11]. However, different authors have pointed out that in heterogeneous groups extreme differences among group members could impair cooperation [19] while homogeneous groups would result in high level of satisfaction which would encourage learners to engage in more interactions and discussions [11]. However not only homogeneity and heterogeneity associated with student profiles but also grouping constraints defined by the instructor or constraints inherited from the CLFP also become important to foster effective interactions within groups.

1.2.1 Collaborative Learning Flow Patterns & Constraints

According to authors [10] a CLFP can be understood as a way of describing a collaborative learning technique. These CLFPs were derived from practice [10] rather than from general learning theories. Thus, these methods become important to collaborative learning practitioners since these methods pre-structure collaboration, ensuring productive interactions among learners.

CLFPs can also be described as patterns which depict commonly used techniques which are repetitively used by practitioners when structuring the flow of types of learning activities involved in collaborative learning scenarios [10]. Examples of well-known CLFPs include Jigsaw CLFP and Pyramid CLFP. Each of these CLFP inherits a set of conditions commonly known as constraints which shape up the desired collaboration [13]. According to [8] these constraints can be divided into two major categories known as intrinsic and extrinsic constraints. Intrinsic constraints are bound to the core mechanisms of the pedagogical design while extrinsic constraints are induced by different issues such as contextual factors or arbitrary decisions [8]. As mentioned above heterogeneity and homogeneity details associated with learner profiles and technological tools supporting learning activities could be modeled as extrinsic constraints [13].
Moreover, as described by [13] intrinsic constraints are critical as it provides the fundamentals. If these constraints could not be satisfied it would result in violating the underlying pedagogy; hence these are also known as *Hard* constraints. Extrinsic constraints are complementary that are preferred to be implied as *Soft* constraints which add more value towards collaboration which result in meaningful interactions [13].

Apart from constraints, most of the CLFPs have a unique definition for grouping within phases [13]. A group requires having specific size including a minimum or sometimes a maximum number of participants. Hence, number of possible groups are also defined with the pedagogical definition of CLFPs along with related group formation policies [13].

**Jigsaw CLFP**

In the context of learning science, *Jigsaw* CLFP\(^1\) is often adopted when several small groups of students are facing to study a lot of information for the resolution of the same problem. This CLFP consists of three major phases known as task allocation, formation of expert groups and formation of Jigsaw groups as described in Figure 1.1. The problem which is trying to be solved during the collaboration can be easily divided into sections or independent tasks.

As it can be seen in Figure 1.1, during **Phase 01** students are assigned to specific tasks and in **Phase 02** students who are working on the same task come together to work collaboratively on the same task. Finally, in **Phase 03** students who have studied different tasks are grouped together forming Jigsaw groups so

\(^{1}\)http://www.gsic.uva.es/daviniahl/clfp/jigsaw-en/
that each Jigsaw group consists students from each task. During the thesis, Phase 01 and Phase 02 are considered as a single phase (Phase 01-Expert Phase) in which both task allocation and expert group formation occur simultaneously forming temporary Expert Groups to exchange ideas about their common tasks. Finally, participants grouped in Jigsaw Groups in Phase 02 to share knowledge on different tasks to solve the global problem [13].

Based on the way, that the collaborative learning is structured during Jigsaw CLFP, it promotes the feeling among team members that they need each other to succeed (positive interdependence), and also it encourages discussions among peers which result in constructing knowledge while ensuring that every learner contribute their fair share (individual accountability) [10].

1.3 Research Objectives

As described in section 1.2.1 the type of constraints and the number of constraints required to be fulfilled during group formation could vary based on the CLFP adhered. In general with the number of constraints used to form groups increase, the task of grouping learners manually becomes difficult and time-consuming [1, 14]. Especially in online learning environments like Massive Open Online Courses (MOOCs) this situation can become even worse, where large numbers of students get registered for a particular course. Yet, due to the complexity of this task it often requires computational support to be completed successfully. Hence, the major objective addressed during the thesis was how to formulate a new algorithm which supports learner group formation based on both intrinsic and extrinsic constraints defined in Jigsaw CLFP.

However, an algorithm which supports only group formation based on constraints specified by the CLFP does not completely fulfill the requirements of the problem being addressed. As pointed out by [8] a collaborative learning script must be flexible, hence it could be easily adaptable by students and/or the teachers. In terms of implementation, computerized scripts, as workflows, should be able to relieve the teacher from back office tasks but should not break the subtle equilibrium between freedom and constraints by increasing the script rigidity beyond pedagogical usefulness [8]. During the thesis flexibility of computerized script was suggested to be implemented during session set up and run time [8].

- Session set up
  - Defining Groups: A teacher can define the ideal group size based on the targeted interactions. Algorithm should formulate student groups accordingly. However, it should also accommodate formulation of groups by varying group sizes if required while adapting to constraints associated with the underlying CLFP.

- Run time
Changing Groups: Flexibility requirement associated with run time parameter occurs due to the changing nature of learning environments. For instance, group structure changes might be required during run time of the collaborative learning session due to a member drop out and/or a new member addition to the course in the middle of a learning activity.

According to [8] flexibility requirements described above may be impossible due to computational reasons, i.e. because the developer did not anticipate that some data could be missing. Hence, during the thesis such requirements were given priority. Acceptability of these modifications was determined to be evaluated based on the intrinsic and extrinsic constraint satisfaction.

Another objective considered during the thesis was the scalability of the suggested algorithmic approach. Scalability of the suggested approach becomes useful when it is being used to formulate student groups in MOOCs where a large number of students required to be allocated to groups while adhering to grouping constraints. Further, group formation needs to finish execution within acceptable time frames. As mentioned by authors [4] for a class of 30 students team creation manually can take up to an hour. Hence, scalability of the suggested approach was also given attention.

In summary, the main objective of this thesis was to propose and test an algorithm that could be utilized when forming learner groups based on Jigsaw CLFP while adhering to constraints specified. Further, the suggested algorithm requires being flexible so that modifications required by the instructor when forming student groups could be handled gracefully. In addition, it should scale well, so that it could be applied to a large number of students and group formation results could be obtained within acceptable time frames.

### 1.4 Research Methodology

The methodology used during the research was the Engineering Method. According to authors of [21] engineering method involves the initial development of a solution to test a hypothesis which could be further improved based on test results until no further improvement is required.

During the work presented in this thesis, we had developed an algorithm to support group formation in collaborative learning environments based on Jigsaw CLFP and subsequent tests were carried out to evaluate its performance. The results obtained during tests were used to improve functional and nonfunctional aspects of the algorithm. Both simulated and real world datasets have been used during experiments. Experiments were carried out in a synthetic environment as simulations [21]. These settings helped to validate the outcome of the algorithm since a simulation is often easier, faster and less expensive to run than the full product in the real environment [21]. Following section describes the methodology in steps.
Requirement identification and formulation of the research question

Before implementing the algorithm it became vital to obtain a comprehensive knowledge on CLFPs applied during collaborative learning scenarios which affect the flow of a collaborative learning session. Required knowledge on such aspects and also a clear idea of pedagogical constraints were obtained via reading related literature [8, 10] and also by discussing with the supervisor. Further discussions carried out with the Doctoral students from the GTI (Interactive Technologies Group) Group at Universitat Pompeu Fabra, helped to clarify functional aspects of the algorithm while adhering to pedagogical design of the collaborative learning design.

Background study

Different authors have developed different solutions to assist group formation problem in CSCL environments over time. A comprehensive literature review was carried out in order to identify which methods had been tried out in the field.

Study and benchmark current applications

Existing solutions had been analyzed carefully in order to identify their implementation approaches, grouping criteria, strengths and weaknesses. This helped when justifying why existing solutions cannot be fully utilized to address the research problem.

Algorithm Development

Based on the nature of the research problem (in which a number of soft and hard constraints related to a CLFP required to be satisfied) it was determined that this problem could be expressed as a Constraint Optimization Problem (COP). Intrinsic constraints which were mandatory to satisfy while forming groups were modeled as hard constraints, while extrinsic constraints which were not mandatory to be satisfied but affect the quality of the solution were encoded into the objective function of the COP. Hence optimization of the objective function would maximize the number of soft constraints satisfied. Selection of a proper platform, and a solver to develop and solve COP was given attention.

Testing and Obtaining Feedback

Developed solution was tested using different learning scenarios in order to verify and also to improve the performance and scalability of the proposed approach.
Chapter 2

LITERATURE REVIEW

Different algorithms, frameworks, tools and techniques have been developed over time to address the learner group formation problem. Based on recent literature these methods can be classified as genetic algorithmic based approaches [1, 18, 19], intelligent systems [16] novel binary integer programming formulations [11] as well as approaches which suggest semantic group formation frameworks [14]. Further some authors have used greedy algorithms [15] and also Hill climbing algorithms [4] to formulate student groups. Clustering algorithms such as Fuzzy C-means [5] have also been used to obtain homogeneous and heterogeneous groups.

In [16] authors have introduced a Multi-Agent system called *I-MINDS* (Intelligent Multiagent Infrastructure for Distributed Systems in Education) which was based on a multiagent coalition formation algorithm. In the system each student, instructor and each group were represented using an intelligent agent. Teacher agent was responsible to disseminating information, maintain student profiles, and to evaluate the progress and participation of different students as well as to manage the progress of classroom sessions. Student agents were created to act as personal helpers to students while group agents were used to formulate and conduct structured cooperative learning sessions using different learning flow patterns, i.e. Jigsaw.

Assignment of students to project groups was considered during a research conducted by authors in [15]. Time slots in which students are available to work on group projects are given priority when forming student groups. Further, project preferences and balanced levels of experience in the groups were also given attention. Algorithm started with the tightest constraint in which students who are having less availability were rated highly and program tried to find a suitable group for them. Then the algorithm was executed further until all students have been allocated to a group. However instructor has to manually adjust these groups for even distribution of grades and for students who are left unassigned to groups.

Team-Maker, a web based system which assigns students to teams based on instructor defined criteria was implemented by authors of [4]. A Hill climbing algorithm was used to obtain optimal groups and authors have mentioned that the Team-Maker outperforms manual group formation. However as mentioned by authors themselves, one limitation of using this system was its inability to provide instant feedback to user input. Similarly in [14] authors proposed a framework for learner group formation based on constraints defined by the person forming groups by reasoning over semantic data about potential participants. Authors have confirmed that use of both semantic web technologies and logic programming increased the constraint satisfaction and it helped to overcome the problem of having
orphan students who have not been assigned to any group. Group generation was based on a DLV solver and the implementation was done using disjunctive logic programming.

In [19] authors suggested a grouping system called DIANA based on a genetic algorithmic formulation. Authors have prioritized student’s thinking styles to formulate heterogeneous groups in order to promote intra group interactions. According to authors system could consider up to seven variables of different student characteristics. Similarly in [1] authors presented a genetic algorithmic based evolutionary algorithm for forming learner groups based on some homogeneous and heterogeneous attributes. Work of [18] also resembles some similarity towards work of [1, 19] since they have also implemented a similar approach using genetic algorithms for task allocation to learners. Implementation of the algorithm was done using MATLAB and evaluations were done in a simulated environment.

In [5] authors have developed a tool to formulate homogeneous and heterogeneous groups. A clustering algorithm called Fuzzy C-means has been used to obtain homogeneous groups while a random selection algorithm was developed to obtain heterogeneous groups. Due to the fact that the generated clusters may result in different sizes instructors required to manually adjust student groups by observing the probabilities generated by the tool for each student belonging to a particular group. Authors highlighted that it is important to consider learner’s negotiation ability towards the groups they belong to.

A novel approach called Enhanced Particle Swarm Optimization (EPSO) have been incorporated to group formation satisfying two grouping criteria (understanding level and interests of students) in collaborative learning context by authors in [12]. Authors have mentioned that this approach could be used to group learners according to various other features apart from two grouping criteria such as learning style, learning motivation or social relations based on pedagogical objectives.

In [11] authors proposed a binary integer programming approach in order to optimally assign learners to appropriate groups. Individual interests were taken into account when formulating student groups. Constraints were modeled to limit the size of groups and the interaction between different users. The model was solved using an optimization tool called CPLEX v12.4.

A summary of the literature on group formation is presented in Table 2.1 summarizing the group formation criteria, problem modeling approach and learner characteristics used when forming collaborative learning groups.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Group Formation Criteria</th>
<th>Approach</th>
<th>Learner Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Instructor defined criteria</td>
<td>Genetic algorithmic approach</td>
<td>Homogeneous &amp; Heterogeneous attributes</td>
</tr>
<tr>
<td>[4]</td>
<td>Instructor defined criteria</td>
<td>A hill climbing algorithm</td>
<td>Preferred time slots, course grades, writing skills</td>
</tr>
<tr>
<td>[5]</td>
<td>Instructor defined criteria</td>
<td>Fuzzy C-means</td>
<td>Knowledge level &amp; learning styles</td>
</tr>
<tr>
<td>[11]</td>
<td>Instructor defined criteria</td>
<td>Binary integer programming approach</td>
<td>Student interests</td>
</tr>
<tr>
<td>[12]</td>
<td>Two grouping criteria</td>
<td>Enhanced Particle Swarm Optimization</td>
<td>Understanding levels &amp; interests</td>
</tr>
<tr>
<td>[14]</td>
<td>Instructor defined criteria</td>
<td>A Semantic group formation framework</td>
<td>Gender, nationality, age, marks, team roles, interests &amp; learning styles</td>
</tr>
<tr>
<td>[15]</td>
<td>Instructor defined criteria</td>
<td>A greedy algorithm</td>
<td>Preferred time slots &amp; projects</td>
</tr>
<tr>
<td>[16]</td>
<td>Self-selecting</td>
<td>Multi-agent coalition algorithm</td>
<td>Learner activities during collaboration</td>
</tr>
<tr>
<td>[18]</td>
<td>Jigsaw CLFP</td>
<td>Genetic algorithmic approach</td>
<td>Learning styles &amp; teamwork skills</td>
</tr>
<tr>
<td>[19]</td>
<td>Instructor defined criteria</td>
<td>Genetic algorithmic approach</td>
<td>Thinking styles</td>
</tr>
</tbody>
</table>

Table 2.1: A Summary of related literature

### 2.1 Limitations of existing methods

As pointed out by [1] major problem of using clustering algorithms to formulate student groups is the difficulty associated when obtaining clusters of balanced size. In order to solve this problem, a refinement algorithm needs to be applied to the result set, which ended up in decreased efficiency and accuracy of the solution.

In [14] authors have mentioned that most systems only model a fixed set of parameters, which does not allow the formation of different types of groups. Hence implementation of different collaborative activities only supports some types of teams. Further, none of the existing efforts have discussed the performance of the applications when data about the users are incomplete.

According to [14] existing applications could handle only a minor number of constraints when forming student groups. The grouping system called DIANA...
which seemed to handle the highest number of constraints [19], was limited to a maximum number of 7 constraints which raises the issues on the scalability of the systems developed so far.

In general, it was noticed that many authors have not discussed about the scalability of the solutions they have proposed, hence lack of information was gained regarding how to put the suggested solutions into practice. On the other hand, many authors who have evaluated the performances of the suggested approaches have only considered a less number of students and constraints for evaluation. Hence it became difficult to obtain an idea about the robustness of those applications. For instance in [5] authors evaluated the tool only on eighteen students using one grouping criteria (Learning styles of students). In [16] authors have done experiments using 15-25 students. In [4] although authors have mentioned that optimal grouping could be obtained using a hill climbing algorithm, they have not mentioned about the complexity of the algorithm or how the performance got affected when increasing the number of constraints. Similarly in [11] authors have evaluated the system only using 32 learners using one constraint (individual interests). Further, in [19] although authors have mentioned that their system could consider up to seven constraints they have not provided test results supporting their argument. In [1] authors have not mentioned about the size of the groups formed, however they have mentioned that the response time of the algorithm was not reasonable.

Moreover, during the literature review it was noticed that although different authors have tried out different methodologies to solve group formation problem, difficulty occurs when trying to reproduce their research. Many authors have not provided a comprehensive description of the datasets used during tests and those who have provided source codes are linked to broken links [4, 15].
Chapter 3

PROPOSED ALGORITHM

During the thesis, assignment of students to groups, while fulfilling pedagogical design of group formation has been modeled using constraint optimization techniques. Constraint optimization can be described as the process of optimizing an objective function with respect to some variables in the presence of constraints on those variables [20]. Constraints can be either hard constraints which set conditions for the variables that are required to be satisfied, or soft constraints which have some variable values that are penalized in the objective function if, and based on the extent that, the conditions on the variables are not satisfied.

When considering the nature of the group formation problem being addressed during the thesis, it was seen that constraint optimization could be adapted to solve this problem. As it was described in section 1.2.1 two main types of constraints applied when forming learner groups are known as intrinsic constraints and extrinsic constraints. Further, the CLFP considered during the thesis was the Jigsaw CLFP, hence it became important to clearly identify the intrinsic/hard constraints and extrinsic/soft constraints of this CLFP before modeling the problem using mathematical notations.

As described in [13] intrinsic constraints which are needed to be adhered when forming groups in Expert phase and Jigsaw Phase are described in Table 3.1.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Intrinsic or Hard Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert</td>
<td>A student can study only one task</td>
</tr>
<tr>
<td>Phase</td>
<td>Students should be distributed accordingly, so that there is a minimum number of students who are allocated to study a particular task</td>
</tr>
<tr>
<td>Jigsaw</td>
<td>A student can work only in one Jigsaw group</td>
</tr>
<tr>
<td>Phase</td>
<td>There should be at least one student for each task within the Jigsaw group</td>
</tr>
<tr>
<td></td>
<td>Each Jigsaw group should have a minimum number of students (based on the requirements of the instructor)</td>
</tr>
</tbody>
</table>

Table 3.1: Intrinsic/Hard constraints applied on Jigsaw CLFP

Different Extrinsic/Soft constraints which could be applied in each phase in order to make a learning scenario more meaningful are described in Table 3.2. However, it should be noted that depending on the context of learning
Phase | Extrinsic or Soft Constraints
--- | ---
Expert Phase | To the extent possible, participants in same expert group should have different educational backgrounds (to favor interesting conversation).
 | To the extent possible, participants in same expert group should have different levels of performances in the average of previous assignment marks (to favor balance across groups).
 | To the extent possible, participants in same expert group should have different levels of research experiences (to favor balance across groups).
Jigsaw Phase | To the extent possible, participants should have possessed the same educational level (to favor significance of conversations and outcome).
 | To the extent possible, participants should have similar language preferences for collaboration (to favor conversation).
 | To the extent possible, participants should share similar interests (to favor significance of conversations and outcome).

Table 3.2: Extrinsic/Soft Constraints applicable for Jigsaw CLFP

environment and based on the expectations of the instructor, nature of the soft constraints as well as the number of soft constraints required to be satisfied within a single phase could differ.

### 3.1 Problem Modeling

A scenario with aforementioned hard constraints in Jigsaw CLFP can be modeled as a constraint optimization problem, using mathematical notations as follows.

Given a total set of $T$ tasks, $N$ students the problem is to assign tasks for each pair of students with the goal of minimizing the cost incurred when assigning tasks to students. The Phase 01 of the problem can be modeled as follows:

\[
\text{Minimize } \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{T} X_{ik}X_{jk}C_{ij} \tag{3.1}
\]

\[\text{Subject to}\]

\[
\sum_{k=1}^{T} X_{ik} = 1 \quad \forall i \in \{1, \ldots, N\} \tag{3.2}
\]

\[
\sum_{i=1}^{N} X_{ik} \geq 1 \quad \forall k \in \{1, \ldots, T\} \tag{3.3}
\]
where $X_{ik}$ denotes assigning student $i$ to task $k$, $X_{jk}$ denotes assigning student $j$ to task $k$. For each pair of students $i$ and $j$, the cost $C_{ij}$ is included as a term in the objective function precisely when $i$ and $j$ are assigned to the same task $k$. Cost $C_{ij}$ could take on any value larger than or equal to 0 depending on the soft constraints applied in each learning scenario (described below). Further constraint (3.2) ensures that each individual $i$ can be assigned to only one task $k$. Constraint (3.3) guarantees that each task $k$ is assigned to at least one student. Similarly in Phase 02, we can formulate $M$ number of total Jigsaw groups with the goal of minimizing the cost incurred when assigning students to Jigsaw groups. However, during this phase an additional constraint (3.4) has been added to the model to make sure that at least one student from each task (from phase 01) is presented in each Jigsaw group.

$$\sum_{i=1}^{N} B_{im}X_{ik} \geq 1 \quad \forall k \in \{1, ..., T\}, \forall m \in \{1, ..., M\} \quad (3.4)$$

$B_{im}$ denotes assigning student $i$ to Jigsaw group $m$, $X_{ik}$ denotes the previous task assignment (during phase 01) of student $i$ to task $k$.

During problem modeling, extrinsic/soft constraints for a particular phase have been incorporated into the objective function parameters. Hence if, and based on the extent that, the conditions on the variables are not satisfied (hard constraints) the soft constraints which have some variable values in the objective function would be penalized. Following three scenarios discuss how the objective function parameters could be encoded differently depending on the requirement of formulating homogenous and heterogeneous student groups.

**Scenario 01 Description (Considering homogeneity of student data)**

To the extent possible participants who are allocated to the same task during phase 01 are required to have similar language preferences for collaboration and they should share similar interests.

In this scenario extrinsic constraints are related to homogeneity of student data since, **similar language preferences** and **similar interests** are considered. Based on the soft constraints specified, cost term $C_{ij}$ associated with a pair of students $i$ and $j$ could be defined as follows:

- $C_{ij} = 0$, if both students have similar language preferences and similar interests
- $C_{ij} = 2$, if both students have different language preferences and different interests
- $C_{ij} = 1$, otherwise (i.e. $i$ and $j$ differ in one parameter but not the other)

**Scenario 02 Description (Considering heterogeneity of student data)**

To the extent possible participants who are allocated to the same task during phase 01 are required to have different educational backgrounds and they should belong to different countries.
In this scenario extrinsic constraints are related to heterogeneity of student data since, different educational backgrounds and different countries are considered. Based on the soft constraints specified, cost term $C_{ij}$ associated with a pair of students $i$ and $j$ could be defined as follows:

- $C_{ij} = 0$, if both students have different educational backgrounds and if they belong to different countries
- $C_{ij} = 2$, if both students have similar educational backgrounds and if they belong to the same country
- $C_{ij} = 1$, otherwise (i.e. $i$ and $j$ differ in one parameter but not the other)

**Scenario 03 Description (Considering both homogeneity and heterogeneity)**

To the extent possible participants who are allocated to the same task during phase 01 are required to have similar knowledge levels and they should belong to different gender categories.

In this scenario soft constraints are related to both homogeneity and heterogeneity of student data since, similar knowledge levels and different gender categories are considered. Based on the soft constraints specified, cost term $C_{ij}$ associated with a pair of students $i$ and $j$ could be defined as follows:

- $C_{ij} = 0$, if both students have similar knowledge levels and if they belong to different genders
- $C_{ij} = 2$, if both students have different knowledge levels and if they belong to similar genders
- $C_{ij} = 1$, otherwise (i.e. $i$ and $j$ differ in one parameter but not the other)

As illustrated in each of the three different scenarios above, during this thesis we have defined the optimization problem as a cost minimization. By minimizing the cost function we could achieve soft constraints specified when grouping students. As it was described during the literature review authors in [11] has described a similar approach for group formation using a binary integer programming approach. During their work individual student interest has been modeled using a people/interest matrix and measured using a cosine coefficient measure. These similarities were then incorporated into the objective function parameters. Finally, by maximizing the objective function learner groups with maximum similarities were obtained (homogeneous groups).

### 3.2 Implementing Flexibility

As described in research objectives attention was also given on the flexibility of the computational script which supports group formation based on Jigsaw CLFP. As
it was described earlier, flexibility requirements occur when defining groups and when changing groups.

When defining groups instructors may require the flexibility to change the group size based on students presented in the class. Since algorithm was designed to take the group size as an input parameter this requirement was addressed during implementation. After obtaining the group size as an input, algorithm evaluates constraints specified for group formation. If the minimum number of students required to formulate student groups based on constraints specified are not presented the system displays the message that no grouping is possible.

Attention was also given on how to handle flexibility requirement associated with changing existing groups and regrouping learners. When regrouping students, algorithm was expected to behave in such a manner so that it reformulates student groups while minimizing changes to existing group structures. Further, regrouping should not result in violations of intrinsic constraints and it should also ensure that the extrinsic constraints were satisfied to the best extent possible.

Considering all these aspects we have modeled the problem associated with flexibility requirement using a similar approach as described in section 3.1. We have modeled the problem so that changes would occur only in the objective function parameters, since no changes should be made to the intrinsic constraints as described earlier using equations (3.2), (3.3) and (3.4). The cost parameters defined earlier are as follows:

\[ C_{ij} = 0, \text{ if students } i \text{ and } j \text{ have been assigned to the same task} \]
\[ C_{ij} = 1, \text{ if students } i \text{ and } j \text{ have been assigned to the different tasks} \]

Intuitively, by maximizing the objective function parameters in these settings, the algorithm could assign students who have been assigned to different tasks early to the same task, so that it will minimize the changes to the existing groups when reformulating new student groups.
Chapter 4

TESTING & EVALUATION

After modeling the problem as a constraint optimization problem, a well-known solver IBM-ILOG Optimization Studio Linux Version 12.6.0 has been used to solve different problem instances. Conceptually, the solving process consisted of two stages. Firstly, the solver searches for candidate solutions that fulfill all hard constraints. Then, these solutions are evaluated against the soft constraints as a minimization or a maximization of the objective function value. Finally, the solver outputs a solution which provides maximum satisfaction for the constraints specified.

![Diagram](image)

Figure 4.1: System Design

4.1 Test Settings

The system was set-up on a personal computer Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz X 4 with 4GB of RAM. The implementation was done using IBM ILOG Optimization Programming Language (OPL), and Oracle Database 11g Release 2 was used as the database. Experiments were carried out using both simulated and real world datasets. During the initial experiments in order to reflect whether the algorithm formulates meaningful student groups based on the given constraints a Group Formation Design Analysis has been carried out. Subsequently a Regrouping Analysis and a Performance Analysis has been carried out.

---

1https://www-01.ibm.com/software/commerce/optimization/modeling/
using a number of test scenarios in order to test the implementation of flexibility and scalability of the suggested approach. Group formation quality during each test scenario have been reported as a qualitative analysis based on constraint satisfaction since a well defined approach to evaluate group formation has not been introduced yet [14].

4.2 Group Formation Design Analysis

Firstly, we have evaluated the outcome of the algorithm in order to verify that the formulated collaborative learning groups are in accordance with the constraints specified. With the guidance of the supervisor, we have selected five different meaningful test scenarios by changing the number and the type of soft constraints applied in each phase. For the clarity of explanation during each test scenario fairly a small number of students (N=25) have been considered and resulted group structures have been reported (See sections 4.2.1 and 4.2.2).

In general, an individual soft constraint applied during a particular phase of Jigsaw CLFP could be defined using the function $p_{ij}$ as follows:

$\begin{align*} p_{ij} &= 0, \text{ if the pair of students } i \text{ and } j \text{ satisfy the soft constraint specified} \\ p_{ij} &= 1, \text{ if the pair of students } i \text{ and } j \text{ do not satisfy the soft constraint specified}\end{align*}$

For problem instances where, a single soft constraint is applied (as in test 02 in both sections 4.2.1 and 4.2.2), the cost term of the integer program $C_{ij}$ could be evaluated as follows:

$C_{ij} = p_{ij}$ (4.1)

However, when there are multiple soft constraints applied during a particular phase (as in test 03, 04 and 05 in both sections 4.2.1 and 4.2.2), we have to define each soft constraint separately (ex: $p_{ij}$, $q_{ij}$, and $r_{ij}$ ) and the cost term $C_{ij}$ could be evaluated as follows:

$C_{ij} = p_{ij} + q_{ij} + r_{ij}$ (4.2)

4.2.1 Tests conducted using simulated dataset

During the thesis, the algorithm implementation was tested using simulated data as well as real world data to analyze how algorithm formulate student groups based on given constraints using different datasets. Simulated dataset is presented in Appendix A, which consists 25 student records. Table 4.1 provides a summary of the dataset.
Student Feature Description

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Learners (LID)</td>
<td>25 Learners were divided to five different educational backgrounds. Each background was represented using a number from 1 to 5.</td>
</tr>
<tr>
<td>Educational Background (BG)</td>
<td>Learners were divided to five different educational backgrounds. Each background was represented using a number from 1 to 5.</td>
</tr>
<tr>
<td>Language Preference (LP)</td>
<td>Only 2 different language codes were used and represented using 1 &amp; 2.</td>
</tr>
<tr>
<td>Knowledge (KN)</td>
<td>Learner’s knowledge on a particular topic was represented using ‘Yes’ or ‘No’ responses and represented using ‘1’ for Yes and ‘0’ for No.</td>
</tr>
<tr>
<td>Age Group (AG)</td>
<td>Two majors age categories were created (Below 40 years of age and Over 40 years of age).</td>
</tr>
<tr>
<td>Research Experience (RE)</td>
<td>Students were divided to two categories and represented using ‘1’ for those who have research experiences and ‘0’ for those who do not have previous experiences.</td>
</tr>
</tbody>
</table>

Table 4.1: Description of simulated dataset

Following subsections describes different test scenarios carried out by varying the number and type of soft constraints applied in each phase of Jigsaw CLFP, along with the test results (group formation design and execution time) and group formation quality evaluations. Further, group size in each scenario, number of tasks allocated during phase 01 and number of Jigsaw groups formed during phase 02 were set to five for clarity in presentation of group structures in each test scenario.

- **Test 01: Excluding soft constraints**
  
  During test 01 groups were formed excluding soft constraints. Hence algorithm was expected to satisfy only the hard constraints applicable as described earlier in Table 3.1. Results obtained during two phases are shown in Table 4.3 and Table 4.4. Related Cost Function Parameters are as follows:

  \[ C_{ij} = 1, \text{ if the pair of students } i \text{ and } j \text{ are assigned to the same task} \]

  \[ C_{ij} = 0, \text{ if the pair of students } i \text{ and } j \text{ are assigned to different tasks} \]

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Learners</td>
<td>Minimum No. of Learners</td>
</tr>
<tr>
<td>Phase 01</td>
<td>25</td>
</tr>
<tr>
<td>Phase 02</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 4.2: Input Parameters & Execution Time - Test 01 (Simulated Dataset)
<table>
<thead>
<tr>
<th>Learners for</th>
<th>Learners for</th>
<th>Learners for</th>
<th>Learners for</th>
<th>Learners for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>Task 2</td>
<td>Task 3</td>
<td>Task 4</td>
<td>Task 5</td>
</tr>
<tr>
<td>19</td>
<td>12</td>
<td>9</td>
<td>17</td>
<td>14</td>
</tr>
<tr>
<td>18</td>
<td>13</td>
<td>8</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>7</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>11</td>
<td>6</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>22</td>
<td>20</td>
<td>10</td>
<td>24</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 4.3: Test 01 Results: Expert Groups (Simulated Dataset)

<table>
<thead>
<tr>
<th>Learners for</th>
<th>Learners for</th>
<th>Learners for</th>
<th>Learners for</th>
<th>Learners for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jigsaw</td>
<td>Jigsaw</td>
<td>Jigsaw</td>
<td>Jigsaw</td>
<td>Jigsaw</td>
</tr>
<tr>
<td>Group 1</td>
<td>Group 2</td>
<td>Group 3</td>
<td>Group 4</td>
<td>Group 5</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>12</td>
<td>21</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>8</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>22</td>
<td>19</td>
<td>1</td>
<td>7</td>
<td>25</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>17</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>24</td>
<td>9</td>
<td>16</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4.4: Test 01 Results: Jigsaw Groups (Simulated Dataset)

*Group formation quality evaluation:*

Based on test results it can be observed that resulted learner groups have satisfied hard constraints applied for both phases as described early in Table 3.1. Execution of the algorithm has finished within acceptable time frames, hence the resulted groups were the optimal solutions for the given problem instance.

- **Test 02: One soft constraint per phase**
  During test 02, soft constraint considered was related to student’s educational background. Groups formed by the algorithm was evaluated such that during expert phase participants in the same expert group required to have different educational backgrounds while in Jigsaw phase participants in each Jigsaw group required to have similar educational backgrounds. Results are shown in Table 4.6 and Table 4.7.
<table>
<thead>
<tr>
<th></th>
<th>No. of Learners</th>
<th>Minimum No. of Learners</th>
<th>No. of Tasks/Groups</th>
<th>Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 01</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>0.02</td>
</tr>
<tr>
<td>Phase 02</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 4.5: Input Parameters & Execution Time - Test 02 (Simulated Dataset)

<table>
<thead>
<tr>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
<th>Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>BG</td>
<td>LID</td>
<td>BG</td>
<td>LID</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>23</td>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>19</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>11</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>25</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 4.6: Test 02 Results: Expert Groups (Simulated Dataset)

<table>
<thead>
<tr>
<th>Jigsaw Group 1</th>
<th>Jigsaw Group 2</th>
<th>Jigsaw Group 3</th>
<th>Jigsaw Group 4</th>
<th>Jigsaw Group 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>BG</td>
<td>LID</td>
<td>BG</td>
<td>LID</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>9</td>
<td>2</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>10</td>
<td>2</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 4.7: Test 02 Results: Jigsaw Groups (Simulated Dataset)

Group formation quality evaluation:
As in test 01, during test 02 also resulted learner groups have satisfied all the hard constraints and the soft constraint applied for both phases. In expert phase, learner’s allocated to the same task have different educational backgrounds while in Jigsaw phase learner’s allocated to the same Jigsaw group have similar educational backgrounds. Since execution of the algorithm has finished within acceptable time frames the resulted groups were the optimal solutions for the given problem instance.

• **Test 03 : Two soft constraints per phase**
During test 03, group formation design was evaluated using two different soft constraints in each phase. Groups formed by the algorithm was evaluated such that during expert phase participants in the same expert group
required to have different educational backgrounds and also they should belong to distinct age categories while in Jigsaw phase participants in each Jigsaw group required to have similar educational backgrounds and they should have similar language preferences for collaboration. Results are shown in Table 4.9 and Table 4.10.

<table>
<thead>
<tr>
<th>No. of Learners</th>
<th>Minimum No. of Learners</th>
<th>No. of Tasks/Groups</th>
<th>Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 01</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Phase 02</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.8: Input Parameters & Execution Time - Test 03 (Simulated Dataset)

Task 1 | Task 2 | Task 3 | Task 4 | Task 5
LID | BG | AG | LID | BG | AG | LID | BG | AG | LID | BG | AG | LID | BG | AG
4 | 1 | 1 | 23 | 5 | 1 | 22 | 5 | 1 | 6 | 2 | 0 | 24 | 5 | 1 |
17 | 4 | 0 | 19 | 4 | 0 | 3 | 1 | 1 | 14 | 3 | 1 | 16 | 4 | 0 |
15 | 3 | 1 | 11 | 3 | 1 | 20 | 4 | 0 | 21 | 5 | 1 | 8 | 2 | 0 |
25 | 5 | 1 | 2 | 1 | 1 | 10 | 2 | 0 | 5 | 1 | 1 | 1 | 1 | 1 |
9 | 2 | 0 | 7 | 2 | 0 | 12 | 3 | 1 | 18 | 4 | 0 | 13 | 3 | 1 |

Table 4.9: Test 03 Results: Expert Groups (Simulated Dataset)

<table>
<thead>
<tr>
<th>Jigsaw Group 1</th>
<th>Jigsaw Group 2</th>
<th>Jigsaw Group 3</th>
<th>Jigsaw Group 4</th>
<th>Jigsaw Group 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>BG</td>
<td>LP</td>
<td>LID</td>
<td>BG</td>
</tr>
</tbody>
</table>
21 | 5 | 2 | 5 | 1 | 1 | 9 | 2 | 1 | 15 | 3 | 2 | 19 | 4 | 2 |
22 | 5 | 2 | 4 | 1 | 1 | 8 | 2 | 1 | 14 | 3 | 2 | 18 | 4 | 2 |
23 | 5 | 2 | 3 | 1 | 1 | 7 | 2 | 1 | 13 | 3 | 1 | 17 | 4 | 2 |
24 | 5 | 2 | 1 | 1 | 1 | 6 | 2 | 1 | 11 | 3 | 1 | 16 | 4 | 2 |
25 | 5 | 2 | 2 | 1 | 1 | 10 | 2 | 1 | 12 | 3 | 1 | 20 | 4 | 2 |

Table 4.10: Test 03 Results: Jigsaw Groups (Simulated Dataset)

Group formation quality evaluation:
Based on the test results it can be seen that during expert phase, task allocation was done while having a maximum difference between learners based on the two soft constraints specified. During Jigsaw phase algorithm has formulated homogeneous groups based on two soft constraints specified except for group 4 in which some heterogeneity was noticed based on the language preferences of learners. However in both phases resulted learner groups have satisfied all the hard constraints specified in Table 3.1.
Test 04: Three soft constraints per phase

During test 04, group formation design was evaluated using three different soft constraints applied in each phase. Groups formed by the algorithm were evaluated such that during expert phase participants in the same expert group required to have different educational backgrounds, they should belong to distinct age categories and also to the extent possible they required to have different research experiences. In Jigsaw phase participants in each Jigsaw group required to have a similar educational background, they should have similar language preferences for collaboration and similar knowledge levels. Results are shown in Table 4.12 and Table 4.13.

<table>
<thead>
<tr>
<th>No. of Learners</th>
<th>Minimum No. of Learners</th>
<th>No. of Tasks/Groups</th>
<th>Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 01</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Phase 02</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.11: Input Parameters & Execution Time - Test 04 (Simulated Dataset)

<table>
<thead>
<tr>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
<th>Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIDBG</td>
<td>AG RE</td>
<td>LIDBG</td>
<td>AG RE</td>
<td>LIDBG</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>21</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 4.12: Test 04 Results: Expert Groups (Simulated Dataset)

<table>
<thead>
<tr>
<th>Jigsaw Group 1</th>
<th>Jigsaw Group 2</th>
<th>Jigsaw Group 3</th>
<th>Jigsaw Group 4</th>
<th>Jigsaw Group 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIDBG LP KN</td>
<td>LIDBG LP KN</td>
<td>LIDBG LP KN</td>
<td>LIDBG LP KN</td>
<td>LIDBG LP KN</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 4.13: Test 04 Results: Jigsaw Groups (Simulated Dataset)

Group formation quality evaluation:
Based on test results it can be seen that during expert phase learners were allocated to expert groups while having maximum differences among them.
based on the three soft constraints specified except in task 03 and task 05. In task 03 and task 05 allocations, some homogeneity was presented with respect to age category and research experience. However, during Jigsaw phase algorithm has formulated only two completely homogeneous groups (group 3 & 5) based on the three soft constraints specified. In groups 1, 2 and 4 algorithm has formulated heterogeneous groups at least based on a single difference presented by a single learner attribute.

- **Test 05 : Testing different types of soft constraints**
  During the last test group formation design was evaluated using different soft constraints which promote homogeneity and heterogeneity of student data at the same time in each phase forming mixed groups. Groups formed by the algorithm were evaluated such that during expert phase participants in the same expert group required to have different educational backgrounds and they required to have similar language preferences for collaboration. In Jigsaw phase participants in each Jigsaw group required to have a similar educational background but different knowledge levels. Results are shown in Table 4.15 and Table 4.16.

<table>
<thead>
<tr>
<th>No. of Learners</th>
<th>Minimum No. of Learners</th>
<th>No. of Tasks/Groups</th>
<th>Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 01</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Phase 02</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.14: Input Parameters & Execution Time - Test 05 (Simulated Dataset)

<table>
<thead>
<tr>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
<th>Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>BG</td>
<td>LP</td>
<td>LID</td>
<td>BG</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td>5</td>
<td>2</td>
<td>22</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>2</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>1</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>1</td>
<td>24</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.15: Test 05 Results: Expert Groups (Simulated Dataset)
Table 4.16: Test 05 Results: Jigsaw Groups (Simulated Dataset)

Group formation quality evaluation:
Based on test results it was noticed that the algorithm has formulated fairly well mixed groups. In expert phase tasks have been allocated in such a way that learners have different educational backgrounds while they have similar language preferences except in task 01 where some heterogeneity is presented. Nevertheless, in Jigsaw phase it was noticed that Jigsaw groups have a well-defined heterogeneity based on knowledge levels but less homogeneity based on educational background.
4.2.2 Tests conducted using real world dataset

This dataset was obtained from authors of [17] who have conducted research on team formation in project based learning environments. Data has been gathered by means of an online survey conducted among learners from School of Psychology and from the Master Educational Sciences of Open University of the Netherlands [17]. The dataset consisted of 168 participants. Table 4.17 provides a description of the dataset.

<table>
<thead>
<tr>
<th>Student Features</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Learners (LID)</td>
<td>168</td>
</tr>
<tr>
<td>Educational Background (BG)</td>
<td>Learners belong to either School of Psychology (represented using 01) or Master in Educational Sciences (represented using 02).</td>
</tr>
<tr>
<td>Gender (GE)</td>
<td>31 Male and 137 Female students represented using 1 for Female and 2 for Male.</td>
</tr>
<tr>
<td>Age Groups (AG)</td>
<td>Learners were categorized into 5 major age categories as follows: 01 (20 - 29), 02 (30 - 39), 03 (40 - 49), 04 (50 - 59) and 05 (60 - 69).</td>
</tr>
<tr>
<td>Research Experience (RE)</td>
<td>Learners have rated their knowledge in the area of formulating a research question and to develop a theoretical design on a scale from 1 to 9.</td>
</tr>
<tr>
<td>Knowledge (KN)</td>
<td>Learners have rated their knowledge in the field of data analysis on a scale from 1 to 10.</td>
</tr>
<tr>
<td>Language Preferences (LP)</td>
<td>Learners have arranged the languages they would like to collaborate during a project. Different languages have been arranged as follows: 1 - Dutch, 2 - English, 3 - German, 4 - French, 5 - Spanish.</td>
</tr>
</tbody>
</table>

Table 4.17: Description of real world dataset

Similar tests conducted using simulated dataset, has been carried out using real world dataset and resulted groups in different learning scenarios are reported below.
• Test 01 : Excluding Soft Constraints

<table>
<thead>
<tr>
<th></th>
<th>No. of Learners</th>
<th>Minimum No. of Learners</th>
<th>No. of Tasks/Groups</th>
<th>Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 01</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>2.05</td>
</tr>
<tr>
<td>Phase 02</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>1.53</td>
</tr>
</tbody>
</table>

Table 4.18: Input Parameters & Execution Time - Test 01 (Real World Dataset)

<table>
<thead>
<tr>
<th>Learners for Task 1</th>
<th>Learners for Task 2</th>
<th>Learners for Task 3</th>
<th>Learners for Task 4</th>
<th>Learners for Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>12</td>
<td>9</td>
<td>17</td>
<td>14</td>
</tr>
<tr>
<td>18</td>
<td>13</td>
<td>8</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>7</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>11</td>
<td>6</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>22</td>
<td>20</td>
<td>10</td>
<td>24</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 4.19: Test 01 Results: Expert Groups (Real World Dataset)

<table>
<thead>
<tr>
<th>Learners for Jigsaw Group 1</th>
<th>Learners for Jigsaw Group 2</th>
<th>Learners for Jigsaw Group 3</th>
<th>Learners for Jigsaw Group 4</th>
<th>Learners for Jigsaw Group 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5</td>
<td>12</td>
<td>21</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>8</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>22</td>
<td>19</td>
<td>1</td>
<td>7</td>
<td>25</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>17</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>24</td>
<td>9</td>
<td>16</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4.20: Test 01 Results: Jigsaw Groups (Real World Dataset)

Group formation quality evaluation:
Similar to tests conducted using simulated dataset resulted learner groups for real world dataset have satisfied all the hard constraints applied for both phases. Since the execution of algorithm finished within acceptable time frames the resulted groups were the optimal solutions for the given problem instance.
• **Test 02 : One soft constraint per phase**

<table>
<thead>
<tr>
<th>No. of Learners</th>
<th>Minimum No. of Learners</th>
<th>No. of Tasks/Groups</th>
<th>Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 01</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Phase 02</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.21: Input Parameters & Execution Time - Test 02 (Real World Dataset)

<table>
<thead>
<tr>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
<th>Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>BG</td>
<td>LID</td>
<td>BG</td>
<td>LID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>10</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>9</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>12</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>13</td>
<td>2</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 4.22: Test 02 Results: Expert Groups (Real World Dataset)

<table>
<thead>
<tr>
<th>Jigsaw Group 1</th>
<th>Jigsaw Group 2</th>
<th>Jigsaw Group 3</th>
<th>Jigsaw Group 4</th>
<th>Jigsaw Group 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>BG</td>
<td>LID</td>
<td>BG</td>
<td>LID</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>17</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.23: Test 02 Results: Jigsaw Groups (Real World Dataset)

*Group formation quality evaluation:*

As in test 01, during test 02 also, resulted learner groups have satisfied all the hard constraints and the soft constraint applied in both phases. Expert groups have promoted heterogeneity based on the educational background while Jigsaw groups have obtained homogeneity based on educational backgrounds. Since the execution of algorithm has finished within acceptable time frames the resulted groups were the optimal solutions for the given problem instance.
• Test 03 : Two soft constraints per phase

<table>
<thead>
<tr>
<th>No. of Learners</th>
<th>Minimum No.of Learners</th>
<th>No. of Tasks/ Groups</th>
<th>Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 01</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Phase 02</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.24: Input Parameters & Execution Time - Test 03 (Real World Dataset)

<table>
<thead>
<tr>
<th>Task</th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
<th>Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>BG</td>
<td>AG</td>
<td>LID</td>
<td>BG</td>
<td>AG</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>3</td>
<td>21</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3</td>
<td>23</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.25: Test 03 Results: Expert Groups (Real World Dataset)

<table>
<thead>
<tr>
<th>Jigsaw Group 1</th>
<th>Jigsaw Group 2</th>
<th>Jigsaw Group 3</th>
<th>Jigsaw Group 4</th>
<th>Jigsaw Group 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>BG</td>
<td>LP</td>
<td>LID</td>
<td>BG</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.26: Test 03 Results: Jigsaw Groups (Real World Dataset)

Group formation quality evaluation:
It can be seen that in expert phase heterogeneity of learner data have been prompted except in task 01 where some homogeneity was presented as the majority of learners have educational background ‘2’. In Jigsaw phase, it was noticed that Groups 2, 4 and 5 are completely homogeneous based on two soft constraints related to learner’s educational background and language preferences. But in Groups 1 and 3 some heterogeneity is presented with respect to educational background and language preferences.
• **Test 04 : Three soft constraints per phase**

<table>
<thead>
<tr>
<th>Phase</th>
<th>No. of Learners</th>
<th>Minimum No.of Learners</th>
<th>No. of Tasks/Groups</th>
<th>Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 01</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>3961.37</td>
</tr>
<tr>
<td>Phase 02</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>40.07</td>
</tr>
</tbody>
</table>

Table 4.27: Input Parameters & Execution Time - Test 04 (Real World Dataset)

<table>
<thead>
<tr>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
<th>Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIDBG</td>
<td>AG RE</td>
<td>LIDBG</td>
<td>AG RE</td>
<td>LIDBG</td>
</tr>
<tr>
<td>1 2 7 8 2 3 5 6 2 3 8 25 1 3 7 12 2 2 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 2 4 5 5 2 2 7 2 2 1 7 11 2 1 6 14 2 3 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17 2 5 3 3 2 1 6 23 1 4 6 4 2 3 6 16 2 4 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 1 1 7 13 2 2 7 22 1 3 6 10 2 4 7 19 1 1 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 1 3 6 24 1 4 8 15 2 2 7 21 1 2 7 9 1 4 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.28: Test 04 Results: Expert Groups (Real World Dataset)

<table>
<thead>
<tr>
<th>Jigsaw Group 1</th>
<th>Jigsaw Group 2</th>
<th>Jigsaw Group 3</th>
<th>Jigsaw Group 4</th>
<th>Jigsaw Group 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIDBG LP KN</td>
<td>LIDBG LP KN</td>
<td>LIDBG LP KN</td>
<td>LIDBG LP KN</td>
<td>LIDBG LP KN</td>
</tr>
<tr>
<td>1 2 1 4 22 1 1 8 24 1 1 6 15 2 2 2 16 2 1 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 2 1 8 21 1 1 6 23 1 1 6 14 2 1 2 11 2 1 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 2 1 4 18 1 1 7 20 1 1 6 10 2 1 5 6 2 2 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 2 1 6 5 2 1 8 19 1 1 3 8 2 1 5 3 2 1 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 2 1 6 9 1 1 3 25 1 1 6 7 2 1 5 17 2 1 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.29: Test 04 Results: Jigsaw Groups (Real World Dataset)

*Group formation quality evaluation:*

It can be seen that in expert phase heterogeneity of learner data have been promoted except in task 02 where some homogeneity is presented as the majority of the learners have educational background ‘2’. In Jigsaw Groups, it was noticed that it is impossible to obtain a completely homogeneous group based on all three soft constraints related to learner’s educational background, language preference and knowledge levels. In almost all the groups, heterogeneity is presented based on knowledge level attribute.
• **Test 05**: Testing different types of soft constraints

<table>
<thead>
<tr>
<th>No. of Learners</th>
<th>Minimum No. of Learners</th>
<th>No. of Tasks/Groups</th>
<th>Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 01</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Phase 02</td>
<td>25</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.30: Input Parameters & Execution Time - Test 05 (Real World Dataset)

<table>
<thead>
<tr>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
<th>Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>BG</td>
<td>LP</td>
<td>LID</td>
<td>BG</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>2</td>
<td>25</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.31: Test 05 Results: Expert Groups (Real World Dataset)

<table>
<thead>
<tr>
<th>Jigsaw Group 1</th>
<th>Jigsaw Group 2</th>
<th>Jigsaw Group 3</th>
<th>Jigsaw Group 4</th>
<th>Jigsaw Group 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID</td>
<td>BG</td>
<td>KN</td>
<td>LID</td>
<td>BG</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>6</td>
<td>17</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>5</td>
<td>12</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.32: Test 05 Results: Jigsaw Groups (Real World Dataset)

**Group formation quality evaluation:**
Based on the resulted grouping structures it was observed that during expert phase algorithm has formulated optimal groups based on the soft constraints specified (learners required to have different educational backgrounds and similar language preferences). During Jigsaw phase algorithm has formulated Jigsaw groups such that all groups have obtained complete homogeneity based on educational background except in group 4. However, heterogeneity is presented in all groups based on students knowledge levels.
4.3 Regrouping Design Analysis

Another objective considered during the thesis was the flexibility requirement which occurs during script ‘run time’ which demands to reformulate existing learner groups due to a member drop out and/or a new member addition to the course in the middle of a learning activity. Further, in such scenarios algorithm requires to regroup students while minimizing changes to existing group structures, since drastic changes in groups would result in frustration within learners, if the grouping activities were planned earlier.

Several tests were carried out as discussed below, in which many students were absent from the collaboration tasks in Phase 01 or in Phase 02 of Jigsaw CLFP and tests are being performed to evaluate how algorithm behaves in such scenarios. Further, the number of learners considered during tests were limited to forty students (N=40), number of tasks required to be allocated during expert phase was set to five and group size was limited to eight.

- **Test Scenario 01: Regrouping during expert phase**
  Activity with groupings were planned, but teacher arrives to the classroom and all students allocated to task 5 (in expert group 5) are missing (marked in gray).

<table>
<thead>
<tr>
<th>Learners for Task 1</th>
<th>Learners for Task 2</th>
<th>Learners for Task 3</th>
<th>Learners for Task 4</th>
<th>Learners for Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>9</td>
<td>37</td>
<td>4</td>
<td>33</td>
</tr>
<tr>
<td>22</td>
<td>12</td>
<td>7</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>26</td>
<td>23</td>
<td>5</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>34</td>
<td>20</td>
<td>27</td>
<td>39</td>
<td>38</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>6</td>
<td>36</td>
<td>01</td>
</tr>
<tr>
<td>19</td>
<td>14</td>
<td>40</td>
<td>32</td>
<td>31</td>
</tr>
<tr>
<td>16</td>
<td>13</td>
<td>18</td>
<td>17</td>
<td>30</td>
</tr>
<tr>
<td>35</td>
<td>11</td>
<td>8</td>
<td>24</td>
<td>29</td>
</tr>
</tbody>
</table>

Table 4.33: Task allocation before regrouping

In here we have tested an extreme scenario in which all students allocated to expert group 05 were missing. A regrouping was carried out in order to allocate students to perform task 5 in expert group 5. During regrouping it was assumed that the instructor’s input for minimum group size (new) was limited to six, so that resulting expert groups required to have at least 6 students. Resulted groups are shown in Table 4.34.
Table 4.34: Task allocation after regrouping

<table>
<thead>
<tr>
<th>Learners for Task 1</th>
<th>Learners for Task 2</th>
<th>Learners for Task 3</th>
<th>Learners for Task 4</th>
<th>Learners for Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>12</td>
<td>7</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>8</td>
<td>17</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td>14</td>
<td>18</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>15</td>
<td>27</td>
<td>32</td>
<td>5</td>
</tr>
<tr>
<td>26</td>
<td>20</td>
<td>37</td>
<td>36</td>
<td>9</td>
</tr>
<tr>
<td>28</td>
<td>23</td>
<td>40</td>
<td>39</td>
<td>11</td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.35: Task allocation after regrouping

<table>
<thead>
<tr>
<th>Learners for Task 1</th>
<th>Learners for Task 2</th>
<th>Learners for Task 3</th>
<th>Learners for Task 4</th>
<th>Learners for Task 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>12</td>
<td>7</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>8</td>
<td>17</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td>14</td>
<td>18</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>15</td>
<td>27</td>
<td>32</td>
<td>5</td>
</tr>
<tr>
<td>26</td>
<td>20</td>
<td>37</td>
<td>36</td>
<td>9</td>
</tr>
<tr>
<td>28</td>
<td>23</td>
<td>40</td>
<td>39</td>
<td>11</td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Group formation quality evaluation:

Based on the regrouping results it can be clearly seen that the algorithm has allocated new students to perform task 05 while minimizing changes to the existing expert group structures. Since the requirement was to allocate minimum six students for each task, the algorithm has cleverly reallocated only six learners to task five while satisfying new group size requirement.

- Test 02: Regrouping during Jigsaw phase

In this scenario, it was assumed that the expert activity was almost done but Jigsaw activity was planned to be done in the classroom. Learner ID’s which are marked in gray color (who have been allocated to different tasks within each Jigsaw group) was assumed to be not presented in the class.

A student regrouping was carried out in order to reallocate students to Jigsaw groups. During this scenario we have set the minimum number of learner’s required to be allocated to each Jigsaw group as 5 and the resulted groups are as follows (Learner ID’s on which changes in groups occurred are marked in gray).
Group Formation Quality Evaluation:
Based on the regrouping results it can be seen that the algorithm reforms student groups while minimizing the changes occurred in existing student groups without violating the constraints. For instance, learner 18 (who was assigned to task 03 in group 01) was absent hence a new learner 6 was added to perform task 03 in group 01. In group 02 although three students 12, 14 and 15 have gone missing the algorithm did not allocate any new learner to this group, because as mentioned during Table 3.1 the hard constraint requires to be preserved here is that there should be at least one student in each Jigsaw group representing a particular task. Since there is at least one learner to represent each task in group 2 no new learners have been allocated. However in group 3 learner 26 (who was assigned to task 01) was missing and the algorithm has cleverly assigned a new learner 21 to fulfill the aforementioned hard constraint. Similarly in group 5 both learners 29 and 33 who was assigned to task 05 was absent and the algorithm has reallocated only a single new learner 30 fulfilling the hard constraint. Similar changes have occurred in group 5 as well.

4.4 Performance Analysis

As described during research objectives, the performance of the proposed group formation approach was also given attention. Hence, a performance analysis was carried out in order to evaluate how an increasing number of students and soft constraints could affect the scalability of the grouping algorithm. Similar tests as described in section 4.2 was carried out while increasing the number of students and soft constraints applied in each phase. Execution time to obtain the optimal groups in each scenario are reported in Table 4.37.
<table>
<thead>
<tr>
<th>No. of Constraints</th>
<th>No. of Students (Expert Group)</th>
<th>Minimum Students</th>
<th>No. of Tasks</th>
<th>Execution Time(sec.)</th>
<th>Minimum Students (Jigsaw Group)</th>
<th>No. of Jigsaw Groups</th>
<th>Execution Time(sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>2.05</td>
<td>5</td>
<td>5</td>
<td>1.53</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>6</td>
<td>5</td>
<td>3.05</td>
<td>10</td>
<td>3</td>
<td>3.74</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>8</td>
<td>5</td>
<td>8.87</td>
<td>10</td>
<td>4</td>
<td>2580.45</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>10</td>
<td>5</td>
<td>Process was killed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>11.11</td>
<td>5</td>
<td>5</td>
<td>4.86</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>6</td>
<td>5</td>
<td>235.27</td>
<td>10</td>
<td>3</td>
<td>1285.21</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>8</td>
<td>5</td>
<td>Process was killed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>164.07</td>
<td>5</td>
<td>5</td>
<td>108.20</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>6</td>
<td>5</td>
<td>2789.39</td>
<td>10</td>
<td>3</td>
<td>1585.21</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>8</td>
<td>5</td>
<td>Process was killed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>5</td>
<td>5</td>
<td>3961.37</td>
<td>5</td>
<td>5</td>
<td>40.07</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>6</td>
<td>5</td>
<td>Process was killed</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.37: Execution time in seconds (Real World Dataset)

4.4.1 Discussion

According to test results, it was observed that algorithm performs fairly well, when it has to satisfy only hard constraints excluding soft constraints (i.e. up to 40 students it took only a few seconds to formulate learner groups). However, it failed to provide optimal grouping results as the number of students increases (i.e. up to 50). Moreover, when the number of soft constraints increased, time it takes to finish execution has also increased drastically. In some test scenarios, it became impossible to obtain Jigsaw groups since execution failed during the expert phase. In general, during most of the successful test scenarios algorithm has taken more time to formulate Jigsaw groups. This observation could be expected since Jigsaw phase is more constrained than the expert phase as described in section 3.1.

In general, according to the test results, it can be concluded that when the problem is less constrained (i.e. one soft constraint per phase, fewer learners) algorithm took less time to finish execution and to provide the optimal solution. But when the problem is more constrained (i.e. three soft constraints per phase, more learners) algorithm takes more time to finish execution.
Chapter 5
PERFORMANCE IMPROVEMENTS

During testing, it became evident when the number of constraints and the number of learners increases algorithm takes more time to finish execution. As it was depicted in Table 4.37 during many test scenarios it became impossible to obtain Jigsaw group formulations, since algorithm failed during the expert phase. Hence as a first step to improve the performance of the algorithm, during the thesis we tried to change the way we have modeled the problem during the expert phase.

5.1 Linearizing Quadratic terms

With the assumption that less performance of the algorithm during expert phase occurred due to quadratic terms presented in both objective function, we have linearized the quadratic terms in order to obtain better performance. Although quadratic assignment problem is known to be one of the most difficult combinatorial problems in mathematical programming, the quadratic terms could be converted to linear expressions reducing the problem to an IP problem [20]. The new model obtained after linearizing quadratic terms in the objective function is described below.

Phase 01 Definition

Given a total number of $N$ students and $T$ tasks let:

$i \in \{1, \ldots, N\}, \quad j \in \{1, \ldots, N\}, \quad k \in \{1, \ldots, T\}$

Desired Parameter : $X_{ik} \in \{0, 1\}$ 1 if student $i$ is assigned to task $k$, 0 otherwise

Objective Function (previous):

$$\text{Minimize } \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{T} X_{ik} X_{jk} C_{ij}$$

(5.1)

When linearizing quadratic terms we have introduced a new (binary) variable $W_{ijk}$ representing the product $X_{ik} X_{jk}$ hence the new objective function becomes:

Objective Function (new):

$$\text{Minimize } \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{T} W_{ijk} C_{ij}$$

(5.2)
Following three linear constraints were introduced in order to ensure that 

\[ W_{ijk} = X_{ik}X_{jk} \]

\[ W_{ijk} \leq X_{ik} \] (5.3)

\[ W_{ijk} \leq X_{jk} \] (5.4)

\[ W_{ijk} \geq X_{ik} + X_{jk} - 1 \] (5.5)

Above three (linear) constraints ensure that \( W_{ijk} = X_{ik}X_{jk} \)

Further, when \( X_{ik} = 0 \) or \( X_{jk} = 0 \), then \( W_{ijk} = 0 \) and if \( X_{ik} = 1 \) and \( X_{jk} = 1 \), based on (5.5), \( W_{ijk} = 1 \).

### Testing Algorithm Performance

After linearizing quadratic terms in the objective function, previous test scenarios were conducted in order to analyze the performance of the algorithm. However, test results were not satisfactory and we were unable to obtain a remarkable difference in execution time with respect to previous experiments.

Nevertheless, we have also setup the experiments in a more powerful virtual machine in order to determine whether the lack of performance has resulted due to lack of resources in the personal computer. However, it was noticed that the execution time was more or less similar to previous test runs.

#### 5.2 Obtaining results as approximations

According to the literature, many authors who have conducted research in group formation in collaborative learning environments using similar approaches have concluded that, finding the optimal solution for the group formation problem using optimization techniques could be time consuming. For instance authors in [12] who have implemented an *Enhanced Particle Swarm Optimization (EPSO)* technique for student group formation using two grouping criteria (based on student’s understanding level and interests) stated,

> “Nevertheless, although EPSO does not guarantee that the final grouping result is the best solution due to the limited computation time allowed in practical applications, it can still obtain a near-optimal solution. Thus, this represents a breakthrough for instructors, as the best solution is almost impossible to derive in real-life contexts.”

Further, in [9] authors who have proposed a mathematical model based on *Ant Colony Optimization* for forming heterogeneous groups based on personality traits and performance levels have mentioned that,

> “the general goal changed from looking for a solution which is close to the optimum to finding a good solution.”

As it was stated above, performance issues associated with optimization techniques when formulating student groups seems much common. As quoted above,
authors who have used optimization techniques for group formation [9, 12] have suggested that when considering the limited computation time allowed during the group formation, it would be advantageous to obtain near-optimal results rather than waiting to obtain optimal group formation.

Based on aforementioned suggestions from previous work done in the field, during the thesis we have also tried to obtain greedy solutions for learner group formation by limiting the time allowed for algorithm execution. Test results are shown in Table 5.1.

<table>
<thead>
<tr>
<th>Number of Soft Constraints</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Students</td>
<td>Expert Phase</td>
<td>Jigsaw Phase</td>
<td>Expert Phase</td>
<td>Jigsaw Phase</td>
</tr>
<tr>
<td>25</td>
<td>2.11</td>
<td>1.62</td>
<td>10.85</td>
<td>4.97</td>
</tr>
<tr>
<td>30</td>
<td>3.27</td>
<td>4.13</td>
<td>60.31</td>
<td>60.01</td>
</tr>
<tr>
<td>40</td>
<td>9.04</td>
<td>7.92</td>
<td>60.00</td>
<td>60.00</td>
</tr>
<tr>
<td>50</td>
<td>13.42</td>
<td>19.46</td>
<td>60.00</td>
<td>60.01</td>
</tr>
<tr>
<td>60</td>
<td>28.08</td>
<td>40.02</td>
<td>60.00</td>
<td>60.01</td>
</tr>
<tr>
<td>70</td>
<td>40.25</td>
<td>40.03</td>
<td>60.00</td>
<td>60.04</td>
</tr>
<tr>
<td>80</td>
<td>60.05</td>
<td>60.03</td>
<td>60.03</td>
<td>61.37</td>
</tr>
<tr>
<td>90</td>
<td>60.05</td>
<td>60.15</td>
<td>60.01</td>
<td>60.07</td>
</tr>
<tr>
<td>100</td>
<td>60.16</td>
<td>60.34</td>
<td>60.03</td>
<td>60.16</td>
</tr>
<tr>
<td>110</td>
<td>60.34</td>
<td>60.53</td>
<td>60.01</td>
<td>60.18</td>
</tr>
<tr>
<td>120</td>
<td>60.32</td>
<td>60.89</td>
<td>60.54</td>
<td>60.23</td>
</tr>
</tbody>
</table>

Table 5.1: Approximation time in seconds (Real World Dataset)

5.2.1 Discussion

As shown in Table 5.1 it can be clearly seen that in almost all the test scenarios, algorithm has provided grouping results within a shorter period of time (less than two seconds). In general, it could be observed that execution time has increased as the number of students and number of constraints increases. We have manually evaluated the group formation quality in each of these test scenarios and it was noticed that acceptable group formations (which satisfy all hard constraints) have been resulted although the execution time was limited. Further, we have evaluated the quality of the greedy solutions against the quality of the optimal solutions only for test scenarios where the number of learners considered is less (N=25 and N=30). It was noticed that greedy solution has resulted in similar grouping structures provided by the optimal solution when the number of students considered are less. However, we could not obtain the optimal solutions when the number of learners is more than 30 as described in Table 4.37 hence it became impossible to compare grouping structures of the greedy solution and optimal solution during such scenarios.

Change of the execution time based on the number of soft constraints added
during the tests conducted so far using different datasets when number of learners are limited to 25 are presented in Figure 5.1 and Figure 5.2. Overall, it can be seen that when the number of constraints are low, time to formulate learner groups are negligible for both datasets. However, as the number of soft constraints increases, execution time has also increased for real dataset but not for the simulated dataset in both phases. However, during expert group formation a significant difference between the execution time taken by greedy solution and the optimal solution was observed as the number of soft constraints increases. On the other hand, during Jigsaw group formation a clear positive relationship between the execution time and the number of constraints could not be observed for both greedy solution and the optimal solution. Execution time to obtain optimal solution has increased as the numbers of soft constraints increase (up to two), and has decreased when three soft constraints were added.

Further, we have evaluated the performance of our algorithm with the results provided by authors who have implemented a similar approach for group formation [11]. During the work of [11], authors have reported that time taken to optimally divide 32 learners to two homogeneous groups (based on one soft constraint) took 8.27 seconds. As a comparison of similar approaches, we have also tested a similar scenario using the real world dataset and it was noticed that our approach took only 0.04 seconds to optimally divide 32 learners to two homogeneous groups. Hence, comparatively it can be argued that the suggested approach performs better than similar work done in the field [11].
Chapter 6
CONCLUSIONS & FUTURE WORK

During the thesis, a novel binary integer programming approach for group formation in collaborative learning environments using Jigsaw CLFP was proposed. The suggested solution can handle different grouping constraints defined with regard to a particular learning scenario hence it addresses the multiple criteria grouping problem. Learner Group formation was based on the CPLEX Optimization engine and implemented in Optimization Programming Language (OPL). To evaluate the algorithm three major types of tests have been carried out namely the group formation design analysis, regrouping design analysis and performance analysis.

The group formation design analysis results had shown that the algorithm formulates meaningful learner groups while preserving intrinsic and extrinsic constraints specified. Based on the extrinsic constraints defined, algorithm resulted in formulating homogeneous, heterogeneous and mixed learner groups. Group formation quality was then assessed against the satisfaction of the constraints specified by the instructor during each learning scenario.

Apart from group formation, another objective of the thesis was to support the flexibility required when reforming learner’s groups. During the tests, it was noticed that the algorithm accurately reformulates student groups while satisfying constraints applied, hence the proposed approach could be used by collaborative learning practitioners in dynamic collaborative learning environments, where changes of existing grouping structures are required.

Moreover, during the performance analysis, it was noticed that increasing the number of students/groups or the constraints applied slow down the solving processes. In complex and large problem instances it was determined that approximations (running algorithm as an any-time solution) could result in a better outcome, which accompanies the cost of loosing optimal solution for the given problem instance.

Hence during the thesis we propose that for learning scenarios in which the instructor could not wait until to obtain the optimal solutions it would be advantageous to obtain approximations since it provides acceptable group formations. This could be especially advantageous when regrouping students during a collaborative learning session on the fly. On the other hand, if instructor is not satisfied with the group formation results obtained during the first round of approximation we suggest that he could increase the execution time allowed based on his requirements inorder obtain different group formation structures. However, it should also be noted here that algorithm performance also depends on the nature of the dataset.
As discussed during section 5.2.1 it was noticed that for datasets in which much learner variations are not observed (simulated dataset) algorithm provides learner groups within seconds. However, aforementioned performance issues occur when the dataset includes more variations in learner data (real world dataset).

Finally, the advantages of the suggested approach over the existing solutions for group formation are as follows. One of the major advantages is that the formulation of the problem using a novel binary integer programming approach which guarantees an optimal solution. The suggested approach allocate every learner to a group while satisfying all the hard constraints and soft constraints to the best extend possible, hence nobody is left alone without a group resulting ‘orphans’. Further, during the literature review, it was noticed that many authors have not discussed about reformulating student groups (flexibility requirement) in unexpected situations. Our approach can gracefully handle this type of situations while minimizing changes to the existing group structure. The groups formulated are balanced in size and it supports instructors by allowing them to formulate groups based on multiple criteria. The algorithm could be utilized in complex grouping scenarios since soft constraints could be clearly and easily modeled using the cost function parameters and results could be obtained as an approximation within acceptable time durations.

As for the future work, it is required to test the suggested approach with more challenging scenarios by increasing the number of students, number of constraints applicable for each phase in order to draw safer conclusions about the adaptability of the algorithm. Different solvers could be tested in order to determine which solver could result in faster solving processes so that algorithm can handle complex grouping scenarios more effectively providing the optimal grouping solutions not the approximations. Further, we can incorporate a weight for each grouping criteria used hence giving the opportunity for the instructor to prioritize the constraints.

Nevertheless during the thesis tests were carried out using complete learner’s data. A much more challenging work would be to test how the algorithm behaves and generate learner groups based on incomplete learner profile details.

Finally this algorithm could be incorporated into an E-Learning environment as a module for forming groups using multiple criteria. A complete framework with supporting graphical user interfaces for instructors and learners could be provided so that instructor would get a chance to intervene into the proposed groupings and to change learner’s among groups manually if required. Further, a group negotiation option could be implemented so that learner group preferences could also be taken into account when forming groups apart from instructor defined criteria. Agreement of learner towards grouping would be more important when reassigning them to different groups changing initial grouping structure.

Since the aforementioned directions seem challenging, further research is required in order to prove the effectiveness and usefulness of the suggested group formation approach.
References


Appendix A
ARTIFICIAL DATASET

<table>
<thead>
<tr>
<th>LID</th>
<th>BG</th>
<th>AG</th>
<th>LP</th>
<th>RE</th>
<th>KN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table A.1: Simulated Dataset
Appendix B

SOURCE CODES

A Sample Model File for Scenarios with No Soft Constraints-Expert Phase

```plaintext
/* OPL 12.5 Model
* Author: Ishari
************************************************************************/

int t = ...;
int minStu = ...;
// range to access array elements
range Task = 1..t;
{int} del = ...;
// set {} of students
{int} student = ...;
execute PARAMS {
    StrongThreadLim = 4;
    WorkMem = 4096;
    TreLim = 4096;
}
// binary decision variable
dvar int t_alloc[student][Task] in 0..1;
// objective function
maximize sum(i in student, j in student, k in Task)
    t_alloc[i][k] * t_alloc[j][k];
// constraints
subject to {
    forall(i in student)
        one_task_per_student:
            sum(j in Task) t_alloc[i][j] == 1;
    forall(j in Task)
        minimum_students_per_task:
            sum(i in student) t_alloc[i][j] >= minStu;
}
tuple result{
    int student;
    int task;
    int t_alloc;
}
{result} Result =
{ <s,t,t_alloc[s][t]> | s in student, t in Task ];
execute DISPLAY_RESULT{
    writeln("Result =", Result)
}
```
A Sample Data File for Scenarios with No Soft Constraints-Expert Phase

```java
/*******************************************************************************
* OPL 12.5 Data
* Author: Ishari
*******************************************************************************/

// number of tasks given as input
t = 5;
minStu = 8;

// connecting to db
DBConnection db("oracle11","dbuser1/123@://localhost:1521/xe");

// access student data
student from DBRead(db,"select S_ID from new_student_data where S_ID <= 40");

// values to be deleted later
del from DBRead(db,"select VAL from TODELETE");

// drop previous result table if any
DBExecute(db,"drop table task_allocation");

// insert data to task allocation
Result to DBUpdate(db,"INSERT INTO task_allocation (S_ID,task,value )
VALUES(:1,:2,:3)");

// update table removing zeros
del to DBUpdate(db,"delete from task_allocation where value= (:1)
");
```

A Sample Model File for Scenarios with No Soft Constraints-Jigsaw Phase

```java
*******************************************************************************
* OPL 12.5 Model
* Author: Ishari
*******************************************************************************/

// parameters
int minStu = ...;

// number of jigsaw groups
int g = ...;

range groups = 1..g;

// number of tasks
int t = ...;

range Task = 1..t;

// student data
{int} student = ...;

{int} del = ...;

execute PARAMS {
  StrongThreadLim = 4;
  WorkMem = 4096;
}
```
TreLim = 4096;

} // task allocation details
int t_alloc[student][Task] = ...;

// decision variable for group allocation
dvar int g_alloc[student][groups] in 0..1;

// objective function—soft constraints
maximize sum(i in student, j in student, m in groups) g_alloc[i][m] * g_alloc[j][m];

// hard constraints
subject to{
    forall (i in student)
        one_group_per_student:
            sum(k in groups) g_alloc[i][k] == 1;
    forall (k in groups)
        to_have_balanced_groups:
            sum(i in student) g_alloc[i][k] >= minStu;
    forall (j in Task)
        forall (k in groups)
            atleast_one_student_per_task_in_group:
                sum(i in student) t_alloc[i][j] * g_alloc[i][k] >= 1;
}

tuple result{
    int student;
    int group;
    float g_alloc;
}

{result} Result =
    {<s, g_alloc[s][g] | s in student, g in groups};
execute DISPLAY_RESULT{
    writeln("Result =", Result)
}

A Sample Data File for Scenarios with No Soft Constraints-Jigsaw Phase

/*****************************************************************************/
/* OPL 12.5 Data
* Author: Ishari
/*****************************************************************************/

t = 5; // number of tasks
g = 4; // number of groups
minStu = 10;
// connecting to db
DBConnection db("oracle11","dbuser1/123@//localhost:1521/xe");

// access student data
student from DBRead(db,"select S_ID from task_allocation");

// values to be deleted later
del from DBRead(db,"select VAL from TODELETE");

// obtain previous task allocation details from db
t_alloc from DBRead(db,"select S_ID,TASK,S_ID
from task_allocation where S_ID <40");

// insert data to group allocation
Result to DBUpdate(db,"INSERT INTO group_allocation(S_ID,G_ID,
VALUE)
VALUES(:1,:2,:3)");

// update table removing zeros
del to DBUpdate(db,"DELETE from group_allocation where value=:1");

A Sample Model File with Soft Constraints-Expert Phase

/****************************
* OPL 12.5 Model
* Author: Ishari
****************************/

// parameters
// input data, number of tasks
int t=...;
range Task=1..t;

// input data, minimum number of students per tasks
int minStu=...;

// delete 0's in post processing
{int} del=...;

// student data
{int} student = ...;
{int} id = ...;

eexecute PARAMS {
    StrongThreadLim = 4;
    WorkMem = 4096;
    TreLim = 4096;
}

// preprocessing block to obtain similar students
tuple bgdetails{
```c
int sid;
int edu;
int age;
}
bgdetails stu_info[id] = ...;
int edu_bg[student][student];
int agediff[student][student];

execute PREPROCESSING{
  stuCount = stu_info.size
  for (i=1; i <= stuCount; i++) {
    for (j=i+1; j <= stuCount; j++) {
      if (stu_info[i].edu == stu_info[j].edu) {
        edu_bg[i][j] = 1
        // writeln("sid", stu_info[i].sid, "sid", stu_info[j].sid)
      } else {
        edu_bg[i][j] = 0
        // writeln("sid", stu_info[i].sid, "sid", stu_info[j].sid)
      }
      if (stu_info[i].age == stu_info[j].age) {
        agediff[i][j] = 1
        // writeln("sid---sid", stu_info[i].sid, stu_info[j].sid)
      } else {
        agediff[i][j] = 0
        // writeln("sid---sid", stu_info[i].sid, stu_info[j].sid)
      }
    }
  }
  // end of preprocessing

  // decision variables for task allocation
  dvar int t_alloc[student][Task] in 0..1;

  // objective function
  minimize sum(i in student, j in student, k in Task) t_alloc[i][k] *
     t_alloc[j][k] *
    (edu_bg[i][j] + agediff[i][j]);

  // constraints
  subject to {
    one_task_per_student:
      forall (i in student)
        sum(j in Task) t_alloc[i][j] == 1;
```
minimum_students_per_task:
forall(j in Task)
sum(i in student) t.alloc[i][j] >= minStu;
}

// Postprocessing to display results
tuple result{
int student;
int task;
float t.alloc;
}

{result} Result=
{ <s,t,t.alloc[s][t]>| s in student, t in Task};

execute DISPLAY_RESULT{
 writeln("Result=", Result)
}

A Sample Data File for Scenarios with Soft Constraints-Expert Phase

/****************************
* OPL 12.5 Data
* Author: Ishari
*******************************/

//number of tasks given as input
 t=5;

//minimum number of students per task given as input
minStu=6;

//connecting to db
DBConnection db("oracle11","dbuser1/123@//localhost:1521/xe");

//access student data
student from DBRead(db," select S_ID from new_student_data where S_ID <=30");

id,stu_info from
DBRead(db," select ID,S_ID,EDU_BG,AGE from new_student_data where S_ID <=30");

//insert data to task allocation
Result to DBUpdate(db," INSERT INTO task_allocation(S_ID,task,value )
 VALUES(:1,:2,:3 ) ");

//update table removing zeros
del from DBRead(db," select VAL from TODELETE");
del to DBUpdate(db," delete from task_allocation where value= (:1) ");

A Sample Model File with Soft Constraints-Jigsaw Phase

49
1 /******************************
2 * OPL 12.5 Model
3 * Author: Ishari
4 *********************************/
5
6 // parameters
7 // input data, number of jigsaw groups
8 int g = ...;
9 range groups = 1..g;
10
11 // input data, number of tasks
12 int t = ...;
13 range Task = 1..t;
14
15 // input data, minimum number of students per group
16 int minStu = ...;
17
18 // student data
19 {int} student = ...;
20 {int} id = ...;
21
22 // delete 0's in post processing
23 {int} del = ...;
24
25 execute PARAMS {
26     StrongThreadLim = 4;
27     WorkMem = 4096;
28     TreLim = 4096;
29 }
30
31 // preprocessing block to obtain students background data
32 tuple bgdetails {
33     int sid;
34     int edu;
35     int lang;
36 };
37
38 bgdetails stu_info[id] = ...;
39
40 int edu_bg[student][student];
41 int langdiff[student][student];
42
43 execute PREPROCESSING{
44     stuCount = stu_info.size
45     for (i=1; i <= stuCount; i++) {
46         for (j=i+1; j <= stuCount; j++) {
47             if (stu_info[i].edu == stu_info[j].edu) {
48                 edu_bg[i][j] = 1
49                 // writeln("sid", stu_info[i].sid, "sid", stu_info[j].sid)
50             }
51             else {
edu_bg[i][j] = 0
    // writeln("sid", stu_info[i].sid, "sid", stu_info[j].sid)
}
if(stu_info[i].lang==stu_info[j].lang){
    langdiff[i][j] = 1
    // writeln("sid---", stu_info[i].sid, "---has similar marks to sid---",
    stu_info[j].sid)
} else{
    langdiff[i][j] = 0
    // writeln("sid", stu_info[i].sid, "sid", stu_info[j].sid)
}
}
}
}
// end of preprocessing

// previous task allocation details
int t_alloc[student][Task] = ...;

// decision variable for group allocation
dvar int g_alloc[student][groups] in 0..1;

// objective function
maximize sum(i in student, j in student, m in groups) g_alloc[i][m]*g_alloc[j][m]*((edu_bg[i][j]+langdiff[i][j]));

// hard constraints
subject to{
    one_group_per_student:
        forall(i in student)
            sum(k in groups) g_alloc[i][k] == 1;

    create_balanced_groups:
        forall(k in groups)
            sum(i in student) g_alloc[i][k] >= minStu;

    atleast_one_student_per_task_in_group:
        forall(j in Task)
            forall(k in groups)
                sum(i in student) t_alloc[i][j] * g_alloc[i][k] >= 1;
}

// Postprocessing to display results
tuple result{
    int student;
    int group;
    float g_alloc;
}
\{result\} Result=
\{ <\ s,g,g_alloc[s][g]>| \ s \ in \ student, \ g \ in \ groups \};

execute DISPLAY_RESULT{
  writeln("Result=", Result)
}

A Sample Data File with Soft Constraints-Jigsaw Phase

A Sample Model File with Soft Constraints-Expert Phase (with execution limits)
11 // input data, minimum number of students per tasks
12 int minStu = ...;
13 // delete 0's in post processing
14 {int} del = ...;
15 // student data
16 {int} student = ...;
17 {int} id = ...;
18
19 execute PARAMS {
20     StrongThreadLim = 4;
21     WorkMem = 4096;
22     TreLim = 4096;
23     cplex.tilim = 60;
24 }
25
26 // preprocessing block to obtain similar students
27
tuple bgdetails{
28     int sid;
29     int edu;
30     int age;
31 };
32
33 bgdetails stu_info[id] = ...;
34
35 int edu_bg[student][student];
36 int agediff[student][student];
37
38 execute PREPROCESSING{
39     stuCount = stu_info.size
40     
41     for (i=1; i <= stuCount; i++){
42         for (j=i+1; j <= stuCount; j++){
43             if (stu_info[i].edu == stu_info[j].edu) {
44                 edu_bg[i][j] = 1
45                 // writeln("sid", stu_info[i].sid, "sid", stu_info[j].sid)
46             }
47         } else {
48                 edu_bg[i][j] = 0
49                 // writeln("sid", stu_info[i].sid, "sid", stu_info[j].sid)
50             }
51             if (stu_info[i].age == stu_info[j].age) {
52                 agediff[i][j] = 1
53                 // writeln("sid---", stu_info[i].sid, "sid---has edu_bg age to sid---", stu_info[j].sid)
54             }
55         } else {
56                 agediff[i][j] = 0
57                 // writeln("sid", stu_info[i].sid, "sid", stu_info[j].sid)
58         }
59     }
60     writeln(edu_bg)

A Sample Model File for Regrouping-Expert Phase

//Sample Model File for Regrouping-Expert Phase

/* OPL 12.5 Model
* Author: Ishari
*/

// parameters
int t =...;
range Task=1..t;
{int} student = ...;
{int} del =...;
execute PARAMS {

//end of preprocessing

// decision variables for task allocation
dvar int t_alloc[student][Task] in 0..1;

// objective function
minimize sum(i in student, j in student, k in Task)
    t_alloc[i][k]*t_alloc[j][k]*(edu_bg[i][j]+agediff[i][j]);

// constraints
subject to{
    one_task_per_student:
        forall(i in student)
            sum(j in Task) t_alloc[i][j] == 1;
    minimum_students_per_task:
        forall(j in Task)
            sum(i in student) t_alloc[i][j] >= minStu;
}

// Postprocessing to display results
tuple result{
    int student;
    int task;
    float t Alloc;
}
{result} Result=
    {<s,t,t Alloc[s][t]>| s in student, t in Task };
execute DISPLAY_RESULT{
    writeln("Result=", Result)
}
13 StrongThreadLim = 4;
14 WorkMem = 4096;
15 TreeLim = 4096;
16 }
17 // task allocation details
18 int t_alloc[student][Task] = ...;
19 // decision variable
dvar boolean new_t_alloc[student][Task];
20 // objective function
21 maximize sum(i in student, j in student, k in Task)
22 new_t_alloc[i][k]*new_t_alloc[j][k]*t_alloc[i][k]*t_alloc[j][k];
23 // constraints
24 subject to{
25 forall(i in student)
26 one_task_per_student:
27 sum(j in Task) new_t_alloc[i][j] == 1;
28 forall(j in Task)
29 at_least_one_student_pertask:
30 sum(i in student) new_t_alloc[i][j] >= 1;
31 }
32 tuple result{
33 int student;
34 int task;
35 float new_t_alloc;
36 }
37 {result} Result =
38 {<s, t, new_t_alloc[s][t]>| s in student, t in Task};
39 execute DISPLAY_RESULT{
40 writeln("Result=", Result)
41 }

A Sample Model File for Regrouping-Jigsaw Phase

1 /***************************************************/
2 * OPL 12.5 Model
3 * Author: Ishari
4 ***************************************************/
5 // parameters
6 // number of jigsaw groups
7 int g = ...;
8 range groups = 1..g;
9 {int} del = ...;
10 // number of tasks
11 // ...
```c
int t = . . . ;
range Task = 1 .. t ;
// student data
{ int } student = . . . ;
execute PARMS {
    StrongThreadLim = 4 ;
    WorkMem = 4096 ;
    TreLim = 4096 ;
}

// task allocation details
int t_alloc [ student ] [ Task ] = . . . ;

// previous group allocation details
int g_alloc [ student ] [ groups ] = . . . ;

decision variable for group allocation
dvar boolean new_g_alloc [ student ] [ groups ];

// objective function—soft constraints
maximize sum ( i in student , j in student , m in groups )
    new_g_alloc [ i ] [ m ] * new_g_alloc [ j ] [ m ] * g_alloc [ i ] [ m ] *
    g_alloc [ j ] [ m ];

// hard constraints
subject to {
    forall ( i in student )
        one_group_per_student :
            sum ( k in groups ) new_g_alloc [ i ] [ k ] == 1 ;

    forall ( j in Task )
        forall ( k in groups )
            atleast_one_student_per_task_in_group :
                sum ( i in student ) t_alloc [ i ] [ j ] * new_g_alloc [ i ] [ k ] >= 2 ;
}

tuple result {
    int student ;
    int groups ;
    int new_g_alloc ;
}

{ result } Result =
    { < s , g , new_g_alloc [ s ] [ g ] > | s in student , g in groups } ;
execute DISPLAY_RESULT{
    writeln ( "Result = ", Result )
}

A Sample Model File with Soft Constraints-Linearised Objective
```

---

/********************
* OPL 12.5 Model
* Author: Ishari
********************/
// parameters
// input data, number of tasks
int t = ...;
range Task = 1..t;

// input data, minimum number of students per tasks
int minStu = ...;

// delete 0's in post processing
{int} del = ...;

// student data
{int} student = ...;
{int} id = ...;

execute PARAMS {
  StrongThreadLim = 4;
  WorkMem = 4096;
  TreLim = 4096;
}

// preprocessing block to obtain similar students
tuple bgdetails{
  int sid;
  int edu;
};
bgdetails stu_info[id] = ...;
int edu_bg[student][student];
execute PREPROCESSING{
  stuCount = stu_info.size
  for (i = 1; i <= stuCount; i++) {
    for (j = i + 1; j <= stuCount; j++) {
      if (stu_info[i].edu == stu_info[j].edu) {
        edu_bg[i][j] = 1
        writeln("sid", stu_info[i].sid, "sid", stu_info[j].sid)
      } else {
        edu_bg[i][j] = 0
        // writeln("sid", stu_info[i].sid, "sid", stu_info[j].sid)
      }
    }
  }
}
// end of preprocessing

// decision variables for task allocation
dvar int t_alloc[student][Task];
dvar int w[student][student][Task] in 0..1;
// objective function
minimize sum(i in student, j in student, k in Task) w[i][j][k] * edu_bg[i][j];

// hard constraints
subject to{
    one_task_per_student:
        for all(i in student)
            sum(j in Task) t_alloc[i][j] == 1;
    minimum_students_per_task:
        for all(j in Task)
            sum(i in student) t_alloc[i][j] >= minStu;

// constraints for linearizing the objective
Cons01:
    for all(i in student)
        for all(j in student)
            for all(k in Task)
                w[i][j][k] <= t_alloc[i][k];
Cons02:
    for all(i in student)
        for all(j in student)
            for all(k in Task)
                w[i][j][k] <= t_alloc[j][k];
Cons03:
    for all(i in student)
        for all(j in student)
            for all(k in Task)
                w[i][j][k] >= t_alloc[i][k] + t_alloc[j][k] - 1;
}

// Postprocessing to display results
tuple result{
    int student;
    int task;
    float t_alloc;
}

{result} Result =
    {<s,t,t_alloc[s][t]>| s in student, t in Task};

evaluate DISPLAY_RESULT{
    writeln("Result =", Result)
}