A Web-Based Decision Support System for Class-Teacher Timetabling Problem

A thesis submitted to the
Graduate School of Natural and Applied Sciences

by

Ahmet ŞAHİN

in partial fulfillment for the
degree of Master of Science

in

Industrial and Systems Engineering

İSTANBUL ŞEHİR UNIVERSITY

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Industrial and Systems Engineering.

APPROVED BY:

Assist. Prof. Hatice Tekiner Moğulkoç ....
(Thesis Advisor)

Assist. Prof. Nuri Cihat Onat ....

Assist. Prof. S. Tankut Atan ....

This is to confirm that this thesis complies with all the standards set by the Graduate School of Natural and Applied Sciences of İstanbul Şehir University:

DATE OF APPROVAL: 19 January 2016

SEAL/SIGNATURE:

Declaration of Authorship

I, Ahmet ŞAHİN, declare that this thesis titled, 'A Web-Based Decision Support System for Class-Teacher Timetabling Problem' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date: 19.01.2017

A Web-Based Decision Support System for Class-Teacher Timetabling Problem

Ahmet ŞAHİN

Abstract

Class-Teacher Timetabling Problem determines weekly timetables for class-teacher pair with respect to certain hard and soft constraints, and institutions' specific constraints. Group balancing problem aims to create heterogeneous structure within the groups, and homogeneous structure between the groups. In this thesis, mathematical models are developed for class-teacher timetabling problem and group balancing problem frequently encountered during the planning of new semester in educational institutions. Additionally, a web-based decision support system (DSS) is designed for the related problems. As a contribution to this sub-field, this study provides tools to the planners to prepare weekly timetables and class lists in a short time.

A mixed-integer model is developed for group balancing problem, and aims to minimize deviation of average score of the generated classes from the general average, and to ensure balanced distribution of students among classes according to gender, international, and repeat status. A binary integer model proposed for class-teacher timetabling problem. This model is formulated as a constraint satisfaction problem with no objective function, but includes general timetabling constraints and specific constraints for the institution.

Many educational institutions still prepare timetables manually due to lack of general applicability of customized software. The end-user interface allows planners to use mathematical model instead of manually preparing weekly timetables and class lists. The exclusive contribution of this study is to design a DSS for the end user. The functions of this DSS include data entry, pre-processing of data, solving the problem, and reporting solution. DSS is based on PHP script, and GAMS is used to solve mathematical models.

The data from İstanbul Şehir University School of Languages English Preparatory Program (SEPP) is used as a case study to demonstrate operation of the system. Class lists and weekly timetables were successfully created within minutes instead of two days taken manually. SEPP approved the timetables and declared that our system satisfies their requests. Class lists and weekly timetables for Spring 2017 have been planned using our system.

Keywords: Class-Teacher Timetabling, Group Balancing Problem, Web-based Decision Support System, Educational Timetabling Problem, Integer Programming

Sınıf-Öğretmen Çizelgeleme Problemi için Web Tabanlı Karar Destek Sistemi

Ahmet ŞAHİN

Öz

Sınıf-Öğretmen Çizelgeleme Problemi, genel sert-yumuşak ve kurumlara özgü kısıtları yerine getirerek, sınıf-öğretmen çifti için haftalık zaman çizelgeleri oluşturmayı içermektedir. Grup dengeleme problemi, oluşturulacak sınıfların kendi içerisinde heterojen, sınıflar arası homojen bir yapı oluşturmayı amaçlamaktadır. Bu tezde, eğitim kurumlarında yeni dönemin planlanmasında sıkça karşılaşılan sınıf-öğretmen çizelgeleme problemi ve grup dengeleme problemi için matematiksel modeller geliştirilmiş ve ilgili problemler için bir web tabanlı karar destek sistemi (DSS) tasarlanmıştır. Bu çalışmanın amacı, planlayanın zaman çizelgelerini ve sınıf listelerini kısa sürede oluşturması için araç geliştirmektir.

Grup dengeleme problemi için oluşturulan karma tam sayı model, sınıfların not ortalamasının genel ortalamadan sapmasını en aza indirmeyi ve sınıf listelerinde öğrencilerin cinsiyet, uluslararası ve tekrar durumuna göre sınıflar arasında dengeli dağılı-mını sağlamayı amaçlamaktadır. Sınıf-öğretmen çizelgeleme problemi için geliştirilen ikili tamsayılı model, genel zaman çizelgesi ve söz konusu kurum için özel kısıtları olan, ancak amaç fonksiyonu olmayan kısıt sağlama problemi olarak formüle edilmiştir.

Birçok eğitim kurumu, kişiselleştirilmiş yazılımların genel uygulanabilirliği eksikliği nedeniyle manuel olarak çizelgeleri hazırlar. Son kullanıcı arayüzü, haftalık zaman çizelgelerini ve sınıf listelerini manuel olarak hazırlamak yerine planlayanın matematiksel yaklaşımları kullanması için gerekli olan bir bileşendir. Son kullanıcı için bir DSS tasarlanmış olması bu çalışmanın çok önemli katkısıdır. Geliştirilen DSS, veri girişi, verilerin ön işlenmesi, sorunun çözülmesi ve çözümün raporlanması için tasarlanmıştır. DSS tasarımında, PHP ana yazılım dili kullanılmış olup, matematiksel modellerin çözümünde GAMS ile yapılmıştır.

Uygulama kısmında, İstanbul Şehir Üniversitesi Diller Okulu'nun verileri test çalışma olarak kullanılmıştır. Yaklaşık iki iş günü süren çalışma ile dakikalar içinde Diller Okulu'nun taleplerini yerine getirilmiş, listeler ve çizelgeler başarılı şekilde oluşturulmuştur. Diller Okulu, sistemin kullanılabilir olduğunu beyan etmiş ve 2017 Bahar dönemi planlaması bu sistem aracılığıyla yapılmıştır.

Anahtar Sözcükler: Sınıf-Öğretmen Çizelgeleme Problemleri, Web Tabanlı Karar Destek Sistemi, Eğitimsel Zaman Çizelgeleme Problemleri, Tamsayılı Programlama

This thesis is dedicated to my charming wife Sena whose unyielding love and support kept me ever motivated, and to my dear son Mehmet Efe whose warmly anticipated and auspicious birth inspired me to conclude this study.

Acknowledgments

Among everyone who have contributed to my personal and academic development hitherto, I owe the most to my beloved parents, Sayime and Mehmet. I lack appropriate words to express my enormous gratitude to them for bringing me to this world. Without their guidance and support, I might not have been the person I am today. I am also indebted to my parents-in-law, Serap and Naci, who always extended me with a second life – my lovely wife – and extended their support whenever I needed. Last but not the least, I thank my siblings Ahmet Abi, Ayça Pelin, Elif, Zeynep, Ömer Faruk, and especially my best friend since childhood, my elder sister Elif Kübra for being the source of my happiness throughout my life.

I feel obliged to my thesis adviser Assist. Prof. Hatice Tekiner Moğulkoç, for her useful remarks, support, and guidance throughout the thesis writing process. No phrases suffice to express my deep regard for Prof. Erkan Türe who steered me to academia as a role model. Besides them, I also thank the rest of my thesis committee: Assist. Prof. Nuri Cihat Onat and Assist. Prof. S. Tankut Atan for their insightful comments and encouragement. Last but not the least, I acknowledge the infinite kindness of İstanbul Şehir University School Of Languages English Preparatory Program administrators, especially Manager İlke Büyükduman, for providing me all kind of assistance.

I am also grateful to my dear brother Abdul Basit Adeel for his unremitting assistance in proofreading this thesis. The same goes for my cousin Muhammed Furkan Şahin who facilitated software development process for this study. I also appreciate the empathy of my immediate friend circle, whom I call "Göy Göy Ailesi". Without their moral support, I would have lost my motivation.

Lastly, I am truly beholden to my dear wife, Sena, who has always inspired me to be a good human being. Her patience during the thesis writing process deserves unlimited praise. Thank you for being in my life, Sena. I love you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Timetabling problems are applied in a wide spectrum of areas including preparation of weekly course or examination timetables in primary schools, high schools and universities; scheduling transportation activities in airway transportation; scheduling personnel shift schedules in health sector; and scheduling shifts in production and service operations.

Timetabling problems require that certain time intervals are appropriately assigned to related entities, while taking into account certain constraints. Information and communication technologies advancing at a fascinating pace over time provide great convenience to researchers and practitioners in solving such problems. Increasing information processing capacities shorten the resolution process and facilitate analysis of the results.

One of the most frequently studied timetabling problems is in educational field. The educational timetabling includes preparing timetables, making room assignments, students sectioning for educational institutions and is described as NP-complete which cannot be solved in polynomial time. The solution of educational timetabling problems, which occurs at least twice per semester, requires intensive labor and valuable resources. The complexity of problem concerning the preparation timetabling in educational institutions is increasing over time. With the increasing enrollment of students at unprecedented level as well as diversification of lectures and inclusion of special constraints that vary from one institution to other, the problem of preparing weekly timetable for the practitioners becoming more challenging.

A number of methods have been developed to solve the problem of preparing weekly timetable in educational institutions along with many computer software that have been developed employing these methods. However, such software has limited application potential due to differences among educational institutions. In other words, any software serves particular needs of certain institutions, and consequently lose their wide scale applicability. Hence, many educational institutions are still engaged in preparing timetables manually by employing various computer based procedures and programs like MS Excel.

This thesis focuses on class-teacher timetabling problem – is a sub-field of educational timetabling – and includes assigning pre-assigned instructor and classes in available time slots and group balancing problem is aimed to diversity within the classes and balanced assignments between the classes.

The purpose of this study is to develop mathematical models for preparing class list and weekly timetables, and to design a user-friendly decision support system for related problem for reducing the time spent in the preparation period. A mixed integer based mathematical model for preparing class lists and a binary integer based mathematical model for preparing weekly timetables are developed, and solved by using GAMS software package.

A web-based decision support system (DSS) is designed to assist for solving the problems that are discussed above. Both the user's preferences as well as the front-end interface is included in the DSS. The former allows user to input certain parameters, while the latter allows data entry to the database. The model base that contains mathematical models to solve the related problems is also included in DSS. The web-based system assists the end-user to solve problems and create reports on-line.

We used data of İstanbul Şehir University School of Languages English Preparatory Program (SEPP) to demonstrate mathematical models and operation of the system, and the outcome is compared with manually generated timetables as a case study. We solve the problems in a minute. According to SEPP, it takes around two days to prepare timetables manually.

In the following part of this thesis, literature review and background information on educational timetabling problems is provided. In the third part, the problem at hand is defined, and both decision support system and mathematical model are explained. In the fourth part, case study will be provided. In the last part, the outcome of our research will be evaluated.

Chapter 2

Literature Review and Background

Assignment problems are among the most common problems in the Operations Research. In assignment problems, it is desirable to find individual person-to-task match that will provide the lowest total cost.

Many different types of assignment problems exist in the literature. The problems include stable marriage problem; generalized assignment problem; Hungarian method, linear bottleneck assignment problem; quadratic assignment problem; linear bottleneck assignment problem; scheduling and timetabling. The class-teacher timetabling problem is considered in this thesis.

The objective of class-teacher timetabling problem is to determine the required assignments according to the limited resources at a specified time interval, to satisfy some predetermined rules (constraints).

Each timetable problem deals with unique set of constraints. Generally, these constraints are divided into two classes: hard and soft. Hard constraints are the constraints that cannot be violated in any way - for instance an event is not assigned to more than one source at the same time. On the other hand, soft constraints are conditions which are not mandatory but favorable for a good timetable. The timetable is enhanced as these constraints are met. Soft constraints are the constraints that are not necessarily bound by consecutive events concerning a common person or resources, but are desirable if available. Minimization of soft constraints that does not satisfy is intended to provide the objective function of these problems or to satisfy all constraints.

In the literature, timetabling problems are evaluated as NP-Complete. NP-Complete is the class of both NP and NP-Hard problems. Therefore, the problems are the hardest problems in this NP class. If any of them can be solved at the polynomial time, as mentioned in definition, all of them can be solved at the polynomial time.

Educational timetabling problems – the widely applied sub-problems the timetabling problems – that includes preparing timetables, making room assignments, students sectioning for educational institutions. The initial studies in educational timetabling took place in 1960s, but according to Bardadym [4], the interest decreased in 1970s. However, it later on regained prominence in 1980s and reached its zenith in 1995 on its 60th anniversary. As a milestone, the first International Conference on the Practice and Theory of Automated Timetabling (PATAT) conferences was inaugurated that year [5, 6]. Studies on educational timetabling problems have not lost their relevance since then [7].

Many studies in the literature deal with the constraints reflecting the priorities of certain institutions. This illustrates that educational timetabling problems can be used in all educational institutions including universities, colleges, and schools. However, the definition and the terminology of these problems may vary from institution to institution.

The hard constraints that must be provided is generally defined as follows:

1. Any resources (student, instructor) cannot be assigned more than one activity at the same time slot,
2. For each time slot, sufficient resources must be allocated to all activities assigned to that time slot.

Burke [8] has compiled various soft constraints as follows:

1. A course can be assigned to a predetermined time slot.
2. Assigning the students' courses to consecutive time slot during the day can be prevented.
3. The course of the instructors can be assigned to specific days. Thus the instructor may choose to leave other days of the week empty.
4. Depending on the wishes of the instructor, courses may be assigned to specific classrooms.

Different approaches are utilized to classify Educational Timetabling into sub-problems. Werra [9] investigated two sub-problems named Class-Teacher Timetabling and Course Scheduling to illustrate the solution of both through graphical method. Petrovic and Burke [10] investigated university timetabling problems under two sub-problems as course and examination.

Bardadym [4] introduced a classification as Faculty Timetabling, Class-Teacher Timetabling, Course Scheduling, Examination Timetabling, and Classroom Assignment. Another classification only three main groups was developed by Schaerf [11]:

1. School timetabling involves weekly timetabling for classes at a given institute (generally primary and secondary schools) in a manner that no two classes given by an instructor clash with each other.
2. Course timetabling involves weekly timetabling for lectures of set of courses given at a university by avoiding possible overlaps of lectures involving common attendees.
3. Examination timetabling involves timetabling for the exams concerning a set of given at a university by avoiding overlap of exams involving common participants, and ensuring the inclusion of maximum number of students in the exams.

Carter and Laporte [12] classified educational timetabling problems into five sub-problems:

1. Assigning instructor only to classes without considering time and location of the class [13].
2. Timetabling only the instructors to give lessons considering the condition that an instructor cannot give more than one lesson at the same time. One of the example studies in which instructors are assigned to classes and classes to time periods is conducted [14].
3. The problem that deal with the assignment of courses to time slots and classrooms [15].
4. A student scheduling problem that deals with the scheduling of lessons to overlapping with the same period [16].

5. Classroom assignment problem that deals with the assignment of courses to time periods considering special requirements such as seating capacity, necessary equipment [17].

As illustrated by the problems described above, the main components in educational time timetabling problems are: class, classroom, time slot, instructor, and student. Any problem encountered in the literature or in real life can be solved by addressing the above mentioned components, or any or a few significant combinations in different orders.

Handling all components simultaneously often yields infeasivle solutions. Schaerf [11] also demonstrated that the problem is NP-complete, so that the best solution can only be obtained for small-sized problems as described in his 1963-1999 work. Dinkel [18] noted that in most of the studies conducted at that time, the number of courses needed to create the problem size and timetable complexity increased with the increment in the number of principal components (mentioned above).

In the following subsections, each problem will be explained in detail.

2.1 Course Timetabling

Course Timetabling Problems, which are also described as University Timetabling Problems, or Course Scheduling Problems in the literature, are intended to assign courses, instructors, classrooms and time slots by taking into account the availability of instructors and classrooms, number of students, and classroom capacities.

In course timetabling problems, the problematic items (student, classroom, etc.) are not fixed. For example, the student group including a student does not have to take or be responsible for the same courses in the same semester – individual schedule of students can vary. Similarly, an instructor does not have to give the same course, and it is possible to assign him/her to any course in his/her own field. For these reasons, course timetabling is more complex and bigger in size than other educational timetabling problems.

Bloomfiled and McSharry [19] assigns classes to classrooms and time slots in two stages taking into account the preferences of the instructors. In a similar research, Glassey [17] addressed the assignment of courses to classrooms and time slots taking into account specific requirements such as classroom capacity, necessary equipment.

Al-Yakoob and Sherali [20] solved the problem of preparing the curriculum of Kuwait University using mixed-integer programming and compared the method with the manual program.

Cura [21] used the simulated annealing algorithm in his study of the problem of preparing the curriculum of Istanbul University Business School. Unlike other problems in the study, he also considered the seniority of the faculty member.

Specific practice of Sarin *et al.* [22] aimed at minimizing the total distance covered by the instructors from their offices to classrooms.

The study applied to Gazi University Department of Industrial Engineering, has developed two new mathematical models for solving the problem. In the first phase of the application, undergraduate courses are scheduled, and the second phase various compensatory course scenarios are created with the help of the mathematical model developed in the first phase [23].

Several other studies related to this section could be found in the literature [24–27].

2.2 Examination Timetabling

In examination timetabling problems, the aim is to plan the exams according to the specified time slot while minimizing the overlaps for the students. Examination timetabling is similar to course timetabling, and it is difficult to distinguish them from each other on the basis of employed methodology. Due to this similarity, some special cases can be modeled utilizing the same methodology. According to Schaerf [11], the differences that distinguish examination timetabling from course timetabling are as:

1. Every course has to have only one exam.
2. The overlap is usually avoided, because is not desirable for the group of students who receive the same course to face a conflict situation (or having different exams in the same time slots).

3. One exam appointment can be made for each student, and no consecutive exams can be held.

4. Fixed time slots do not have to be used as in Course Timetabling.

5. Multiple exam planning can be done in the same classroom.

Examination timetabling problems are a highly complex combinatorial problem involving NP-complete sub-problems, and it is described in various forms in the literature on the basis of constraints and objectives [11, 12]. An examination timetabling that provides the full range of hard constraints is considered an appropriate solution.

When the exams are assigned to classrooms and time slots, the size of the problem grows considerably. For this reason, the solution approaches employed divide the problems to solve them in appropriate time [28]. In this process, the examination problem is divided into two sub-problems to be solved respectively. In the first sub-problem, the examinations are assigned to the time slots considering the total classroom capacity instead of the capacity of each classroom for the relevant time slot. In the second phase, exams, which have been assigned to time slots, are now assign to classrooms on the basis of capacities, and relevant needs.

Marti *et al.* [29] modelled a multi-objective problem for proctor and final exam assignment in a university. The complexity of the problem increases with the addition of time slots preferences of the proctors. There is an upper limit work load for each proctor. The limit of workload is determined by the administration.

A study proposing the best solution for a selected application problem in Eskişehir Osmangazi University has developed a multi-objective mixed integer model to perform balanced assignment of proctors while minimizing assignments made to undesired time slots, taking proctor preferences and constraints into consideration. They also have taken proctor preferences and constraints in their work through a web-based decision support system [30].

Al-Yakoob *et al.* [31] have handled examination timetabling and proctor assignment sub-problems. They developed a mixed integer programming model for the proctor assignment system at Kuwait University. In this study, day and time preferences of proctors were also included in the model.

Kahar and Kendall [32] proposed a mathematical model for the examination timetabling that takes into account by capacities of classroom at University of Malaysia, Pahang.

In a study conducted at Pamukkale University, researchers have developed mathematical model based on the idea that the exams with student achievement scores are positively correlated with the adequate preparation and resting version. For this reason, the basic objective is to maximize time between the exams (i.e. paper spread) considering the challenges of exams. Two distinct genetic algorithm models were developed to optimize paper spread [33].

Several other studies related to this section could be found in the literature [34–38].

2.3 Class-Teacher Timetabling

The Class-Teacher Timetabling Problem – also described as School Timetabling Problem in the literature – requires the creation of the best possible weekly timetables, considering the specific constraints of teachers, classrooms, and courses. The primary aim is to produce class-teacher combinations in accordance with the classrooms and time slots. Unlike course timetabling problem, the lesson that teachers will give and students will attend are known beforehand, such that the lessons teachers will give are known-before; and a student group is obliged to take the same courses during the same semester, and the basic objective is to assign the lessons and the teachers to the appropriate time intervals under the basic constraints that classrooms are fixed (i.e. any student group is always in the same classroom), while avoiding any free lesson slot between any two courses.

As the teachers and their branches are known in advance, the process through which the class is coupled with the instructors at appropriate time slots, constitutes the problem. Assigning instructor depends on the variables like the availability at given time, nature of employment (part-time or full-time), or assigned courses type (depending branches). There may be other constrains of similar nature in the model.

As we mentioned in previous sections, class-teacher timetabling problem is a sub-problem of educational timetabling problems. Solving class timetabling problem involves allocation of course, classrooms, and teachers to relevant time slots to satisfy the hard and soft constraints of the problem. Students are generally known before the timetabling process. Class-teacher Timetabling differs for each country due to the content of particular education system. Moreover, it may also be different in distinct educational institutions in the same country. In some cases, the classroom is not included in the course, and each class should be assigned to a pair of teachers, a time period, and a place. The number and duration of courses may vary for classrooms or grade levels in the school. Moreover, some schools may offer training at multiple campuses or buildings. Relevant studies conclude that the field of educational timetabling is unable to solve all possible instances of high school timetabling [11, 39–43].

Carmusciano and Cardillo [44] developed a mathematical model employing an approach that uses a combination of annealing simulation and tabu search, because the objective function is not linear and the size of the problem is too large, by establishing the decision model of the class, tutorial, and time slot assignment problem.

Kwok *et al.* [45] surveyed 480 middle schools in Hong Kong and investigated the problems encountered school timetabling in Hong Kong secondary schools to identify a general structure. As a result of the survey, they emphasized the importance of automatic timetabling defined in terms of computer-generated timetabling and the necessity of further research on this topic.

Kong and Kwok [46] modeled a high school timetabling problem with a knowledge-based approach and produced an heuristic based method for solving this model.

The objective of Veenstra's study [47] is to define methods for effectively solving the problem of school timetabling under disturbances, and present three different solution methods: a simple rule-of-thumb, a heuristic, and an optimization approach. It has been solved with unique data from five high school in the Netherlands.

The other studies related to this section could found in the literature [48–53].

2.4 Solution Approaches

Different approaches such as mathematical programming, heuristics, meta-heuristics, graph coloring and constraint programming are used to solve educational timetabling problems.

During the initial development of timetabling problems, the solutions were reached manually. The prevalent approach course with maximum constraints is assigned in the beginning, which is followed by other courses in a descending order. These techniques, which generate timetables by producing partial solutions, are called heuristics [11].

Heuristics (also described as heuristic rules, heuristic method) is the use of any rules, strategies, tricks, simplifications, and other factors that precisely limit the search for solutions when the problem space is too large. Therefore, heuristics is the key to finding the solution when the problem involves complexity.

Caster examined these approaches in the following four groups [54]:

1. Cluster methods
2. Sequential methods
3. Meta-heuristics
4. Constraint-based approaches

Petrovic and Burke [10] added multi-criteria approaches, case-based reasoning techniques and hyper-heuristics to this category.

2.4.1 Mathematical Programming Based Approaches

If all or some of the decision variables are assigned an integer value in a problem, such problems are called Integer Programming Problems. Depending on the distinction based on whether the decision variables in a problem have integer values or not, the approaches are classified as integer programming; pure integer programming, mixed-integer programming; and binary integer programming [55].

Pure Integer Programming is the type of problem in which all decision variables receive an integer value. The employed decision variables are; people, machinery, spare parts, vehicles and so on.

Mixed-Integer Programming is the type of problem containing some of divisible decision variables and some non-divisible variables.

Binary integer programming is type of problem associated with or determined by "yes" or "no" decisions. In such problems, the decision variables take binary value: 1 or 0, depending on whether the decision is yes or no, or whether the task is assigned or not.

Integer Programming as a method has been employed in Educational Timetabling Problems since the beginning. Decision variables in the binary structure that form the basis of the Educational Timetabling Problem model can create problems in the solution of large-scale problems. However, advances in the contemporary computer technology have re-popularized mathematical solution techniques.

Akkoyunlu [56] and Csima *et al.* [57] have addressed the problem of assigning lectures to time slot by employing a linear programming model. Akkoyunlu has developed a linear program that assigns cources to time slots for only one department in college and assumes that there are enough classroom capacities. This program has yielded the best overall solution. Lawrie [58] modeled the timetabling problem as an integer programming problem with linear constraints and no objective function.

McClure *et al.* [13] and Tripathy [59] have described timetabling problems as a linear integer programming problem. However, all these techniques can only be applied to small-sized problems. The manual usage of integer optimization techniques becomes more difficult as the problem size grows.

Dimopoulou and Miliotishave [60] addressed the problem with a constraint-based model in their study that assigns classes to classrooms and time slots.

Ozdemir *et al.* [61] have solved the assignment of course-instructor with a multi-objective structure that takes into account the preferences of the administration and instructor, and solve it with an integer model. The constraints in this study are the weekly work load of each instructor that is between certain limits and each course is assigned to only one instructor.

Ismayilova *et al.* [62], considered the problem of assignment of course-instructor-time slots as multi-objective, considering the preferences of administration and instructor.

Al-Yakoob *et al.* [63] have developed a mixed integer model for solving the timetabling problem at Kuwait University. Boland *et al.* [64] provide an integer linear programming model to solve the high school timetabling problem and class blocking problem. Ozturk *et al.* [30] developed a mixed integer model for solving the supervisor-exam assignment problem.

Ribiu and Konjicija [65] used integer programming while solving the school timetabling problem in two stages. While the first stage focuses on day placement, the second stage solves the other part. The approach has been tested on real scenarios.

Kristiansen *et al.* [66] took the desires of the students into accounts and aimed at maximizing the satisfaction or minimizing the violation of hard and soft constraints while assigning students to elective classes .

Sørensen *et al.* [67] applied a two-stage decomposition approach to an IP model formulated for a high school timetabling problem in Denmark.

There are other studies using the mathematical programming technique in the literature [64, 68, 69].

2.4.2 Graph Coloring Techniques

The basic timetabling problems are similar to the vertex coloring problem in network theory. In an educational timetabling problem, the network is created in the following way:

1. Each course or exam is represented by a corner.
2. If any two courses or exams have at least one counterpart, the relevant two corners are combined at one edge. Two corners (course or exam) combined at one edge means into using a common source. Therefore, two courses or exams cannot be in same time slot.

Different colour is used to represent every distinct time slot. Thus, a grid is created such a way that any adjacent (combined at one edge) of the corners does not have the same color. The relationship between graph coloring and a simple timetabling problem is given in Figure 2.1.

If examination timetabling problem has additional constraints such as classroom capacities, pre-assignments, and sequence, the graph coloring problem changes. Hence,

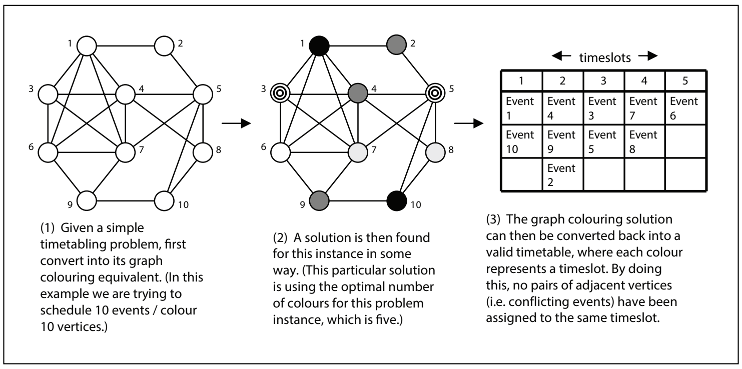


Figure 2.1: Demonstrating the relationship between the graph colouring problem and a simple timetabling problem where only the event-clash constraint is considered [1]

different graph coloring algorithms have been developed such as "largest degree first". The basic operating principle of all these algorithms is to handle the most difficult events – primarily timetabling [70–72].

Graph coloring based techniques produces good results for small-scale problems, but they are not successful for large-scale problems, and changes cannot easily be reflected in the model [73].

2.4.3 Heuristic Techniques

One of the most used algorithms in computer science is heuristic algorithms. These algorithms yield uncertain outcomes. They either do not display same performance or guarantee to always yield the desired results, but they are still considered useful algorithms for optimization. In other words, heuristic algorithms do not guarantee the best result, but they guarantee that they will provide a solution within an appropriate time period. They usually arrive at the good solution quickly and easily.

There are many different heuristics in the literature, and the most common ones are as follows on educational timetabling problem:

1. Annealing simulation
2. Tabu search
3. Genetic algorithms
4. Ant Colony Algorithm

For instance, in a problem including specific hard constraints such as exam that has more than 500 students are in the morning session, Metlot [74] employs external constraint-based algorithm to identify the annealing simulation algorithm to be used for the initial solution.

In annealing simulation algorithms, different strategies are applied in the search process. For example, Bullnheimer [75] found a way to switch between the time slots as well as the exams, while Burke's technique [76] depends only on switching exams.

Alvarez-Valdes *et al.* [16] developed a two-stage tabu search algorithm for course-student assignment, which is not directly timetabling problem, but a sub-problem of the timetabling problem. In the first step, the best solutions are obtained for each student with a balanced assignment so that student preferences are satisfied. In the second stage, all the solutions are brought together to create a satisfactory balance between the courses.

Burke [77, 78] used genetic algorithm for examination timetabling and also a graph coloring algorithm to obtain an initial solution and a combination of a local search procedure and an evolutionary algorithm to find the problem of examination timetabling. The parallel GA developed by Abramson and Abela [79] for solving the timetabling problem has also considerably reduce the problem resolution time.

Rossi *et al.* [80], Azimi [81], and Eley [28] have used ant colony algorithms for timetabling problems. In a similar approach solved for examination timetabling problem with the ant colony algorithm developed based on graph coloring algorithm [82].

2.5 Novelty and Motivation

Although successful results have been obtained from timetabling studies, an absolute absence of interface is the biggest obstacle to the practical use of these studies for the end user, which reduces the practical application of the solutions. For this purpose, meaningful data must be fed to the mathematical model that produces solutions to the timetabling problems. However, end user is unable to feed such data due to high complexity of this task.

The exlsuive feature of this study is a web-based decision support system designed for solving the relevant developed mathematical models for end-users. A decision support system (DSS) enables users to work with various data sources, modeling techniques, and store information using user-friendly Graphical User Interface (GUI). The user does not need programming or mathematical expertise to use this system.

2.5.1 Decision Support System

A decision support system (DSS) is a model-based or knowledge-based system designed to support managerial decision making in semi-structured or unstructured situations [2, 83].

Decision support system (DSS) combines user requests and computer functionality. It takes user's data through simple Graphical User Interface (GUI), converts them meaningful data, and utilizes them to help the user make a decision. DSS is useful in the administration of almost every phase of operations. During the decision phase, it aims extracting useful data from non-regular data.

According to the definition proposed by Sprauge [84], a system must carry the following four basic features in order to be a DSS:

1. DSS is mainly concerned with resolving problems faced by senior executives in poorly structured environments
2. DSS generally aims to solve problems by using traditional data access and processing methods.
3. DSS allows non-computer specialists to communicate easily and interactively with data.
4. DSS should be flexible in order to adapt the changes in the field to itself.

DSS can be classified in two different groups or kinds: Knowledge Base and Model Base. The schematic view of a Decision Support System are shown in Figure 2.2:

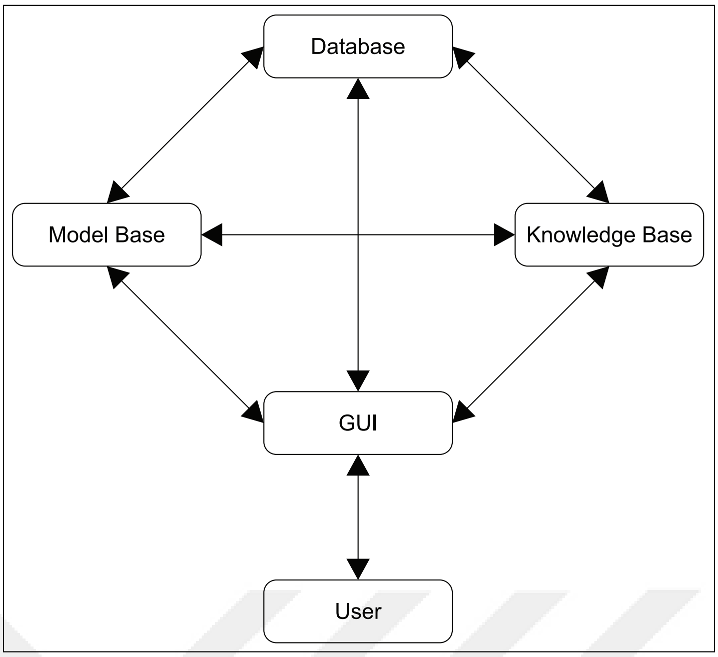


FIGURE 2.2: Schematic View of a Decision Support System [2]

A decision support system (DSS) has 5 main components [83]:

1. Database: providing data on which decisions are based.

2. Model Base: analysis capabilities is provided by components like statistical, financial, optimization, or simulation models in this base.

3. Knowledge base: generally enabling the storage and retrieval of previous knowledge of the subject.

4. GUI: covering all the communication in User and DSS application. For this reason, it may be the most important component of DSS.

5. User: the person who uses DSS to support the decision-making process.

The concept of decision support is based on the theoretical work carried out at the Carnegie Institute of Technology in the late 1950s and early 1960s on organizational decision making, and later during the 1960s [85].

In 1973 the first prototype of DSS Generator emerged [86]. Prior to the advent of the World Wide Web, DSS Generator was spreadsheet-based. In the mid-1990s, the first web-based decision support system software (DSS Generator) emerged [87]. Currently, there are many DSS Generator products (mostly web-based) [88].

According to Sol [89], the definition and extent of DSS varied from year to year:

1. In 1970, DSS was defined as a "computer-based system to help decision-making".
2. In the late 1970s, the DSS started to be associated with "computer-based systems that help decision makers use databases and models to solve poorly structured problems."
3. In the 1980s, the DSS was referred to as "systems that use existing technology to enhance the effectiveness of administrative and professional activities."
4. By the end of 1980s, DSS was considered the design of intelligent workstations.

DSS's status as a communication platform of utmost significance and as an information source increased rapidly from the early 1990s to the World Wide Web (WWW) [90]. As a result of the development of web technologies, it became possible to use programs – otherwise used by experts – without any particular expertise. Web-based applications can be accessed from any computer or any location. Thus, complicated processes like solving optimization problems can also be done by ordinary users – once DSS is established. Also, instead of installing the required software and programs on all computers, the software and programs installed on a single server can be updated by an authorized person, who provides access to rest of the users.

The use of computers and other technological devices in the creation of timetabling has become widespread to satisfy the needs of users in the education systems (instructors, students, etc.), and to allow the administrators to take the system as a whole, and to improve the performance of education system.

When necessary data is entered through web-based decision support system, which has been discussed in this story, the solution to the related educational timetabling problem will only be a click away. Moreover, web-based system will be readily accessible from any device or any location. The overall system will not only make the task easy, but also save the valuable time for the end user.

The basic components of a DSS are shown in Figure 2.3:

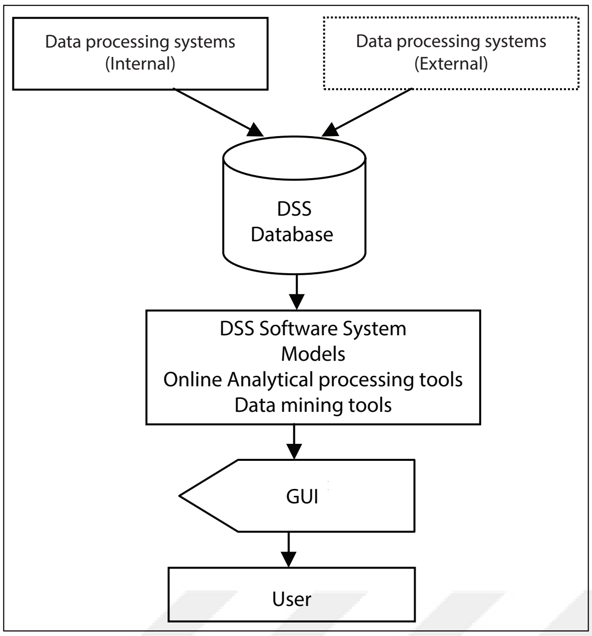


FIGURE 2.3: The basic components of a Decision Support System [3]

Decision support systems specific to the institution's needs to generate timetables for distinct educational institutions have already been developed.

Miranda's eClasSkeduler [91] is a decision support system based on mathematical models consisting of four components: user interface module, input information module, optimization module, and report generation module. The input information module is a database storing all information about courses, classrooms, and instructors. The user interface module facilitates DSS operations at user end. The optimization module produces timetabling solutions based on mathematical models. The report generation module produces administration reports on timetables, and availability of classrooms.

The SlotManager has similar components. It has three components, the interface, the database, and the algorithm engine. Its interface is for data input and report management. Its database contains the details of created timetables. The algorithm engine determines block-schedule conflicts, instructor schedule conflicts, and provides available resources for creating timetables [92].

Miranda *et al.* [93] have developed a web-based decision support system that uses an integer programming model to generate appropriate timetables for classroom and classroom planning. Unlike Miranda's previous work, the decision support system is housed at a server, where instructors have the ability to directly enter data to collect and retrieve specific data.

Şuşnea [94] claims that universities have become dependent on collection, storage and processing of educational data. An intelligent decision support system is proposed to produce meaningful data and improve decision-making (which is expected to maximize the performance of universities). The study explains a three-component system; a data management system, a model management system and a user interface.

Chapter 3

Problem Description

The main aim of class-teacher timetabling is to meet instructor combinations with the classes in available time slots. To simplify, it generates the weekly timetables of teachers and classes. The general hard and soft constraints of class-teacher timetabling problem has to satisfy is as follows:

1. Every instructor cannot be allocated to more than one lesson in same time slot,
2. Every class cannot be allocated to more than one lesson in same time slot,
3. Every instructors cannot be allocated at lessons that are not available for the instructor,
4. All instructors have to fulfill their weekly workload,
5. Every instructor cannot conduct more than two daily lesson hours with same class,
6. The lessons should be consecutive (3 lessons are conducted by the instructor in a row).

The preparing class list problem is dividing students owning various attributes (gender, international status, repeat status, and score) into a certain number of classes, so that classes are as balanced as possible with respect to the number of students owning each attribute.

As discussed in the previous chapters; each problem has its own unique constraints and requirements. In this study we focused on solving the problem of ŞEHİR English Preparatory Program (SEPP).

ŞEHİR English Preparatory Program (SEPP)'s curriculum consists of five terms spanning seven weeks each. Moreover, each term contains five modules. These modules are called Elementary, Pre-Intermediate, Intermediate, Upper-Intermediate and Pre-Faculty, which correspond to a certain level of English learning. There are 5 modules in every semester.

Opening of a module and the number of relevant classes are determined as a result of the module final exam that is held at the end of each term. Once one term has finished, the next terms must start by the first working day of the next week. Thus, SEPP management faces the problem of preparing the weekly timetables and the relevant class lists within a very short period of time, which increases stress and workload. For example; module final examinations are held on Friday, the preparation and announcement of class lists and weekly timetables must be announced on Sunday, or on Monday morning at the latest.

Approximately 50 classes are planned each semester. There are 2 types of courses – Lecture and Tutorial – in every class of each module. The number of classes to be opened and the number of expected students in the classroom varies depending on the particular period. Each course consists of 50 minutes long sessions and the sequence is requested according to the patterns to be created by the SEPP.

There are approximately 80 instructors in full-time or part-time status at SEPP. While part-time instructors are usually appointed as support instructors, full-time instructors can both be main and support instructors. Both the number of the instructor and the availability of the instructor can vary from term to term.

There are approximately 650 students in each term. When the classes are being created, gender of students, grades they receive in the final exams of the module, nationality they belong to are taken into account. SEPP seeks to provide students with an equal opportunity learning environment by creating classes where grades, gender, nationality and other personal attributes of students are not significantly dispersed.

Even though SEPP education has no direct effect on the undergraduate GPA of the students, the language proficiency of the students after finishing English preparation is nonetheless is directly related to the design of the class. Therefore, the balanced distribution of the classes and the objectivity of class appointment process are very significant in this process.

In short, the requests for preparing weekly timetables and class lists by SEPP are as follows:

1. 8 hours of planning every week day – except in the case of unusual exceptions.
2. There are 2 types of courses in each module: Lecture and Tutorial.
3. Two type of instructors are assigned to each class: Main instructor and Support instructor.
4. Both support instructor and main instructor can be assigned to lecture courses.
5. Except Friday, the last lesson must be a tutorial.
6. The main instructor must conduct all tutorials.
7. The support instructor must conduct one of the 4 tutorials.
8. Some instructors (usually full-time instructors) should conduct a course on every day.
9. The workload for instructors varies depending on the classes they teach, and each instructor must fulfill the assigned workload.
10. Courses are not held on Monday's first lesson and Tuesday's last lesson because of SEPP meetings.
11. In the same manner, there is no course after 12 pm on Friday.
12. Lecture lessons are required to be in a sequence shown in Figure 3.1. This configuration does not include tutorial sessions.
13. The average grades of created classes must not be significantly different from each other.

14. The number of male and female students assigned to classes must be same. Assignment of only one male or female students is undesirable.

15. In the similar fashion, the number of international students assigned to classes must be same. Assignment of only one international students is undesirable.

16. Repeat students (who have not been successful in the previous term) must also be equally distributed among the assigned classes.

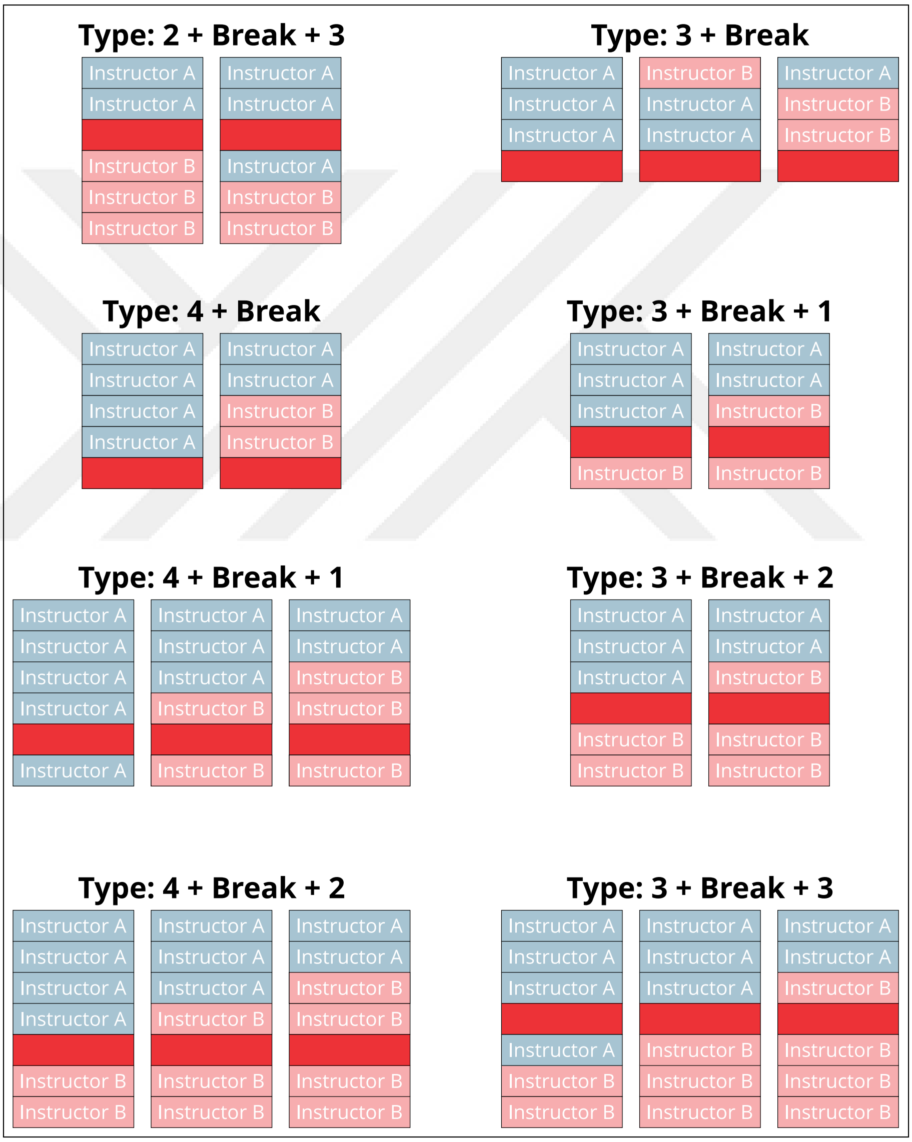


Figure 3.1: Types of daily configurations

Before the implementation of this study, timetabling and preparation of class lists was done by a three-person planning team and a coordinator. The timetabling process was done manually, and the team used the previous semester timetabling as a base format. Due to excessive constraints, limited time to prepare, and the condition that each case must be distinct has made this process difficult.

This study is a decision support system based on mathematical programming models [95]. It produces optimum class lists and appropriate weekly timetables taking into account all the constraints and demands of the SEPP by using a web based platform. Due to the existence of two distinct and independent objectives that are creating student lists and timetables, the problem is divided into two components. Each objective would be reached through formulation of different models.

The architecture of the study consists of five main modules depicted Figure 3.2.

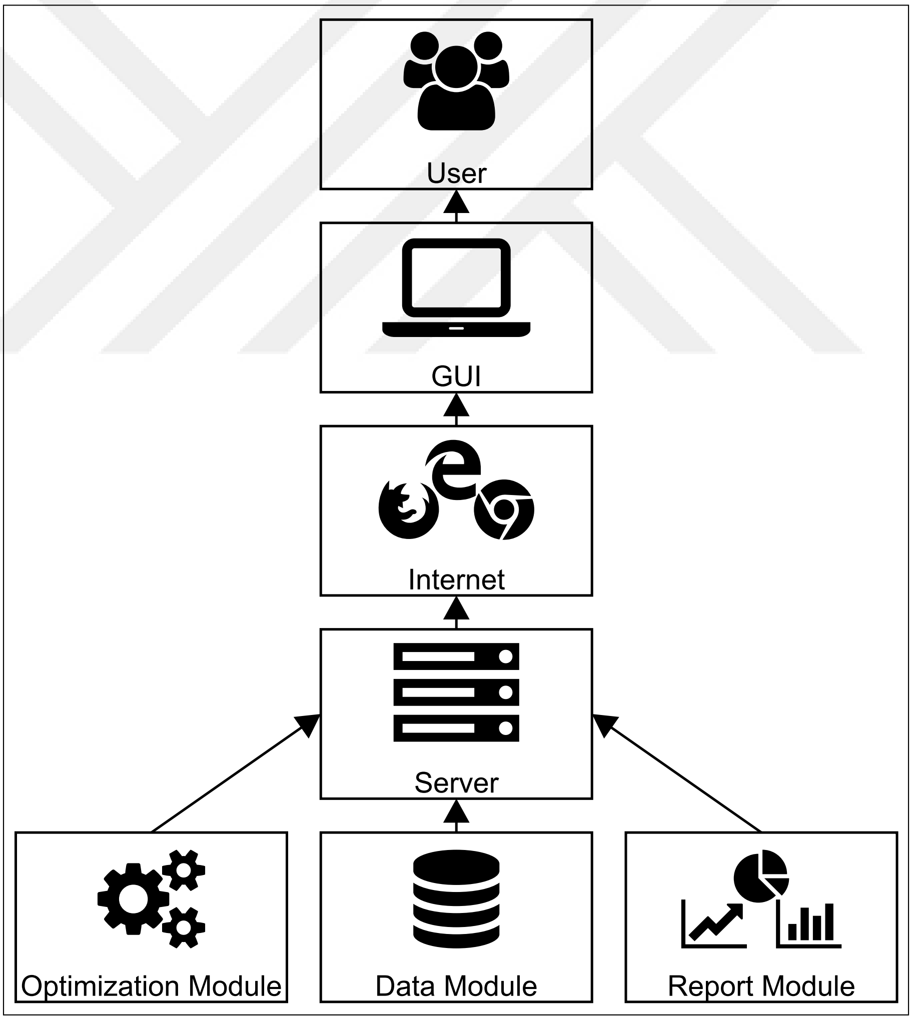


Figure 3.2: Schematic View of the Decision Support System

The first is the User Interface module where the parameters of the system are set and the outputs are displayed to manage the information. This interface allows users to enter type of lecture and tutorial courses, availability of instructors, as well as other parameters related to the instructors. In a similar fashion, users can enter student attributes to prepare class lists. Users can also view the created report, output is printable, and can be saved in different file formats.

The second module, data module stores weekly timetables and all information needed to create classes in the relational database. To elaborate, some of the information stored in this section, include instructor and class availability, pre-generated instructor pairs, and types of instructors. Previously created projects are also stored on the database. In the designed system, different users can be defined and separate database records for each user are kept. The architecture of the relational database is shown in Figure 3.3.

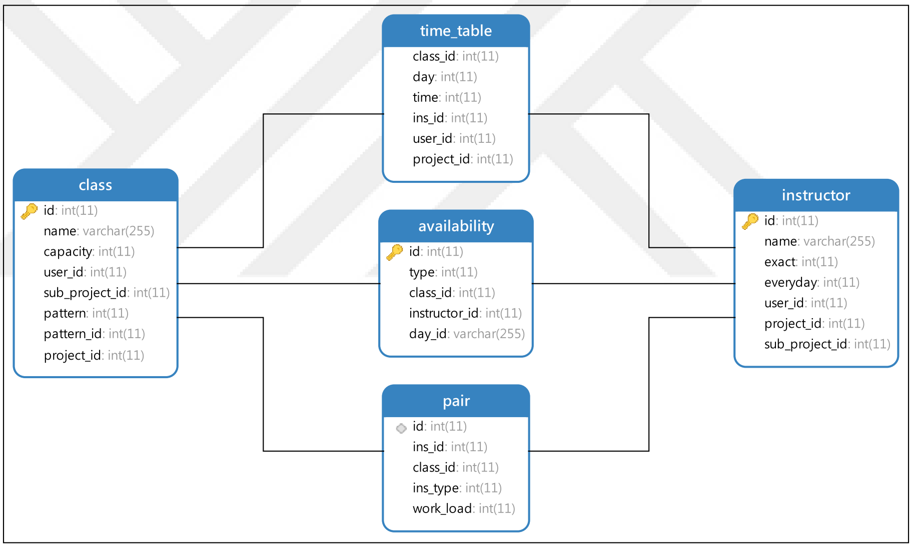


FIGURE 3.3: Architecture of the database

The third module is the optimization module that contains the code of binary integer programming model for weekly timetables and mixed-integer programming model for preparing class lists. The models have been solved using GAMS software package.

Finally, the fourth module is the report module that creates a progression of administration reports after every execution of the optimization module. It can present reports for evaluation of the results produced by the optimization module, such as the number of students in the class according to nationality or gender, the overall grade average of the classes, and timetable for classes or instructors.

PHP (Hypertext Pre-processor) is used as a programming language to develop the web based platform. PHP is a server-side, widely used, general purpose, scripting and programming language embedded in HTML. This language was developed by Rasmus Lerdorf in 1995, which is now maintained by the PHP community.

The PHP code is interpreted by a web server with a PHP processing module and a web page is generated as output. These codes can be embedded into the HTML code as well as saved in an external file to process the data. It is comfortable and fast coded, and does not need to be compiled. Code writing is similar to popular programming languages such as Perl, C, Javascript, etc. Hence, anyone who is aware of these languages can easily encode PHP.

PHP and SQL have become an integral duo because PHP works very efficiently with the MySQL database. This duo is easy to operate and secure to store information. For this reason, while creating this system, MySQL database and PHP software are used.

MySQL is a multi-threaded, multi-user, fast and robust database management system. Multiple users can manage MySQL data over the web. With MySQL, people can access the data through Internet using the user interface. The advantage in MySQL's basic design is speed, security and stability. MySQL is easy to manage when compared to other database software. The database can be created with a few commands, and it is compatible with the data.

The problem is divided to two separate components. The first part of the system is concerned with preparing class lists and the second part is concerned with generating timetables.

3.1 Preparing Class Lists: Karmalama

The problem is given a special name in Turkish by SEPP: "Karmalama". SEPP prepares student lists and module final exam scores list using Excel spreadsheet. The user should enter the student's name, student number, gender, nationality, repeat status, final exam score and number of required class. After the project has been created, it is possible to enter data in two different ways in the Step A1 (Figure 3.4): (1) By copying and pasting the data from the specific table into the given input area (2) Upload File: the web based platform produces pre-prepared list draft on request that can be downloaded by users, filled in with information, and re-uploaded to the platform to feed the data into the system.

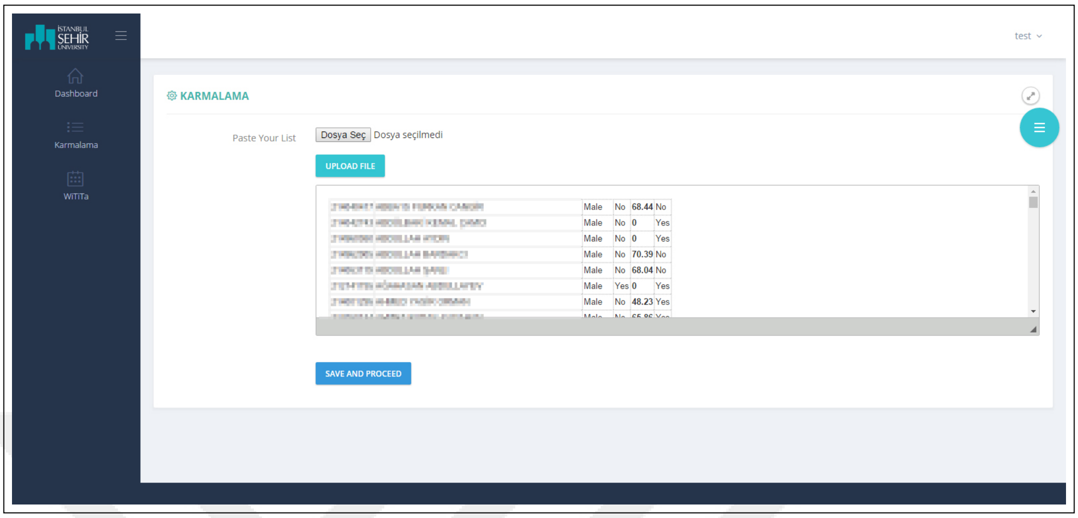


FIGURE 3.4: Screenshot of Karmalama: Step A1

The Step A2 (Figure 3.5) can be defined as checkpoint as user is allowed to review the copied/pasted data (first method in the Step A1) – and select certain row/columns or make changes in case of any discrepancies. For instance, if user fed a table containing 150 rows and 8 columns; and the useful data is provided in only first 6 column. The reversion to Step A1 after uploading the data will allow user to select those particular rows and columns. On the other hand, if user has fed data through upload file method (second method in the Step 1), this step will be skipped.

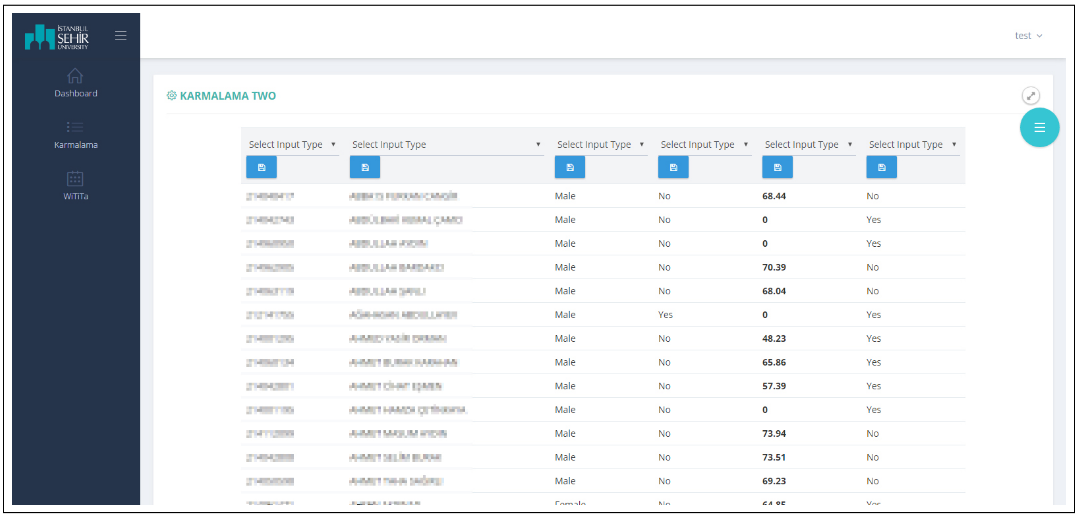


FIGURE 3.5: Screenshot of Karmalama: Step A2

In step A3(Figure 3.6), it is necessary to enter the number of classes to be created and to select which constraints are to be solved by applying our model. The options presented are as follows:

1. Assign same number of female students to each group(if applicable),
2. Assign same number of international students to each group(if applicable),
3. Assign same number of repeat students to each group(if applicable),
4. The average score of classes are close to each other.

Based on these options, the mathematical model is created in the background.

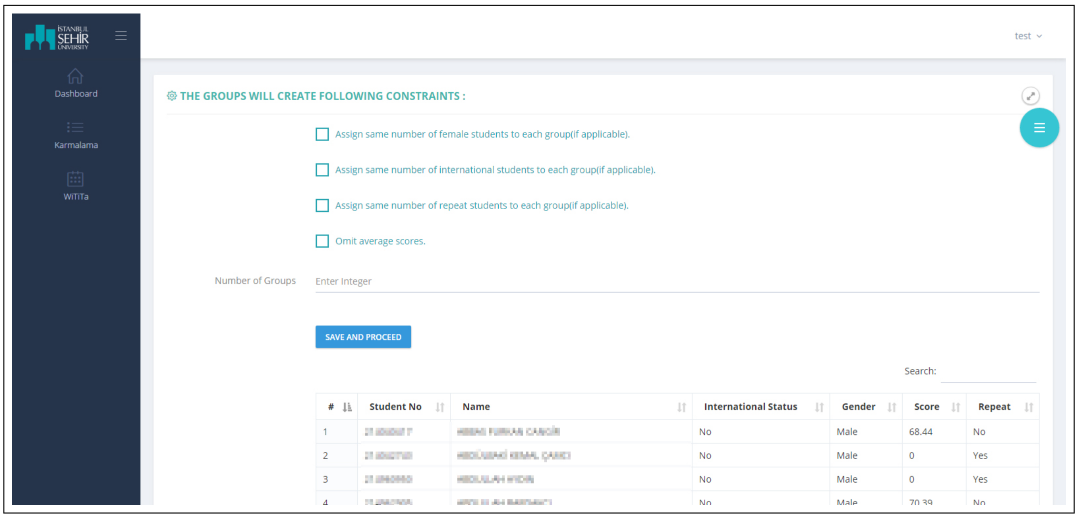


FIGURE 3.6: Screenshot of Karmalama: Step A3

In the last Step A4 (Figure 3.7), if the problem has a feasible solution, the result and graphical reports (for commenting number of female/international/repeat student in classes, and average score of classes) of the project are displayed. The user can export student list by classes or the list as a whole in two file formats: PDF and Excel, as well as print the list via the printer. In addition, a database is maintained so that the user can access first step, third step and last step at the desired time.

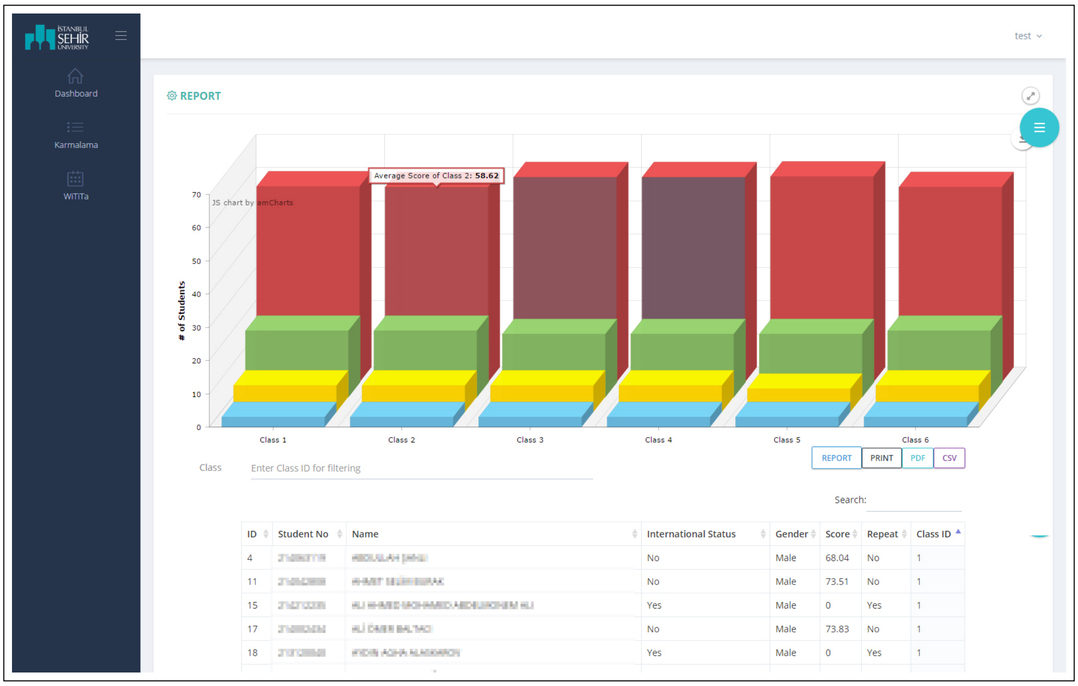


FIGURE 3.7: Screenshot of Karmalama: Step A4

We have formulated team selection problem, which we name "Karmalama", as an optimization problem that has a linear objective function and linear constraints. In order to reduce the complexity of the mathematical model, we have implemented some simplifications: (1) We calculated average number of female, repetition and international students in each module by number of the class that would be generated. While determining our parameters, we set the limits by rounding up and down the averages so that we can distribute them evenly to the generated classes. (2) We have reduced the minimum and maximum difference (deviation) from the average grade of classes generated (based on final grades) from the overall grade average. These simplifications have facilitated the solution of the created model.

The sets, parameters and decision variables used while constructing the mixed-integer programming model are as follows:

Sets

i : index for students,

j : index for classes.

Parameters

G_i : Gender type of student i (Male=0, Female=1),

N_i : Nationality type of student i (Turkish=0, International=1),

R_i : Repeat status type of student i (Regular=0, Repeat=1),

P_i : Score of student i,

A : General average score(out of 100),

l, l_{up}, l_{down} : Average, minimum and maximum number of female students should be in class,

is, is_{up}, is_{down} : Average, minimum and maximum number of international students should be in class,

r, r_{up}, r_{down} : Average, minimum and maximum number of repeat students should be in class,

c, c_{up}, c_{down} : Average, minimum and maximum number of students should be in class.

Decision variables

$$x_{ij} = \begin{cases} 1 & \text{if student } i \text{ is assigned to class } j \\ 0 & \text{otherwise} \end{cases},$$

Cap_j = number of students assigned to class j,

δ = deviation from general average score.

Mathematical Model for Karmalama

$$\min \delta$$

subject to

$$\sum_j x_{ij} = 1, \qquad \forall i \in I \tag{3.1}$$

$$\sum_i x_{ij} = Cap_j, \qquad \forall j \in J \tag{3.2}$$

$$c_{down} \leq Cap_j \leq c_{up}, \qquad \forall j \in J \tag{3.3}$$

$$l_{down} \leq \sum_i x_{ij} G_i \leq l_{up}, \qquad \forall j \in J \tag{3.4}$$

$$is_{down} \leq \sum_i x_{ij} N_i \leq is_{up}, \qquad \forall j \in J \tag{3.5}$$

$$r_{down} \leq \sum_i x_{ij} R_i \leq r_{up}, \qquad \forall j \in J \tag{3.6}$$

$$c(A - \delta) \leq \sum_i x_{ij} P_i \leq c(A + \delta), \qquad \forall j \in J \tag{3.7}$$

$$Cap_j \geq 0, \qquad \forall j \in J \tag{3.8}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in I, \forall j \in J \tag{3.9}$$

$$\delta \geq 0. \tag{3.10}$$

c, is, l, and r are calculated by dividing the given number of students by desired number of classes. If the outcome is in decimal form, up and down value of related value are obtained by rounding up and rounding down the outcome to nearest integer, respectively.

Constraint 3.1 allows each student to be assigned to only one class. Constraint 3.2 calculates the capacity of the class. Constraint 3.3 ensures that the class capacity is between the maximum and minimum number of students that should be in the classroom. Constraint 3.4 allows the number of female students to be in the classroom to be between the maximum and the minimum number of female students, if there are female students. Constraint 3.5 imposes a similar restriction for international students. Constraint 3.6 imposes a similar restriction for students who are in a repeat state. Constraint 3.7 ensures that each class has an average score deviating least from the general average. The rest of the constraints are non-negative, while certain variables are binary in nature.

We made an assumption in our mathematical model to avoid non-linearity of constraints.

The problem becomes non-linear when the number of students in the classroom multiply with the deviation (Constraint 3.7). Thus, we had to address this problem rounding up and rounding down the number of students in the classroom.

3.2 Preparing Weekly Timetables: WeTiTa

The problem is given a special name: "WeTiTa". User is also required to create a new project for this section. After the project is created, there are four different sections: (1) Identification of Classes, (2) Identification of Instructors, (3) Preliminary Review, and (4) Report.

In the Step B1 (Figure 3.8), the eligibility conditions for the particular classes, and the classes themselves need to be defined. There are two options of data entry for the user: (1) create classes manually and enter data, and (2) create classes by uploading a pre-prepared list draft in Excel.

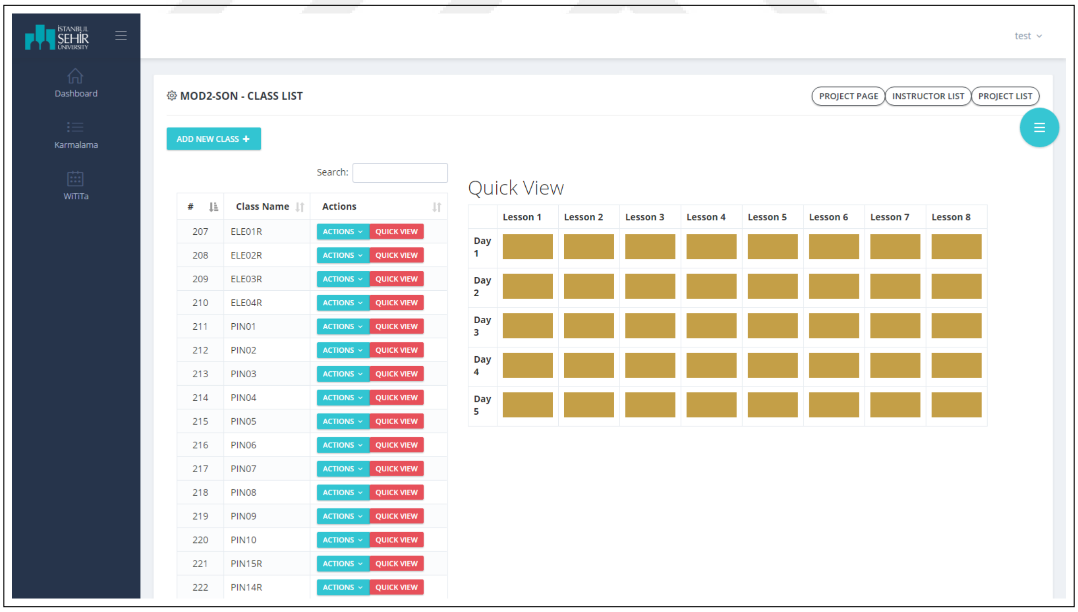


FIGURE 3.8: Screenshot of WeTiTa: List of Classes

When the user creates the classes manually, the class name must be specified on the table in the edit screen to indicate which lesson of the day is lecture, or tutorial, or unavailable. In addition, the user has been given the opportunity to select already defined possible patterns (Figure 3.9).

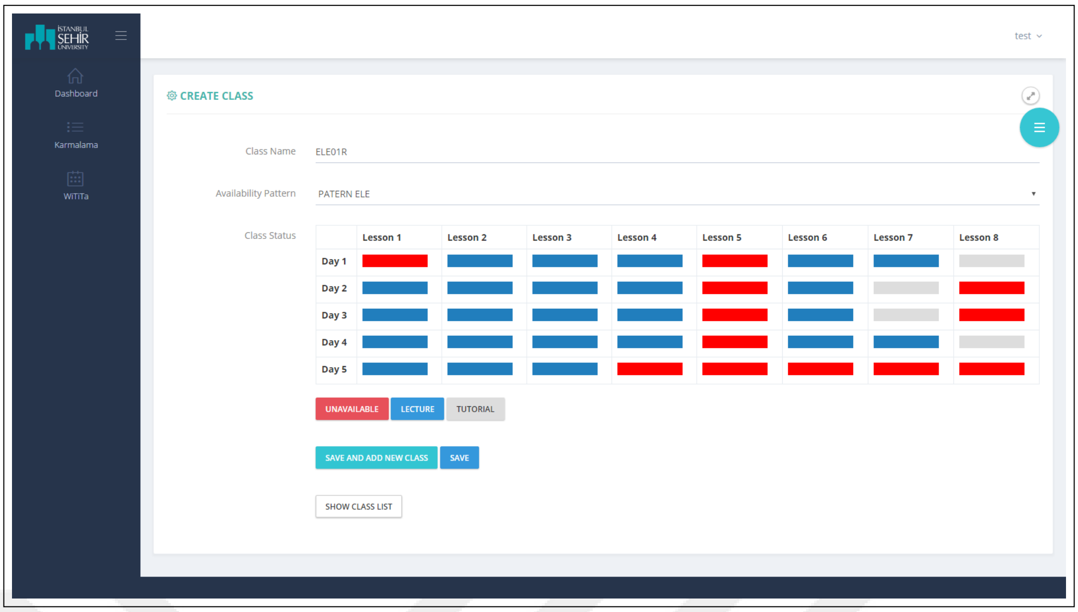


FIGURE 3.9: Screenshot of WeTiTa: Edit Class

When feeding data with Excel (Figure 3.10), the user must enter the class name and enter one of the defined pattern IDs. If Pattern ID is empty, all hours will be defined as lecture. Tutorial or unavailable indicator must be included in the class otherwise. There is a quick view option on the list of classes screen to check the status of the created classes. This will allow user to view generated class patterns without having to enter the edit screen.

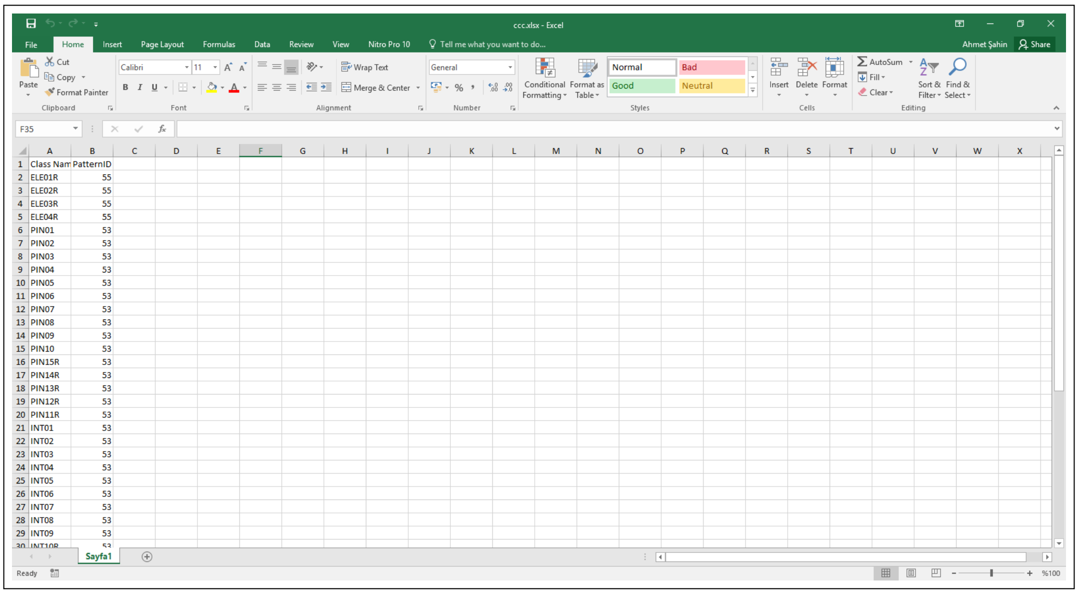


FIGURE 3.10: Screenshot of WeTiTa: Excel Draft for Class Upload

After the successful identification of classes, it is necessary to proceed to Step B2 (Figure 3.11): identification of the instructors. The Instructors list screen provides summary information. There are two alternatives for user to define instructors similar to the Step B1: (1) create instructors manually and enter data, and (2) create instructor by uploading a pre-prepared list draft in Excel.

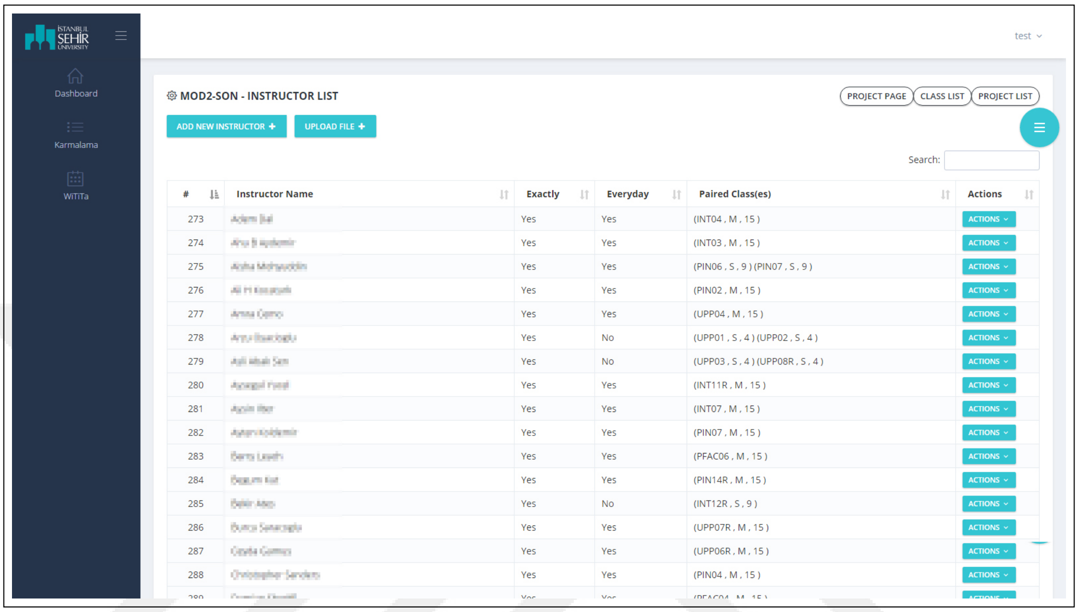


FIGURE 3.11: Screenshot of WeTiTa: List of Instructors

When creating instructor manually, user must specify the name, availability, workload, and type of the instructor. An additional input constitutes whether the instructor is exactly assign and having class every day (Figure 3.12).

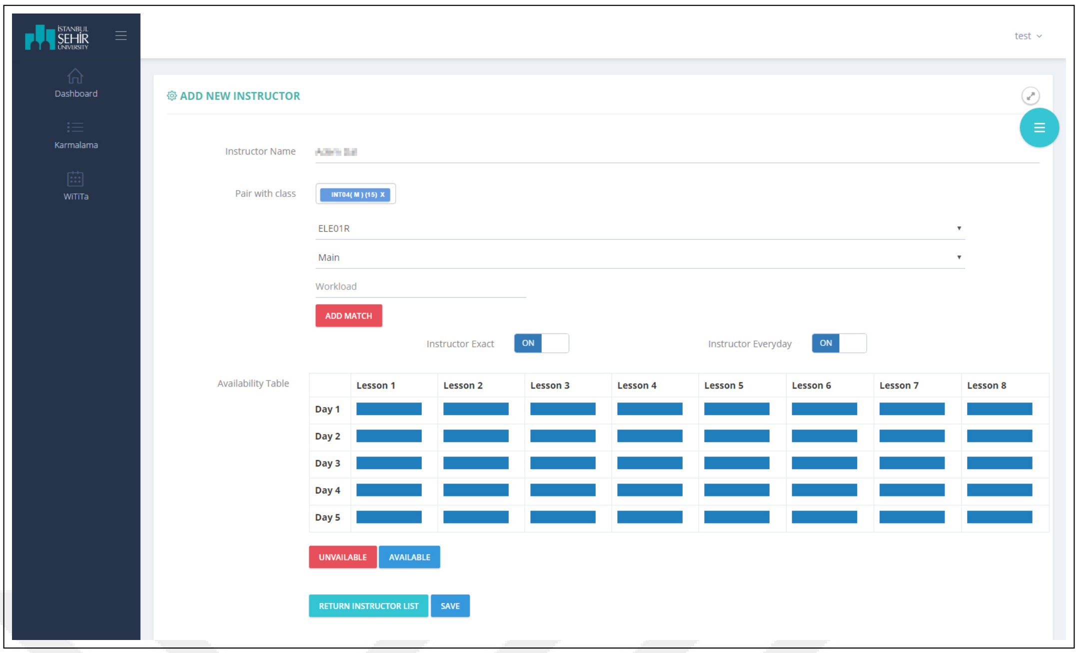
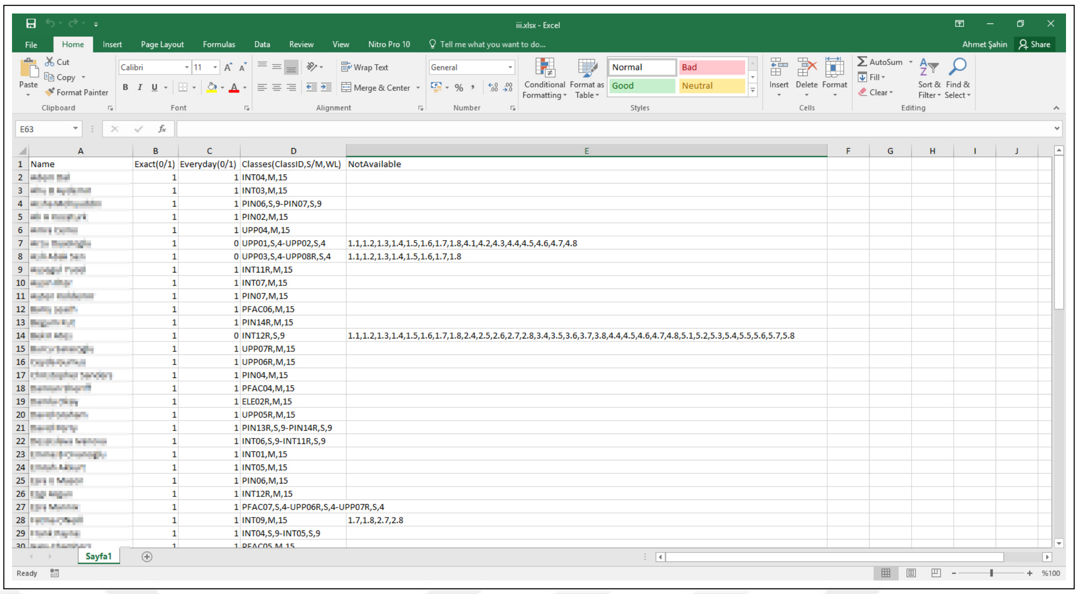


FIGURE 3.12: Screenshot of WeTiTa: Edit Instructor

When user uploads an Excel draft to create an instructor (Figure 3.13), the user must specify the name of the instructor, the classes that can be assigned as main or support, whether or not he/she is a exact instructor (exactly assigned), and whether or not he/she will conduct the class every day. Defined instructors will be displayed as –all hours available– in the absence of any in the unavailable column. To specify the available state, user must select a table on the instructor edit screen. In the Instructor List page (Figure 3.11), all the entered information is presented as summary in tabular form. In this way, the information entered in the main list will be confirmed.



	Name	Exact(0/1)	Everyday(0/1)	Classes(ClassID,S/M,WL)	NotAvailable
2		1	1	INT04,M,15	
3		1	1	INT03,M,15	
4		1	1	PIN06,S,9-PIN07,S,9	
5		1	1	PIN02,M,15	
6		1	1	UPP04,M,15	
7		1	0	UPP01,S,4-UPP02,S,4	1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,4.1,4.2,4.3,4.4,4.5,4.6,4.7,4.8
8		1	0	UPP03,S,4-UPP08R,S,4	1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8
9		1	1	INT11R,M,15	
10		1	1	INT07,M,15	
11		1	1	PIN07,M,15	
12		1	1	PFAC06,M,15	
13		1	1	PIN14R,M,15	
14		1	0	INT12R,S,9	1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,2.4,2.5,2.6,2.7,2.8,3.4,3.5,3.6,3.7,3.8,4.4,4.5,4.6,4.7,4.8,5.1,5.2,5.3,5.4,5.5,5.6,5.7,5.8
15		1	1	UPP07R,M,15	
16		1	1	UPP06R,M,15	
17		1	1	PIN04,M,15	
18		1	1	PFAC04,M,15	
19		1	1	ELE02R,M,15	
20		1	1	UPP05R,M,15	
21		1	1	PIN13R,S,9-PIN14R,S,9	
22		1	1	INT06,S,9-INT11R,S,9	
23		1	1	INT01,M,15	
24		1	1	INT05,M,15	
25		1	1	PIN06,M,15	
26		1	1	INT12R,M,15	
27		1	1	PFAC07,S,4-UPP06R,S,4-UPP07R,S,4	
28		1	1	INT09,M,15	1.7,1.8,2.7,2.8
29		1	1	INT04,S,9-INT05,S,9	
30		1	1	PFAC05,M,15	

Figure 3.13: Screenshot of WeTiTa: Excel Draft for Instructor Upload

In Step B3 – Review (Figure 3.14), data is checked from database before the generated data is sent to solver. Database is checked to see if it is infeasible. If there is infeasibility, user is prompted about correction. For example, Ahmet who is defined as the main teacher for the ELE1 class, and Sena who is defined as the support instructor, run the conformity check for every lesson of every day. Suppose Ahmet is unavailable on Wednesday for 2nd and 3rd lesson hours. If Sena is not available for 2nd and 3rd class hours, user will be warned and prompted for correction. Likewise, in Tutorial lessons, the main instructor is checked for availability. The final step can be defined as a summary step, error reports – if exist – are displayed with the class list and the instructor list to the user. Whether or not the error is valid, the user can proceed further and solve the problem by sending it to solver.

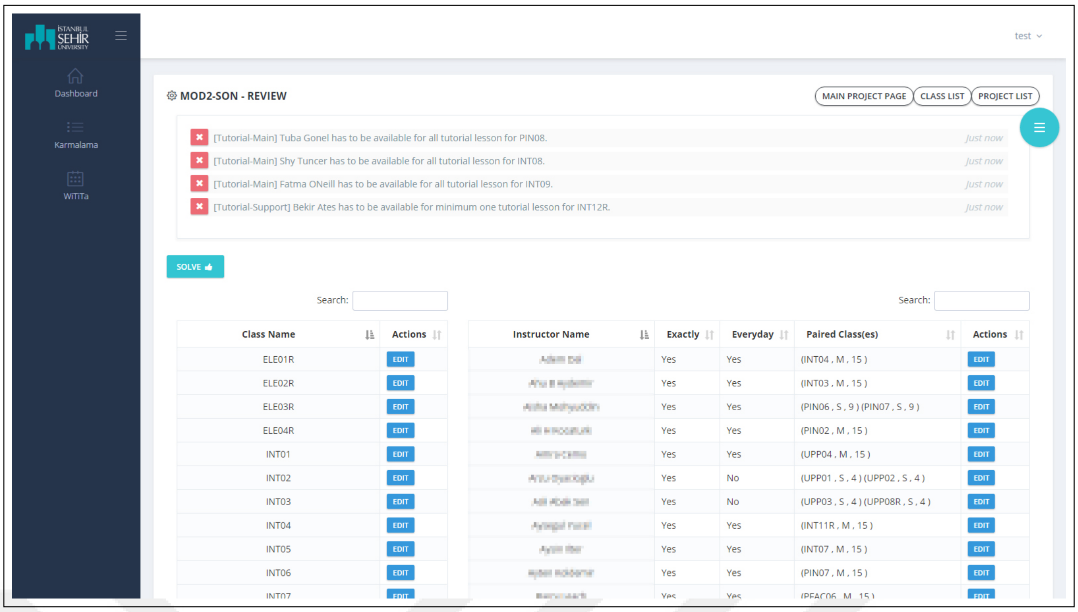


FIGURE 3.14: Screenshot of WeTiTa: Review

In the last Step B4 – Reports (Figure 3.15), reports are presented in timetabling format. Error information about charts that are not generated are also provided in this step. The user can list timetables in two different ways: list for instructors, and list for classes. User can export outputs as PDF or Excel file as well as print them via the printer.

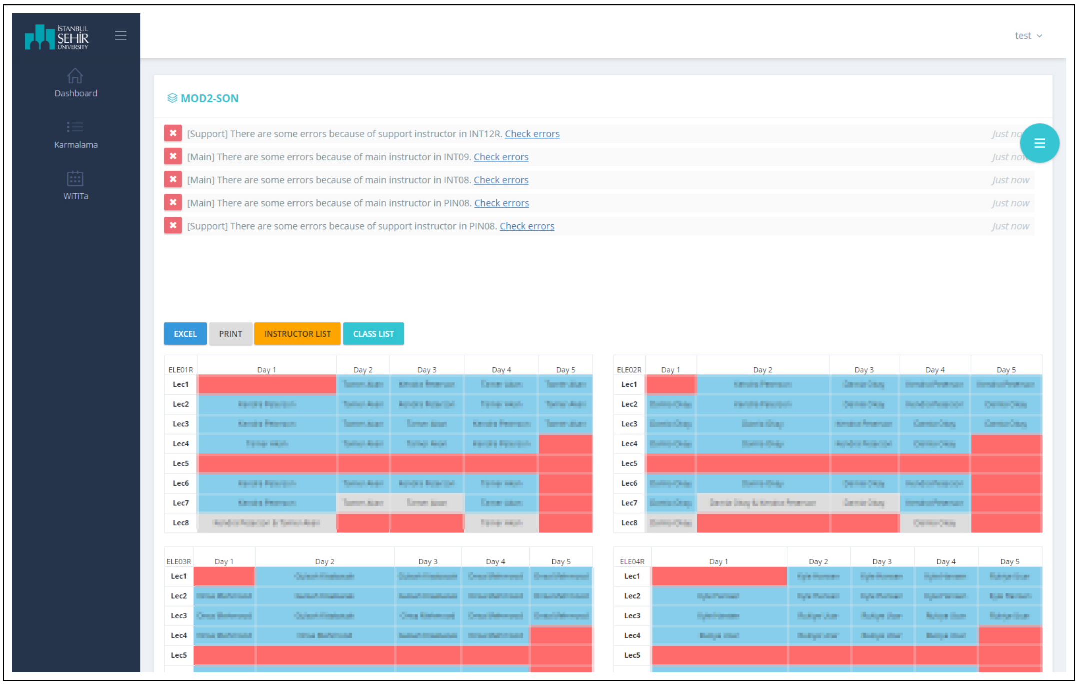


FIGURE 3.15: Screenshot of WeTiTa: Report

In addition, database records are kept for the user to access these section again at the desired time. User Review or Report section is accessible from the project list.

To create the timetables, we formulated a constraint satisfaction problem that is without objective function, but has linear constraints. Constraint satisfaction problems are mathematical problems defined as a set of variables (i.e., events) whose state must satisfy a number of constraints (i.e., resources such as rooms and time periods) [96, 97].

Since the model is developed using 4 different indexes (i for instructor, j for class, d for day, t for time slots), it would be extremely difficult to solve with 80 instructors, 50 classes, 5 days and 8 class hours. To ease the solution, indexes related sub-sets were created. This has allowed us to simplify the problem without reducing the resolution time and without the need for extra constraints. We also created sub-problems from the main problem. Sub-problems are automatically generated by following the steps below:

1. Any teacher with instructor type main is selected.
2. If the class to which the main instructor is associated is selected, the support instructor is identified.
3. There are other classes to which the support instructor is associated.
4. There are other main instructors in those classes, who are then identified.
5. This cycle stops when all such associations are identified.

The sets, parameters and decision variables used while constructing the binary integer programming model are as follows:

Sets

i: Set of instructors,

j: Set of classes,

d: Set of days,

t: Set of time slots,

A_{idt}: Set of time slots for each day d where instructor i is available,

B_{jdt}: Set of lessons on day d time slot t where lecture can be conduct for class j,

C_{jdt}: Set of lessons on day d time slot t where tutorial can be conduct for class j,

DD_{ij}: Set of instructor i matched for class j,

E_{ij}: Set of main instructor i matched for class j,

F_{ij}: Set of support instructor i matched for class j,

K_{jdt}: The time slot sets indicating second lessons that should be conducted consequently without a break in between,

L_{jdt}: The time slot sets indicating second lessons that should be conducted consequently with a break in between,

M_{jdt}: The sets of lesson that should be assigned by checking previous two lessons,

O_{jdt}: The sets of lesson that should be assigned by checking previous two lessons with ignoring break,

P_{jdt}: The sets of lesson that should be assigned by checking four previous lesson,

EX_i: Set of instructors should be assigned exactly,

H_i: Set of instructors should be assigned everyday,

ZZ_{jdt}: Set of off-lessons for class j,

Parameters

N_{ij}: Weekly workload of instructor i for class j

Decision variables

$$x_{ijdt} = \begin{cases} 1 & \text{if instructor } i \text{ is assigned to class } j \text{ on day } d \text{ at time slot } t \\ 0 & \text{otherwise} \end{cases}$$

Mathematical Model for WeTiTa

$$\min z = 0$$

subject to

$$\sum_{i \in DD_{ij},\, and\, A_{idt}} x_{ijdt} = 1, \qquad \forall j, d, t \in B_{jdt} \tag{3.11}$$

$$\sum_{i \in E_{ij},\, and\, A_{idt}} x_{ijdt} = 1, \qquad \forall j, d, t \in C_{jdt} \tag{3.12}$$

$$\sum_{j \in B_{jdt},\, and\, DD_{ij}} x_{ijdt} \leq 1, \qquad \forall i, d, t \in A_{idt} \tag{3.13}$$

$$\sum_{i,d,t \in F_{ij},\, C_{jdt},\, and\, A_{idt}} x_{ijdt} = 1, \qquad \forall j \tag{3.14}$$

$$\sum_{j \in C_{jdt}, F_{ij},\, and\, DD_{ij}} x_{ijdt} \leq 1, \qquad \forall i, d, t \in A_{idt} \tag{3.15}$$

$$\sum_{i \in DD_{ij}} x_{ijdt} = 0, \qquad \forall j, d, t \in ZZ_{jdt} \tag{3.16}$$

$$x_{ijdt} \geq x_{ijd(t-1)} - x_{ijd(t-2)}, \qquad \forall i, j, d, t \in K_{jdt}, B_{jdt}, DD_{ij},\ and\ A_{idt} \tag{3.17}$$

$$x_{ijdt} \geq x_{ijd(t-2)} - x_{ijd(t-3)}, \qquad \forall i, j, d, t \in L_{jdt}, B_{jdt}, DD_{ij},\ and\ A_{idt} \tag{3.18}$$

$$x_{ijdt} \geq x_{ijd(t-1)}, \qquad \forall i, j, d, t \in M_{jdt}, B_{jdt}, DD_{ij},\ and\ A_{idt} \tag{3.19}$$

$$x_{ijdt} + x_{ijd(t-4)} = 1, \qquad \forall i, j, d, t \in O_{jdt}, B_{jdt}, DD_{ij},\ and\ A_{idt} \tag{3.20}$$

$$x_{ijdt} \geq x_{ijd(t-2)}, \qquad \forall i, j, d, t \in P_{jdt}, B_{jdt}, DD_{ij},\ and\ A_{idt} \tag{3.21}$$

$$\sum_{d,t \in B_{jdt},\, and\, A_{idt}} x_{ijdt} = N_{ij}, \qquad \forall i, j \in DD_{ij},\ and\, EX_i \tag{3.22}$$

$$\sum_{d,t \in B_{jdt},\, and\, A_{idt}} x_{ijdt} \leq N_{ij}, \qquad \forall i, j \in DD_{ij} \notin EX_i \tag{3.23}$$

$$\sum_{j,t \in B_{jdt}, A_{idt},\, and\, DD_{ij}} x_{ijdt} \geq 1, \qquad \forall i, d \in H_i \tag{3.24}$$

$$x_{ijdt} \in \{0,1\}. \qquad \forall\, i, j, d, t \tag{3.25}$$

Constraint 3.11 ensures that there is only one instructor per time slot per lecture every day. Constraint 3.12 ensures that the main instructor conducts tutorial lessons per time slot every day. Constraint 3.13 prevents each instructor from conducting more than one lesson at the same time slot. Constraint 3.14 forces the Support instructor to conduct a tutorial on the class that is defined as support for four days. Constraint 3.15 prevents the Support instructor from conducting tutorials in more than one class during the same time slot. Constraint 3.16 prevents assignments to off-time. Constraint 3.22 ensures that the instructor, which must be exactly assigned, does as much work as the specified workload. The constraint 3.23 ensures that the instructor, which is not exactly assigned, fulfill as much workload as specified. Constraint 3.24 ensures that full-time instructors have a course every day. Constraint 3.17, 3.18, 3.19, 3.20, and 3.21 allow the lecture lessons to be conducted in a sequence shown in Figure 3.1. For example, in the Sequence Type 3 + Break + 3, we implement M set that allows us to schedule two consecutive of same instructor followed by a third class that can be associated with any related instructor. We implement L set that allows the user to choose an instructor for fourth lesson corresponding to the instructors of the two lessons before the break. A different instructor is assigned to the fifth lesson from the first lesson. Finally, we ensure that sixth lesson is also constructed by the instructor from the fifth lesson by implementing M set.

This mathematical model is appropriate for solving the problem. However, we encountered a few errors in practice. These errors were also witnessed while solving the problems manually. On the other hand, SEPP relaxes out some constraints manually while creating manual timetables. For example, if a main instructor is unavailable after lunch, the solution will be infeasible for related sub-problem because of the instructor has to conduct all tutorial courses. Manual timetabling delivers the solution through variations like in instructor's availability or workload, or passes through the lesson which has problem to others and then thinks the solution. However, this creates an issue for our system. The data are checked in Review section and corrected before sending to solver (as we mentioned above). If the solution is sent from Review section despite the error, weekly timetables of the non-error sub-problems will be created and errors for non-created timetables will be displayed on Report screen as also mentioned before. But, the weekly timetable is not created yet. We made some revisions in our model and built a new model to overcome this and send to solver with errors though with newly created model.

By assigning dummy instructors to the lessons which have error in sub-problems, we aimed to show user where the system is having error for creating weekly timetables. The system solves the problem without regard to whether it has any error with the newly mathematical model , and also shows weekly timetables and errors are encountered in Report screen. In the mathematical model, only the objection function has changed and a new set has been added. The aim of the newly created mathematical model is to minimize the dummy instructors who are assigned. Added set and revised objection function (3.26) are as follows:

Dm_i: Set of dummy instructors

$$\min z = \sum_{i,j,d,t \in Dm_i} x_{ijdt} \tag{3.26}$$

Chapter 4

Case Study

4.1 Preparing Class Lists

We ran the preliminary study of Karmalama with 117 students to be distributed in six different classes. In Table 4.1, the students were classified as:

TABLE 4.1: Classification of Second Example: Karmalama

	Male	Female	Total
Turkish	57	42	99
Regular	33	27	60
Repeat	24	15	39
International	13	5	18
Regular	8	1	9
Repeat	5	4	9
Total	70	47	117

The absentees or unsuccessful students on module final exam are considered as repeat students, and their scores are taken as 0. The general average score is 60.17. It is difficult to produce classes manually due to the combinatorial complexity. According to SEPP, this process spans at least 3-4 hours. Even then, the solution cannot be considered optimum. The only way to achieve the optimum solution is to create classes with a mathematical model. This preliminary study took merely a second to produce classes with the mathematical model we created. This time depends on the number of students in Karmalama and the iterations that occur during the solution. The largest deviation from the overall grade average was 1.67. Acceptable class averages deviation range determined by SEPP are ± 2. The graphical results produced in first case study for Karmalama are as follows:

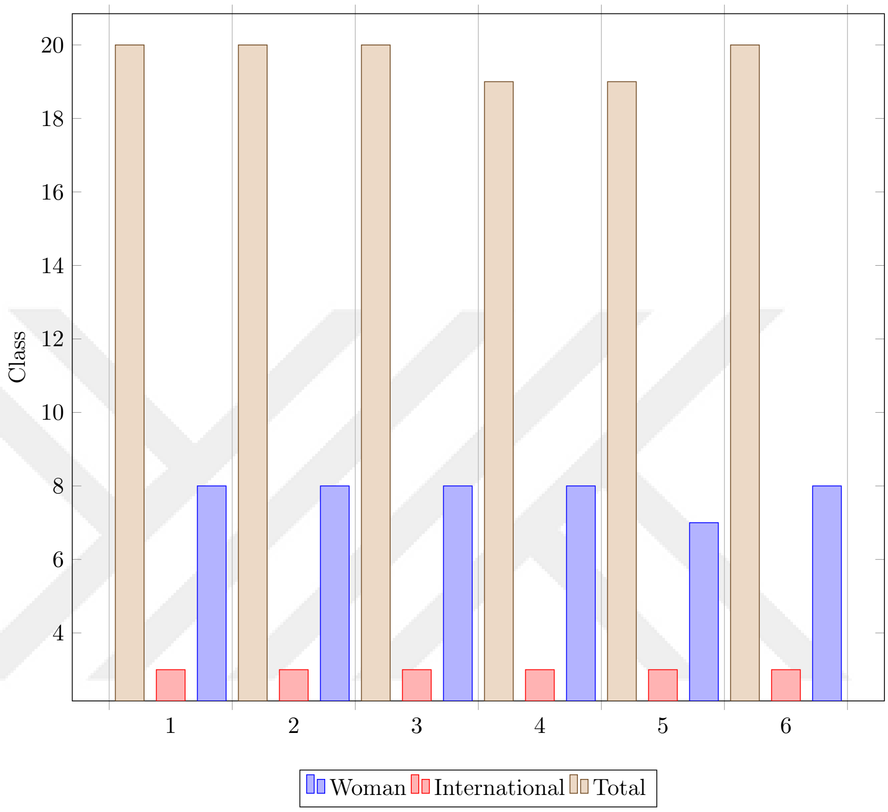


FIGURE 4.1: The graphical results: Karmalama 1

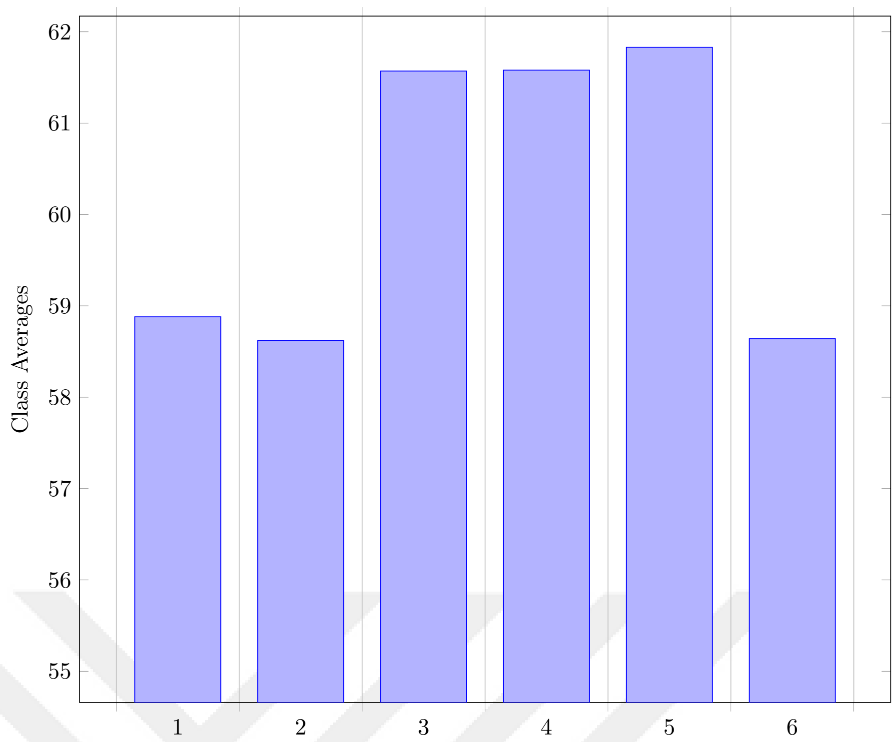


FIGURE 4.2: The graphical results: Karmalama 1

The second example we have performed Karmalama to distribute 133 students to 7 different classes. The classification of students is done in Table 4.2.

TABLE 4.2: Classification of First Example: Karmalama

	Male	Female	Total
Turkish	58	70	128
Regular	19	29	48
Repeat	39	41	80
International	1	4	5
Regular	0	2	2
Repeat	1	2	3
Total	59	74	133

The general average score is 25.12. In this case study, it took 7 seconds to create classes with our mathematical model. This is due to the number of iterations as we have described above. The maximum deviation from the general average was 0.18. The graphical results produced in second case study for Karmalama are as follows:

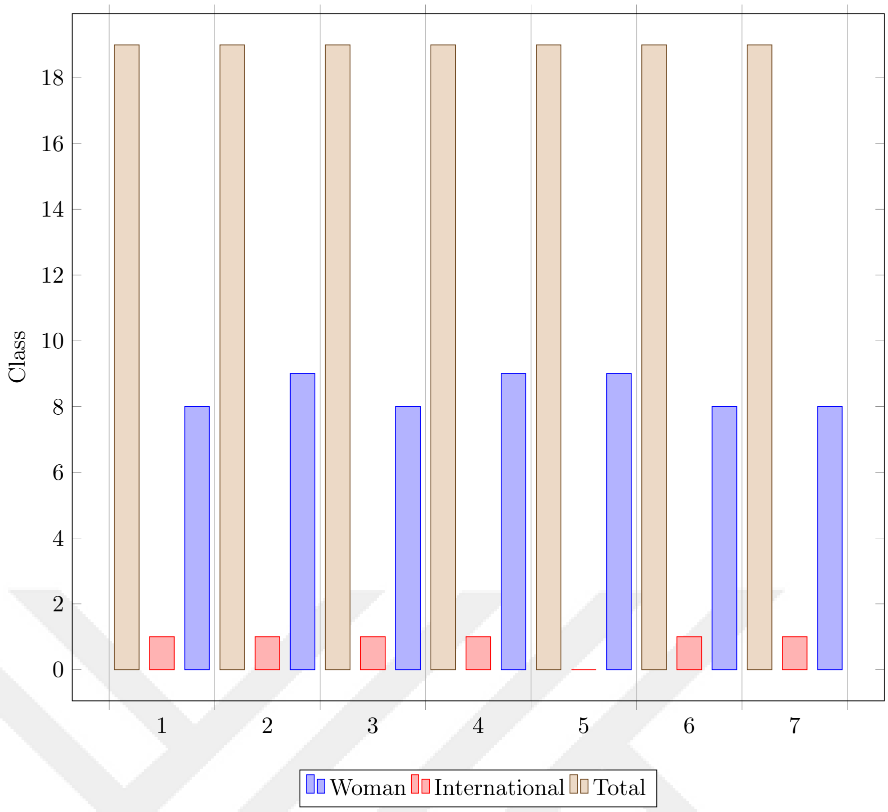


FIGURE 4.3: The graphical results: Karmalama 2

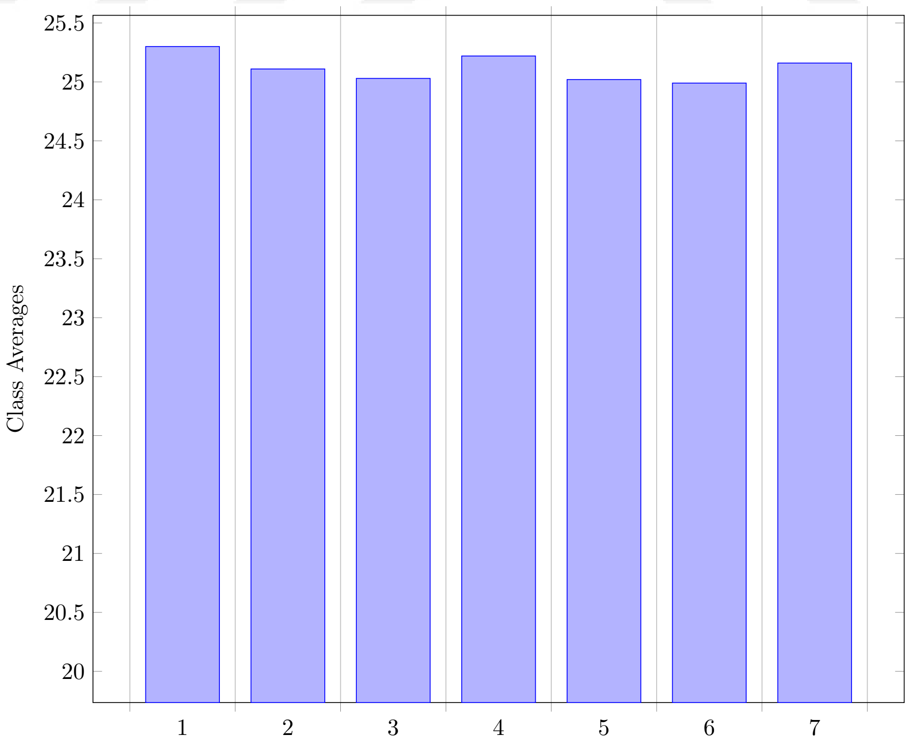


FIGURE 4.4: The graphical results: Karmalama 2

As a result of the assumption aforementioned, it must be noticed while evaluating solution of two case studies that the deviation in the first problem is higher due to the average number of student in class between 19 and 20. These are integer values, however, the deviation calculated for the number of student in class is over 19.5.

4.2 Preparing Weekly Timetables

We have implemented a project with 46 class and 71 instructors to create timetables.

According to SEPP, it takes around 1 day to prepare timetables manually. After the data is entered, the database query takes approximately 4 seconds, and the mathematical models take approximately 7 seconds to solve.

We compared two incomplete weekly timetables (dummy instructor assigned) and a complete weekly timetable with manually generated outcomes.

In the first case, FaO is paired with INT09 as main instructor, but she is unavailable for Lessons 7 and 8 on Mondays and Tuesdays. Nonetheless, the main instructor has to conduct all tutorials, as mentioned before. The system automatically assigns a dummy instructor to the lessons to compensate for her unavailability. In manually generated case, May was paired with INT09 as support instructor and assigned to tutorials on Monday and Tuesday instead of FaO. Moreover, an additional tutorial was arranged for FaO. The weekly timetable for INT09 was generated by the system and the manually generated timetable are shown in Table 4.3 and Table 4.4, respectively.

Table 4.3: Weekly timetable for INT09 generated by the system

INT09	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		Fa.O	M.Ay	M.Ay	Fa.O
Less.2	Fa.O	Fa.O	M.Ay	M.Ay	Fa.O
Less.3	Fa.O	Fa.O	M.Ay	Fa.O	Fa.O
Less.4					
Less.5	Fa.O	M.Ay	Fa.O	Fa.O	
Less.6	M.Ay	M.Ay	Fa.O	Fa.O	
Less.7	M.Ay	Dummy	Fa.O	Fa.O	
Less.8	M.Ay & Dummy			Fa.O	

Table 4.4: Weekly timetable for INT09 created manually

INT09	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		Fa.O	Fa.O	Fa.O	M.Ay
Less.2	Fa.O	Fa.O	Fa.O	Fa.O	Fa.O
Less.3	Fa.O	Fa.O	Fa.O	Fa.O	Fa.O
Less.4					
Less.5	M.Ay	M.Ay	Fa.O	Fa.O	
Less.6	M.Ay	M.Ay	Fa.O	M.Ay	
Less.7	M.Ay	T:M.Ay	T:Fa.O	M.Ay	
Less.8	T:M.Ay		T:Fa.O	T:Fa.O	

In the second case, BeA is paired with INT12R as a support instructor, but is available for only first 3 lessons on Tuesdays, Wednesdays, and Thursdays. Therefore, BeA is not able to conduct a tutorial lesson assigned to him. Also, for ensuring consecutive (The sequences control constraint forces exemption of an instructor assigned to pre-break lesson from post-break lesson), once again, the system automatically generates a weekly timetable using dummy instructor instead of BeA for the tutorial lesson shown in Table 4.5. In manual approach, the working hours of BeA were re-arranged. First 3 lessons on Wednesday were changed to last 5 lessons (Lesson 3, 4, 5, 6, and 7) on Wednesday. As a result of this arrangement, and also ignoring the consecutive, the weekly timetables that satisfies requests was generated manually as shown in Table 4.6.

Table 4.5: Weekly timetable for INT12R generated by the system

INT12R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Be.A	Be.A	Be.A	Ex.A
Less.2	Dummy	Be.A	Be.A	Be.A	Ex.A
Less.3	Ex.A	Be.A	Be.A	Be.A	Ex.A
Less.4					
Less.5	Ex.A	Ex.A	Ex.A	Ex.A	
Less.6	Ex.A	Ex.A	Ex.A	Ex.A	
Less.7	Ex.A	Ex.A	Ex.A	Ex.A	
Less.8	Ex.A & Dummy			Ex.A	

Table 4.6: Weekly timetable for INT12R created manually

INT12R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Be.A	Ex.A	Be.A	Ex.A
Less.2	Ex.A	Be.A	Ex.A	Be.A	Ex.A
Less.3	Ex.A	Be.A	Be.A	Be.A	Ex.A
Less.4					
Less.5	Ex.A	Ex.A	Be.A	Ex.A	
Less.6	Ex.A	Ex.A	Be.A	Ex.A	
Less.7	Ex.A	T:Ex.A	T:Ex.A & Be.A	Ex.A	
Less.8	T:Ex.A			T:Ex.A	

Finally, the weekly timetable for PFAC05 was generated by the system and manually are shown in Table 4.7 and Table 4.8. The timetables are similar to each other. However, instructors' pattern are different, which is a normal condition.

Table 4.7: Weekly timetable for PFAC05 generated by the system

INT09	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		M.A	M.A	G.C	G.C
Less.2	G.C	M.A	M.A	G.C	G.C
Less.3	G.C	G.C	G.C	G.C	G.C
Less.4	G.C	G.C	G.C	G.C	
Less.5					
Less.6	G.C	G.C & M.A	G.C	G.C	
Less.7	G.C				
Less.8					

Table 4.8: Weekly timetable for PFAC05 created manually

INT09	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		G.C	M.A	G.C	G.C
Less.2	G.C	G.C	M.A	G.C	G.C
Less.3	G.C	G.C	M.A	G.C	G.C
Less.4	G.C	G.C	M.A	G.C	
Less.5					
Less.6	G.C	T:G.C	T:G.C & M.A	T:G.C	
Less.7	T:G.C				

The sequences control constraints are generally ignored during manual preparation of timetables. Reconfiguration of instructors working hours is advised to resolve this problem. Spring 2017 planning encounters two errors: unavailability of main instructor for a tutorial lesson, and dissatisfaction of desired the sequences control constraints. Both errors are ignored by assigning dummy instructors.

The SEPP approved timetables and declared that our study satisfied their requests. The list of classes and the list of instructors to be generate, as well as the generated timetables are in Appendix A, B, C.

Chapter 5

Summary and Conclusions

Educational timetabling problems are in NP-complete problem class. Many methods and computer software have been developed to solve such problems. There are many of academic journals on this subject, and the methods used to solve timetabling problems are presented in problem-specific international events (Practice and Theory of Automated Timetabling Conference Series). In addition, international timetabling competitions are held to determine the best solution of specially designed problems.

The purpose of this study was to examine the methods used to prepare timetables in the educational institutions, and create timetables and class lists of the İstanbul Şehir University School of Languages English Preparatory Program (SEPP) and to evaluate the results. We have prepared a program with web based decision support system by applying mixed and binary integer programming models. To find the exact result with integer programming, weekly timetables were created in total 11 seconds and the class lists were created in total 5 seconds.

The planning of classrooms is done by the university administration; it is not part of this thesis. However, if this changes, it is possible to integrate such planning with Karmalama section.

Also, the study could be extended to assign instructors to classes. Depending on the qualifications instructors and relation between instructors by prizing, the pairs could be determine in which class to assign. This application shortens the user's preparation time before solution (data entry). SEPP did not require us to allocate instructors according to some predefined attributes (like module level, qualification etc.), which has appeared to be a major limitation of our study.

The solution of educational timetabling problems that occur at least twice during a training period requires intensive work and valuable resources. Therefore, it is as difficult as it is important for educational institutions. In addition, the constraints and objectives may vary from institution to institution. This situation allows the sustainability of research and further application in this area. It is extremely difficult to develop a general system with universal applicability that allows all educational institutions to solve myriads of distinct educational timetabling problems.

This method produces a structure that has been designed to allow solution approaches while taking into account flexible constraints, which shall certainly be meaningful for elementary schools, high schools and preparation programs for many universities, independently of expert knowledge in a DSS system.

Appendix A

Karmalama: Results of Case Studies

Table A.1: Karmalama:Data and Results of First Case Study

ID	International Status	Gender	Score	Repeat	Assigned Class
4	No	Male	68.04	No	1
11	No	Male	73.51	No	1
15	Yes	Male	0	Yes	1
17	No	Male	73.83	No	1
18	Yes	Male	0	Yes	1
22	No	Female	61.43	Yes	1
25	No	Male	70.94	No	1
26	No	Male	73.3	No	1
39	No	Female	69.53	No	1
44	No	Male	78.41	No	1
45	No	Male	69.13	No	1
48	No	Male	65.73	Yes	1
54	No	Female	69.03	No	1
55	No	Female	65.46	Yes	1
77	No	Male	76.03	No	1
81	No	Female	69.83	No	1
94	Yes	Female	0	Yes	1
97	No	Male	57.75	Yes	1

Table A.1 – Continued

ID	International Status	Gender	Score	Repeat	Assigned Class
98	No	Female	75.1	No	1
108	No	Female	60.64	Yes	1
2	No	Male	0	Yes	2
5	Yes	Male	0	Yes	2
7	No	Male	65.86	Yes	2
10	No	Male	73.94	No	2
13	No	Female	64.85	Yes	2
16	Yes	Male	75.06	No	2
30	No	Female	67.03	No	2
38	No	Female	74.65	No	2
47	No	Female	78.13	No	2
51	No	Male	0	Yes	2
53	No	Male	66.66	No	2
66	Yes	Male	76.95	No	2
70	No	Male	70.48	No	2
75	No	Male	77.78	No	2
82	No	Female	50.43	Yes	2
88	No	Male	68.18	No	2
106	No	Male	75.73	No	2
110	No	Female	67.63	No	2
112	No	Female	58.36	Yes	2
113	No	Female	60.76	Yes	2
9	No	Male	0	Yes	3
23	Yes	Female	53.21	Yes	3
27	No	Male	65.64	Yes	3
31	No	Female	67.45	No	3
33	No	Female	72.63	No	3
35	No	Female	67.53	No	3
41	No	Female	79.31	No	3
46	No	Female	68.41	No	3

Table A.1 – Continued

ID	International Status	Gender	Score	Repeat	Assigned Class
62	No	Male	67.73	No	3
64	No	Male	75.36	No	3
65	Yes	Male	87.86	No	3
73	No	Male	55.34	Yes	3
83	No	Female	86.04	No	3
87	No	Male	0	Yes	3
90	No	Male	67.1	No	3
92	Yes	Male	62.45	Yes	3
93	No	Male	64.65	Yes	3
103	No	Male	62.65	Yes	3
114	No	Female	68.38	No	3
0	No	Male	68.44	No	4
6	No	Male	48.23	Yes	4
8	No	Male	57.39	Yes	4
21	No	Female	73.56	No	4
32	No	Female	76.29	No	4
37	No	Female	70.79	No	4
49	No	Female	71.21	No	4
58	No	Male	0	Yes	4
61	No	Male	61.58	Yes	4
67	Yes	Male	67.45	No	4
72	No	Male	64.1	Yes	4
78	No	Male	68.06	No	4
86	Yes	Male	69.1	No	4
95	No	Female	70.25	No	4
96	No	Male	47.03	Yes	4
100	No	Female	67.51	No	4
102	No	Male	61.34	Yes	4
104	Yes	Female	72.66	No	4
116	No	Female	54.94	Yes	4

Table A.1 – Continued

ID	International Status	Gender	Score	Repeat	Assigned Class
1	No	Male	0	Yes	5
3	No	Male	70.39	No	5
14	No	Female	55.64	Yes	5
40	No	Male	75.4	No	5
42	No	Female	83.89	No	5
50	No	Female	67.83	No	5
57	Yes	Male	59.15	Yes	5
60	No	Male	61.16	Yes	5
63	No	Female	55.31	Yes	5
68	No	Male	76.43	No	5
71	No	Male	0	Yes	5
74	Yes	Male	82.09	No	5
79	No	Male	70.85	No	5
85	Yes	Male	69.68	No	5
89	No	Male	73.45	No	5
99	No	Female	73.06	No	5
105	No	Male	66	Yes	5
111	No	Female	74.5	No	5
115	No	Female	60.01	Yes	5
12	No	Male	69.23	No	6
19	No	Female	74.89	No	6
20	No	Female	64.68	Yes	6
24	No	Male	73.1	No	6
28	No	Female	0	Yes	6
29	Yes	Female	45.96	Yes	6
34	Yes	Male	70.88	No	6
36	No	Male	69.46	No	6
43	No	Male	63.73	Yes	6
52	No	Female	0	Yes	6
56	No	Male	71.76	No	6

Table A.1 – Continued

ID	International Status	Gender	Score	Repeat	Assigned Class
59	No	Male	0	Yes	6
69	No	Male	75.95	No	6
76	No	Male	81.84	No	6
80	No	Male	69	No	6
84	No	Male	70.51	No	6
91	No	Female	58.24	Yes	6
101	Yes	Female	64.69	Yes	6
107	No	Male	73.08	No	6
109	No	Female	75.86	No	6

Table A.2: Karmalama:Data and Results of Second Case Study

ID	International Status	Gender	Score	Repeat	Assigned Class
42	No	Male	61.08	No	1
43	No	Male	60.86	No	1
44	No	Male	60.71	No	1
45	No	Female	59.83	No	1
46	No	Female	59.63	No	1
47	No	Male	59.59	No	1
48	No	Male	59.53	No	1
49	No	Male	59.5	No	1
66	Yes	Female	0	Yes	1
75	No	Female	0	Yes	1
77	No	Female	0	Yes	1
80	No	Female	0	Yes	1
98	No	Male	0	Yes	1
99	No	Female	0	Yes	1
103	No	Female	0	Yes	1
106	No	Male	0	Yes	1
113	No	Male	0	Yes	1
123	No	Male	0	Yes	1
130	No	Male	0	Yes	1
3	No	Male	76.4	No	2
6	No	Male	75.15	No	2
11	No	Male	72.28	No	2
23	No	Male	65.25	No	2
30	No	Female	63.4	No	2
31	No	Female	63.18	No	2
40	No	Male	61.38	No	2
61	No	Female	0	Yes	2
62	No	Female	0	Yes	2
63	No	Male	0	Yes	2

Table A.2 – Continued

ID	International Status	Gender	Score	Repeat	Assigned Class
104	No	Female	0	Yes	2
105	No	Female	0	Yes	2
110	No	Female	0	Yes	2
116	No	Male	0	Yes	2
122	Yes	Male	0	Yes	2
125	No	Male	0	Yes	2
127	No	Female	0	Yes	2
129	No	Male	0	Yes	2
131	No	Female	0	Yes	2
1	No	Female	77.18	No	3
7	No	Male	75.01	No	3
18	No	Female	68.76	No	3
24	No	Male	64.84	No	3
27	No	Male	63.54	No	3
29	No	Male	63.48	No	3
32	No	Male	62.83	No	3
53	No	Male	0	Yes	3
69	No	Male	0	Yes	3
87	No	Female	0	Yes	3
89	No	Female	0	Yes	3
91	No	Female	0	Yes	3
93	No	Male	0	Yes	3
108	No	Male	0	Yes	3
114	No	Female	0	Yes	3
115	No	Female	0	Yes	3
118	No	Male	0	Yes	3
121	No	Male	0	Yes	3
126	No	Female	0	Yes	3
2	No	Male	76.76	No	4
8	No	Female	73.56	No	4

Table A.2 – Continued

ID	International Status	Gender	Score	Repeat	Assigned Class
12	No	Male	71.82	No	4
17	No	Male	69.04	No	4
28	No	Female	63.49	No	4
34	Yes	Male	62.63	No	4
37	No	Male	61.88	No	4
54	No	Male	0	Yes	4
55	No	Female	0	Yes	4
58	No	Male	0	Yes	4
60	No	Female	0	Yes	4
73	No	Male	0	Yes	4
74	No	Female	0	Yes	4
81	No	Female	0	Yes	4
92	No	Male	0	Yes	4
97	No	Female	0	Yes	4
100	No	Male	0	Yes	4
112	No	Female	0	Yes	4
119	No	Female	0	Yes	4
13	No	Male	71.65	No	5
14	No	Male	71.64	No	5
15	No	Female	70.53	No	5
16	No	Male	69.18	No	5
21	No	Female	66.4	No	5
25	No	Female	64.24	No	5
38	No	Female	61.83	No	5
50	No	Female	0	Yes	5
51	No	Male	0	Yes	5
64	No	Male	0	Yes	5
72	No	Male	0	Yes	5
76	No	Male	0	Yes	5
78	No	Female	0	Yes	5

Table A.2 – Continued

ID	International Status	Gender	Score	Repeat	Assigned Class
84	No	Female	0	Yes	5
86	No	Male	0	Yes	5
90	No	Female	0	Yes	5
94	No	Male	0	Yes	5
124	No	Male	0	Yes	5
132	No	Female	0	Yes	5
0	No	Female	80.08	No	6
9	No	Female	72.73	No	6
19	No	Female	67.74	No	6
20	No	Male	67.68	No	6
33	No	Female	62.8	No	6
36	No	Male	62.1	No	6
39	No	Male	61.75	No	6
52	No	Male	0	Yes	6
56	No	Female	0	Yes	6
57	No	Female	0	Yes	6
59	No	Female	0	Yes	6
67	No	Male	0	Yes	6
82	No	Male	0	Yes	6
83	No	Female	0	Yes	6
85	No	Male	0	Yes	6
96	No	Male	0	Yes	6
102	No	Male	0	Yes	6
109	Yes	Male	0	Yes	6
128	No	Male	0	Yes	6
4	No	Male	75.88	No	7
5	No	Female	75.66	No	7
10	No	Female	72.69	No	7
22	No	Male	66.33	No	7
26	No	Male	63.59	No	7

Table A.2 – Continued

ID	International Status	Gender	Score	Repeat	Assigned Class
35	Yes	Male	62.53	No	7
41	No	Female	61.31	No	7
65	No	Female	0	Yes	7
68	No	Male	0	Yes	7
70	No	Male	0	Yes	7
71	No	Male	0	Yes	7
79	No	Male	0	Yes	7
88	No	Male	0	Yes	7
95	No	Female	0	Yes	7
101	No	Male	0	Yes	7
107	No	Male	0	Yes	7
111	No	Female	0	Yes	7
117	No	Female	0	Yes	7
120	No	Female	0	Yes	7

Appendix B

WeTiTa: Data of Case Study

Table B.1: WeTiTa: Class List

Class Name	Pattern	Class Name	Pattern
ELE01R	Pattern ELE	INT05	Pattern INT
ELE02R	Pattern ELE	INT06	Pattern INT
ELE03R	Pattern ELE	INT07	Pattern INT
ELE04R	Pattern ELE	INT08	Pattern INT
PIN01	Pattern INT	INT09	Pattern INT
PIN02	Pattern INT	INT10R	Pattern INT
PIN03	Pattern INT	INT11R	Pattern INT
PIN04	Pattern INT	INT12R	Pattern INT
PIN05	Pattern INT	UPP01	Pattern UPP
PIN06	Pattern INT	UPP02	Pattern UPP
PIN07	Pattern INT	UPP03	Pattern UPP
PIN08	Pattern INT	UPP04	Pattern UPP
PIN09	Pattern INT	UPP05R	Pattern UPP
PIN10	Pattern INT	UPP06R	Pattern UPP
PIN15R	Pattern INT	UPP07R	Pattern UPP
PIN14R	Pattern INT	UPP08R	Pattern UPP
PIN13R	Pattern INT	PFAC01	Pattern UPP
PIN12R	Pattern INT	PFAC02	Pattern UPP

Table B.1 – Continued

Class Name	Pattern	Class Name	Pattern
PIN11R	Pattern INT	PFAC03	Pattern UPP
INT01	Pattern INT	PFAC04	Pattern UPP
INT02	Pattern INT	PFAC05	Pattern UPP
INT03	Pattern INT	PFAC06	Pattern UPP
INT04	Pattern INT	PFAC07	Pattern UPP

Table B.2: WeTiTa: Pattern for Elementary Module

Pattern ELE	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1	OFF	Lecture	Lecture	Lecture	Lecture
Less.2	Lecture	Lecture	Lecture	Lecture	Lecture
Less.3	Lecture	Lecture	Lecture	Lecture	Lecture
Less.4	Lecture	Lecture	Lecture	Lecture	OFF
Less.5	OFF	OFF	OFF	OFF	OFF
Less.6	Lecture	Lecture	Lecture	Lecture	OFF
Less.7	Lecture	Tutorial	Tutorial	Lecture	OFF
Less.8	Tutorial	OFF	OFF	Tutorial	OFF

Table B.3: WeTiTa: Pattern for Pre-Intermediate Module and Intermediate Module

Pattern INT	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1	OFF	Lecture	Lecture	Lecture	Lecture
Less.2	Lecture	Lecture	Lecture	Lecture	Lecture
Less.3	Lecture	Lecture	Lecture	Lecture	Lecture
Less.4	OFF	OFF	OFF	OFF	OFF
Less.5	Lecture	Lecture	Lecture	Lecture	OFF
Less.6	Lecture	Lecture	Lecture	Lecture	OFF
Less.7	Lecture	Tutorial	Tutorial	Lecture	OFF
Less.8	Tutorial	OFF	OFF	Tutorial	OFF

Table B.4: WeTiTa: Pattern for Upper-Intermediate Module and Pre-Faculty Module

Pattern UPP	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1	OFF	Lecture	Lecture	Lecture	Lecture
Less.2	Lecture	Lecture	Lecture	Lecture	Lecture
Less.3	Lecture	Lecture	Lecture	Lecture	Lecture
Less.4	Lecture	Lecture	Lecture	Lecture	OFF
Less.5	OFF	OFF	OFF	OFF	OFF
Less.6	Lecture	Tutorial	Tutorial	Tutorial	OFF
Less.7	Tutorial	OFF	OFF	OFF	OFF
Less.8	OFF	OFF	OFF	OFF	OFF

Table B.5: WeTiTa: Instructor List

Instructor Name	**Exact**	**Everyday**	**Paired Classes (Class,Ins Type,WL)**	**Not Available Lessons**
Ad.D	Yes	Yes	INT04,Main,15	
A.B.A	Yes	Yes	INT03,Main,15	
Ai.M	Yes	Yes	PIN06,Support,9 PIN07,Support,9	
A.H.K	Yes	Yes	PIN02,Main,15	
A.Ce	Yes	Yes	UPP04,Main,15	
Ar.O	Yes	No	UPP01,Support,4 UPP02,Support,4	Mon.1-8, Thu.1-8
A.A.S	Yes	No	UPP03,Support,4 UPP08R,Support,4	Mon.1-8
Ay.Y	Yes	Yes	INT11R,Main,15	
Ay.I	Yes	Yes	INT07,Main,15	
A.K	Yes	Yes	PIN07,Main,15	
B.L	Yes	Yes	PFAC06,Main,15	
Be.K	Yes	Yes	PIN14R,Main,15	

Table B.5 – Continued

Instructor Name	**Exact**	**Everyday**	**Paired Classes (Class,Instructor Type,WL)**	**Not Available Lessons**
Be.A	Yes	No	INT12R,Support,9	Mon.1-8, Tue.4-8, Wed.4-8, Thu.4-8, Fri.1-8
Bu.S	Yes	Yes	UPP07R,Main,15	
Ce.G	Yes	Yes	UPP06R,Main,15	
C.S	Yes	Yes	PIN04,Main,15	
D.S	Yes	Yes	PFAC04,Main,15	
D.O	Yes	Yes	ELE02R,Main,15	
Da.G	Yes	Yes	UPP05R,Main,15	
Da.P	Yes	Yes	PIN13R,Support,9 PIN14R,Support,9	
De.S	Yes	Yes	INT06,Support,9 INT11R,Support,9	
E.B.C	Yes	Yes	INT01,Main,15	
E.Ak	Yes	Yes	INT05,Main,15	
E.U.M	Yes	Yes	PIN06,Main,15	
Ex.A	Yes	Yes	INT12R,Main,15	
E.M	Yes	Yes	PFAC07,Support,4 UPP06R,Support,4 UPP07R,Support,4	
Fa.O	Yes	Yes	INT09,Main,15	Mon.7-8, Tue.7-8
F.Pa	Yes	Yes	INT04,Support,9 INT05,Support,9	
G.C	Yes	Yes	PFAC05,Main,15	
G.K	Yes	No	ELE03R,Support,9	Mon.1-8
Gu.O	Yes	Yes	INT02,Main,15	

Table B.5 – Continued

Instructor Name	**Exact**	**Everyday**	**Paired Classes (Class,Instructor Type,WL)**	**Not Available Lessons**
H.D	Yes	Yes	PFAC01,Support,4 PFAC02,Support,4 PFAC03,Support,4	
He.M	Yes	Yes	UPP08R,Main,15	
H.K	Yes	Yes	PFAC01,Main,15	
Ih.B	Yes	No	INT08,Support,9	Mon.1-8, Tue.4-8, Thu.4-8, Fri.4-8
Ip.E	Yes	Yes	UPP02,Main,15	Mon.8, Tue.8, Wed.8, Thu.8, Fri.8
I.G	Yes	No	PIN04,Support,9	Mon.1-8, Wed.1-8
J.K	Yes	Yes	PIN02,Support,9 PIN03,Support,9	Thu.4-8
K.P	Yes	Yes	ELE01R,Support,9 ELE02R,Support,9	
K.S	Yes	Yes	PIN05,Main,15	
K.H	Yes	Yes	ELE04R,Support,9 PIN01,Support,9	
L.B	Yes	Yes	PFAC07,Main,15	
Me.A	Yes	No	PIN15R,Support,9	Mon.1-8, Tue.4-8
M.In	Yes	No	UPP04,Support,4 UPP05R,Support,4	Mon.1-8
M.A	Yes	No	PFAC05,Support,4	Mon.1-8, Thu.1-8
M.So	Yes	Yes	INT06,Main,15	

Table B.5 – Continued

Instructor Name	**Exact**	**Everyday**	**Paired Classes (Class,Instructor Type,WL)**	**Not Available Lessons**
M.Ay	Yes	Yes	INT09,Support,9 INT10R,Support,9	
Me.I	Yes	Yes	PIN11R,Main,15	Tue.4-8
M.J.J	Yes	Yes	PFAC02,Main,15	
Ne.W	Yes	Yes	PIN11R,Support,9 PIN12R,Support,9	
O.M	Yes	Yes	ELE03R,Main,15	Wed.4-6
O.O.D	Yes	No	PFAC04,Support,4 PFAC06,Support,4	
R.G	Yes	Yes	PIN08,Support,9 PIN09,Support,9	
R.V	Yes	Yes	PFAC03,Main,15	
R.U	Yes	Yes	ELE04R,Main,15	
Se.M	Yes	Yes	INT03,Support,9 INT07,Support,9	
S.U.Y	Yes	Yes	UPP01,Main,15	
Se.I	Yes	Yes	PIN12R,Main,15	
S.Y	Yes	Yes	PIN03,Main,15	
S.A	Yes	Yes	PIN09,Main,15	
Sh.T	Yes	Yes	INT08,Main,15	Wed.3-8
Ta.Y	Yes	Yes	PIN13R,Main,15	
T.A	Yes	Yes	ELE01R,Main,15	
Th.M	Yes	Yes	INT01,Support,9 INT02,Support,9	
T.G	Yes	No	PIN08,Main,15	Mon.1-8, Tue.1-8
Tu.S	Yes	Yes	INT10R,Main,15	
Tu.A	Yes	Yes	PIN10,Main,15	
T.U.I	Yes	Yes	PIN15R,Main,15	
V.R	Yes	Yes	PIN10,Support,9 PIN05,Support,9	

Table B.5 – Continued

Instructor Name	**Exact**	**Everyday**	**Paired Classes (Class,Instructor Type,WL)**	**Not Available Lessons**
Y.B	Yes	Yes	PIN01,Main,15	
Ze.G	Yes	Yes	UPP03,Main,15	

Appendix C

WeTiTa: Results of Case Study

ELE01R	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		T.A	T.A	T.A	T.A
Less.2	T.A	T.A	T.A	T.A	T.A
Less.3	T.A	T.A	K.P	K.P	T.A
Less.4	T.A	T.A	K.P	K.P	
Less.5					
Less.6	K.P	T.A	K.P	K.P	
Less.7	K.P	T.A	T.A	K.P	
Less.8	K.P & T.A			T.A	

ELE02R	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		K.P	K.P	K.P	K.P
Less.2	K.P	K.P	K.P	K.P	DO
Less.3	K.P	D.O	D.O	D.O	D.O
Less.4	D.O	D.O	D.O	D.O	
Less.5					
Less.6	D.O	D.O	D.O	D.O	
Less.7	D.O	D.O & K.P	D.O	D.O	
Less.8	D.O			D.O	

ELE03R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		G.K	O.M	O.M	O.M
Less.2	Dummy	G.K	O.M	O.M	G.K
Less.3	O.M	G.K	O.M	O.M	G.K
Less.4	O.M	O.M	G.K	O.M	
Less.5					
Less.6	O.M	O.M	G.K	G.K	
Less.7	O.M	G.K & O.M	O.M	G.K	
Less.8	O.M			O.M	

ELE04R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		K.H	K.H	R.U	R.U
Less.2	K.H	K.H	K.H	R.U	R.U
Less.3	K.H	R.U	R.U	R.U	R.U
Less.4	R.U	R.U	R.U	K.H	
Less.5					
Less.6	R.U	R.U	R.U	K.H	
Less.7	R.U	R.U	R.U	K.H	
Less.8	K.H & R.U			R.U	

PIN01	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Y.B	Y.B	K.H	Y.B
Less.2	Y.B	Y.B	Y.B	K.H	Y.B
Less.3	Y.B	Y.B	Y.B	Y.B	K.H
Less.4					
Less.5	Y.B	K.H	K.H	Y.B	
Less.6	K.H	K.H	K.H	Y.B	
Less.7	K.H	K.H & Y.B	Y.B	Y.B	
Less.8	Y.B			Y.B	

PIN02	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		J.K	J.K	J.K	A.H.K
Less.2	A.H.K	J.K	J.K	J.K	A.H.K
Less.3	A.H.K	A.H.K	J.K	A.H.K	A.H.K
Less.4					
Less.5	A.H.K	A.H.K	A.H.K	A.H.K	
Less.6	J.K	A.H.K	A.H.K	A.H.K	
Less.7	J.K	A.H.K	A.H.K	A.H.K	
Less.8	A.H.K & J.K			A.H.K	

PIN03	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		S.Y	S.Y	S.Y	S.Y
Less.2	J.K	S.Y	S.Y	S.Y	J.K
Less.3	J.K	S.Y	S.Y	S.Y	J.K
Less.4					
Less.5	S.Y	J.K	J.K	S.Y	
Less.6	S.Y	J.K	J.K	Dummy	
Less.7	S.Y	J.K & S.Y	S.Y	S.Y	
Less.8	S.Y			S.Y	

PIN04	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		C.S	Dummy	I.G	I.G
Less.2	Dummy	C.S	C.S	I.G	C.S
Less.3	C.S	C.S	C.S	I.G	C.S
Less.4					
Less.5	C.S	I.G	C.S	I.G	
Less.6	C.S	I.G	C.S	C.S	
Less.7	C.S	C.S & I.G	C.S	C.S	
Less.8	C.S			C.S	

PIN05	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		K.S	K.S	K.S	K.S
Less.2	K.S	K.S	K.S	K.S	K.S
Less.3	K.S	K.S	V.R	K.S	K.S
Less.4					
Less.5	K.S	V.R	V.R	K.S	
Less.6	V.R	V.R	V.R	V.R	
Less.7	V.R	K.S	K.S	V.R	
Less.8	K.S & V.R			K.S	

PIN06	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Ai.M	E.U.M	E.U.M	E.U.M
Less.2	E.U.M	Ai.M	E.U.M	E.U.M	E.U.M
Less.3	E.U.M	E.U.M	Ai.M	E.U.M	E.U.M
Less.4					
Less.5	E.U.M	E.U.M	Ai.M	E.U.M	
Less.6	Ai.M	E.U.M	Ai.M	Ai.M	
Less.7	Ai.M	E.U.M	E.U.M	Ai.M	
Less.8	Ai.M & E.U.M			E.U.M	

PIN07	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		A.K	Ai.M	Ai.M	A.K
Less.2	Ai.M	A.K	Ai.M	Ai.M	A.K
Less.3	Ai.M	A.K	A.K	A.K	Ai.M
Less.4					
Less.5	A.K	Ai.M	A.K	A.K	
Less.6	A.K	Ai.M	A.K	A.K	
Less.7	A.K	Ai.M & A.K	A.K	A.K	
Less.8	A.K			A.K	

PIN08	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		Dummy	R.G	Dummy	T.G
Less.2	Dummy	Dummy	R.G	R.G	T.G
Less.3	Dummy	R.G	T.G	T.G	T.G
Less.4					
Less.5	R.G	R.G	T.G	T.G	
Less.6	R.G	R.G	T.G	T.G	
Less.7	R.G	Dummy	T.G	T.G	
Less.8	R.G & Dummy			T.G	

PIN09	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		R.G	SA	SA	R.G
Less.2	R.G	R.G	SA	SA	SA
Less.3	R.G	SA	SA	SA	SA
Less.4					
Less.5	SA	SA	R.G	SA	
Less.6	SA	SA	R.G	R.G	
Less.7	SA	R.G & SA	SA	R.G	
Less.8	SA			SA	

PIN10	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		V.R	V.R	V.R	V.R
Less.2	V.R	V.R	V.R	V.R	Tu.A
Less.3	V.R	Tu.A	Tu.A	Tu.A	Tu.A
Less.4					
Less.5	Tu.A	Tu.A	Tu.A	Tu.A	
Less.6	Tu.A	Tu.A	Tu.A	Tu.A	
Less.7	Tu.A	Tu.A & V.R	Tu.A	Tu.A	
Less.8	Tu.A			Tu.A	

PIN15R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Me.A	T.U.I	Me.A	T.U.I
Less.2	Dummy	Me.A	T.U.I	Me.A	Me.A
Less.3	T.U.I	Me.A	T.U.I	T.U.I	Me.A
Less.4					
Less.5	T.U.I	T.U.I	Me.A	T.U.I	
Less.6	T.U.I	T.U.I	Me.A	T.U.I	
Less.7	T.U.I	T.U.I	Me.A & T.U.I	T.U.I	
Less.8	T.U.I			T.U.I	

PIN14R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Da.P	Da.P	Be.K	Da.P
Less.2	Be.K	Da.P	Da.P	Be.K	Be.K
Less.3	Be.K	Be.K	Be.K	Be.K	Be.K
Less.4					
Less.5	Be.K	Be.K	Be.K	Be.K	
Less.6	Da.P	Be.K	Be.K	Da.P	
Less.7	Da.P	Be.K	Be.K	Da.P	
Less.8	Be.K & Da.P			Be.K	

PIN13R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Ta.Y	Ta.Y	Da.P	Ta.Y
Less.2	Da.P	Ta.Y	Ta.Y	Da.P	Ta.Y
Less.3	Da.P	Ta.Y	Ta.Y	Da.P	Ta.Y
Less.4					
Less.5	Ta.Y	Da.P	Da.P	Ta.Y	
Less.6	Ta.Y	Da.P	Da.P	Ta.Y	
Less.7	Ta.Y	Da.P & Ta.Y	Ta.Y	Ta.Y	
Less.8	Ta.Y			Ta.Y	

PIN12R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Ne.W	Se.I	Ne.W	Se.I
Less.2	Se.I	Ne.W	Se.I	Ne.W	Se.I
Less.3	Se.I	Se.I	Se.I	Se.I	Ne.W
Less.4					
Less.5	Se.I	Se.I	Ne.W	Se.I	
Less.6	Ne.W	Se.I	Ne.W	Se.I	
Less.7	Ne.W	Se.I	Se.I	Se.I	
Less.8	Ne.W & Se.I			Se.I	

PIN11R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Me.I	Ne.W	Me.I	Ne.W
Less.2	Ne.W	Me.I	Ne.W	Me.I	Me.I
Less.3	Ne.W	Me.I	Me.I	Me.I	Me.I
Less.4					
Less.5	Me.I	Ne.W	Me.I	Me.I	
Less.6	Me.I	Ne.W	Me.I	Ne.W	
Less.7	Me.I	Me.I & Ne.W	Me.I	Ne.W	
Less.8	Me.I			Me.I	

INT01	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		E.B.C	E.B.C	E.B.C	E.B.C
Less.2	E.B.C	E.B.C	E.B.C	E.B.C	E.B.C
Less.3	E.B.C	E.B.C	Th.M	E.B.C	E.B.C
Less.4					
Less.5	E.B.C	Th.M	Th.M	E.B.C	
Less.6	Th.M	Th.M	Th.M	Th.M	
Less.7	Th.M	E.B.C	E.B.C	Th.M	
Less.8	E.B.C & Th.M			E.B.C	

INT02	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Th.M	Th.M	Th.M	Th.M
Less.2	Th.M	Th.M	Th.M	Th.M	Gu.O
Less.3	Th.M	Gu.O	Gu.O	Gu.O	Gu.O
Less.4					
Less.5	Gu.O	Gu.O	Gu.O	Gu.O	
Less.6	Gu.O	Gu.O	Gu.O	Gu.O	
Less.7	Gu.O	Gu.O & Th.M	Gu.O	Gu.O	
Less.8	Gu.O			Gu.O	

INT03	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		A.B.A	A.B.A	A.B.A	A.B.A
Less.2	A.B.A	A.B.A	A.B.A	A.B.A	A.B.A
Less.3	A.B.A	A.B.A	Se.M	A.B.A	A.B.A
Less.4					
Less.5	A.B.A	Se.M	Se.M	A.B.A	
Less.6	Se.M	Se.M	Se.M	Se.M	
Less.7	Se.M	A.B.A	A.B.A	Se.M	
Less.8	A.B.A & Se.M			A.B.A	

INT04	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Ad.D	Ad.D	Ad.D	Ad.D
Less.2	Ad.D	Ad.D	Ad.D	Ad.D	Ad.D
Less.3	Ad.D	Ad.D	F.Pa	Ad.D	Ad.D
Less.4					
Less.5	Ad.D	F.Pa	F.Pa	Ad.D	
Less.6	F.Pa	F.Pa	F.Pa	F.Pa	
Less.7	F.Pa	Ad.D	Ad.D	F.Pa	
Less.8	Ad.D & F.Pa			Ad.D	

INT05	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		F.Pa	F.Pa	F.Pa	F.Pa
Less.2	F.Pa	F.Pa	F.Pa	F.Pa	E.Ak
Less.3	F.Pa	E.Ak	E.Ak	E.Ak	E.Ak
Less.4					
Less.5	E.Ak	E.Ak	E.Ak	E.Ak	
Less.6	E.Ak	E.Ak	E.Ak	E.Ak	
Less.7	E.Ak	E.Ak & F.Pa	E.Ak	E.Ak	
Less.8	E.Ak			E.Ak	

INT06	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		M.So	M.So	De.S	M.So
Less.2	M.So	M.So	M.So	De.S	M.So
Less.3	M.So	M.So	De.S	M.So	M.So
Less.4					
Less.5	M.So	De.S	De.S	M.So	
Less.6	De.S	De.S	De.S	M.So	
Less.7	De.S	M.So	M.So	M.So	
Less.8	De.S & M.So			M.So	

INT07	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Se.M	Se.M	Se.M	Se.M
Less.2	Se.M	Se.M	Se.M	Se.M	Ay.I
Less.3	Se.M	Ay.I	Ay.I	Ay.I	Ay.I
Less.4					
Less.5	Ay.I	Ay.I	Ay.I	Ay.I	
Less.6	Ay.I	Ay.I	Ay.I	Ay.I	
Less.7	Ay.I	Ay.I & Se.M	Ay.I	Ay.I	
Less.8	Ay.I			Ay.I	

INT08	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		Ih.B	Sh.T	Ih.B	Sh.T
Less.2	Dummy	Ih.B	Sh.T	Ih.B	Ih.B
Less.3	Sh.T	Sh.T	Sh.T	Ih.B	Ih.B
Less.4					
Less.5	Sh.T	Sh.T	Ih.B	Sh.T	
Less.6	Sh.T	Sh.T	Ih.B	Sh.T	
Less.7	Sh.T	Sh.T	Ih.B & Dummy	Sh.T	
Less.8	Sh.T			Sh.T	

INT09	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		Fa.O	M.Ay	M.Ay	Fa.O
Less.2	Fa.O	Fa.O	M.Ay	M.Ay	Fa.O
Less.3	Fa.O	Fa.O	M.Ay	Fa.O	Fa.O
Less.4					
Less.5	Fa.O	M.Ay	Fa.O	Fa.O	
Less.6	M.Ay	M.Ay	Fa.O	Fa.O	
Less.7	M.Ay	Dummy	Fa.O	Fa.O	
Less.8	M.Ay & Dummy			Fa.O	

INT10R	Mon.	Tue.	Wed.	Thu.	Fri.
Less.1		M.Ay	Tu.S	Tu.S	Tu.S
Less.2	M.Ay	M.Ay	Tu.S	Tu.S	Tu.S
Less.3	M.Ay	Tu.S	Tu.S	Tu.S	M.Ay
Less.4					
Less.5	Tu.S	Tu.S	M.Ay	Tu.S	
Less.6	Tu.S	Tu.S	M.Ay	M.Ay	
Less.7	Tu.S	M.Ay & Tu.S	Tu.S	M.Ay	
Less.8	Tu.S			Tu.S	

INT11R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		De.S	De.S	Ay.Y	De.S
Less.2	De.S	De.S	De.S	Ay.Y	Ay.Y
Less.3	De.S	Ay.Y	Ay.Y	Ay.Y	Ay.Y
Less.4					
Less.5	Ay.Y	Ay.Y	Ay.Y	Ay.Y	
Less.6	Ay.Y	Ay.Y	Ay.Y	De.S	
Less.7	Ay.Y	Ay.Y & De.S	Ay.Y	De.S	
Less.8	Ay.Y			Ay.Y	

INT12R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Be.A	Be.A	Be.A	Ex.A
Less.2	Dummy	Be.A	Be.A	Be.A	Ex.A
Less.3	Ex.A	Be.A	Be.A	Be.A	Ex.A
Less.4					
Less.5	Ex.A	Ex.A	Ex.A	Ex.A	
Less.6	Ex.A	Ex.A	Ex.A	Ex.A	
Less.7	Ex.A	Ex.A	Ex.A	Ex.A	
Less.8	Ex.A & Dummy			Ex.A	

UPP01	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Ar.O	S.U.Y	S.U.Y	S.U.Y
Less.2	S.U.Y	Ar.O	S.U.Y	S.U.Y	S.U.Y
Less.3	S.U.Y	S.U.Y	Ar.O	S.U.Y	S.U.Y
Less.4	S.U.Y	S.U.Y	Ar.O	S.U.Y	
Less.5					
Less.6	S.U.Y	Ar.O & S.U.Y	S.U.Y	S.U.Y	
Less.7	S.U.Y				
Less.8					

UPP02	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Ip.E	Ar.O	Ip.E	Ip.E
Less.2	Ip.E	Ip.E	Ar.O	Ip.E	Ar.O
Less.3	Ip.E	Ip.E	Ip.E	Ip.E	Ar.O
Less.4	Ip.E	Ip.E	Ip.E	Ip.E	
Less.5					
Less.6	Ip.E	Ip.E	Ar.O & Ip.E	Ip.E	
Less.7	Ip.E				
Less.8					

UPP03	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Ze.G	A.A.S	Ze.G	Ze.G
Less.2	Ze.G	Ze.G	A.A.S	Ze.G	Ze.G
Less.3	Ze.G	A.A.S	Ze.G	Ze.G	Ze.G
Less.4	Ze.G	A.A.S	Ze.G	Ze.G	
Less.5					
Less.6	Ze.G	A.A.S & Ze.G	Ze.G	Ze.G	
Less.7	Ze.G				
Less.8					

UPP04	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		A.Ce	M.In	A.Ce	A.Ce
Less.2	A.Ce	A.Ce	M.In	A.Ce	M.In
Less.3	A.Ce	A.Ce	A.Ce	A.Ce	M.In
Less.4	A.Ce	A.Ce	A.Ce	A.Ce	
Less.5					
Less.6	A.Ce	A.Ce & M.In	A.Ce	A.Ce	
Less.7	A.Ce				
Less.8					

UPP05R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Da.G	Da.G	M.In	Da.G
Less.2	Da.G	Da.G	Da.G	M.In	Da.G
Less.3	Da.G	Da.G	M.In	Da.G	Da.G
Less.4	Da.G	Da.G	M.In	Da.G	
Less.5					
Less.6	Da.G	Da.G	Da.G & M.In	Da.G	
Less.7	Da.G				
Less.8					

UPP06R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Ce.G	Ce.G	E.M	Ce.G
Less.2	Ce.G	Ce.G	Ce.G	E.M	Ce.G
Less.3	Ce.G	Ce.G	E.M	Ce.G	Ce.G
Less.4	Ce.G	Ce.G	E.M	Ce.G	
Less.5					
Less.6	Ce.G	Ce.G & E.M	Ce.G	Ce.G	
Less.7	Ce.G				
Less.8					

UPP07R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		Bu.S	E.M	Bu.S	Bu.S
Less.2	Bu.S	Bu.S	E.M	Bu.S	E.M
Less.3	Bu.S	Bu.S	Bu.S	Bu.S	E.M
Less.4	Bu.S	Bu.S	Bu.S	Bu.S	
Less.5					
Less.6	Bu.S	Bu.S	Bu.S	Bu.S	
Less.7	Bu.S & E.M				
Less.8					

UPP08R	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		A.A.S	He.M	A.A.S	He.M
Less.2	He.M	A.A.S	He.M	A.A.S	He.M
Less.3	He.M	He.M	He.M	He.M	He.M
Less.4	He.M	He.M	He.M	He.M	
Less.5					
Less.6	He.M	He.M	A.A.S & He.M	He.M	
Less.7	He.M				
Less.8					

PFAC01	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		H.D	H.K	H.K	H.K
Less.2	H.D	H.D	H.K	H.K	H.K
Less.3	H.D	H.K	H.K	H.K	H.K
Less.4	H.K	H.K	H.K	H.K	
Less.5					
Less.6	H.K	H.D & H.K	H.K	H.K	
Less.7	H.K				
Less.8					

PFAC02	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		M.J.J	H.D	H.D	M.J.J
Less.2	M.J.J	M.J.J	H.D	H.D	M.J.J
Less.3	M.J.J	M.J.J	M.J.J	M.J.J	M.J.J
Less.4	M.J.J	M.J.J	M.J.J	M.J.J	
Less.5					
Less.6	M.J.J	M.J.J	M.J.J	M.J.J	
Less.7	H.D & M.J.J				
Less.8					

PFAC03	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		R.V	R.V	R.V	R.V
Less.2	R.V	R.V	R.V	R.V	H.D
Less.3	R.V	R.V	R.V	H.D	H.D
Less.4	R.V	R.V	R.V	H.D	
Less.5					
Less.6	R.V	R.V	H.D & R.V	R.V	
Less.7	R.V				
Less.8					

PFAC04	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		D.S	D.S	O.O.D	D.S
Less.2	D.S	D.S	D.S	O.O.D	D.S
Less.3	D.S	D.S	O.O.D	D.S	D.S
Less.4	D.S	D.S	O.O.D	D.S	
Less.5					
Less.6	D.S	D.S	D.S	D.S	
Less.7	D.S & O.O.D				
Less.8					

PFAC05	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		M.A	M.A	G.C	G.C
Less.2	G.C	M.A	M.A	G.C	G.C
Less.3	G.C	G.C	G.C	G.C	G.C
Less.4	G.C	G.C	G.C	G.C	
Less.5					
Less.6	G.C	G.C & M.A	G.C	G.C	
Less.7	G.C				
Less.8					

PFAC06	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		B.L	O.O.D	B.L	B.L
Less.2	B.L	B.L	O.O.D	B.L	B.L
Less.3	B.L	B.L	B.L	O.O.D	B.L
Less.4	B.L	B.L	B.L	O.O.D	
Less.5					
Less.6	B.L	B.L & O.O.D	B.L	B.L	
Less.7	B.L				
Less.8					

PFAC07	**Mon.**	**Tue.**	**Wed.**	**Thu.**	**Fri.**
Less.1		E.M	L.B	L.B	L.B
Less.2	E.M	E.M	L.B	L.B	L.B
Less.3	E.M	L.B	L.B	L.B	L.B
Less.4	L.B	L.B	L.B	L.B	
Less.5					
Less.6	L.B	L.B	E.M & L.B	L.B	
Less.7	L.B				
Less.8					

Bibliography

[1] R. Lewis. A survey of metaheuristic-based techniques for university timetabling problems. *OR spectrum*, 30(1):167–190, 2008.

[2] A. A. Pol and R. K. Ahuja. Developing web-enabled decision support systems. *Dynamic Ideas, Belmont*, 2007.

[3] T. Çetinyokuş and H. Gökçen. Borsada göstergelerle teknik analiz için bir karar destek sistemi. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 17(1), 2002.

[4] V. A. Bardadym. Computer-aided school and university timetabling: The new wave. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 22–45. Springer, 1995.

[5] E. Burke, J. Kingston, and D. De Werra. Applications to timetabling. *Handbook of graph theory*, 445, 2004.

[6] E. Burke and P. Ross. *Practice and Theory of Automated Timetabling: First International Conference, Edinburgh, UK, August 29-September 1, 1995. Selected Papers*, volume 1153. Springer Science & Business Media, 1996.

[7] N. Pillay. A survey of school timetabling research. *Annals of Operations Research*, 218(1):261–293, 2014.

[8] E. Burke and S. Petrovic. Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2):266–280, 2002.

[9] D. de Werra. An introduction to timetabling. *European journal of operational research*, 19(2):151–162, 1985.

[10] S. Petrovic and E. Burke. University timetabling. *Handbook of scheduling: algorithms, models, and performance analysis*, 45:1–23, 2004.

[11] A. Schaerf. A survey of automated timetabling. *Artificial intelligence review*, 13(2): 87–127, 1999.

[12] M. W. Carter and G. Laporte. Recent developments in practical course timetabling. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 3–19. Springer, 1997.

[13] R. H. McClure and C. E. Wells. A mathematical programming model for faculty course assignments. *Decision Sciences*, 15(3):409–420, 1984.

[14] A. Gunawan, K. M. Ng, and K. L. Poh. Solving the teacher assignment-course scheduling problem by a hybrid algorithm. *Int. J. Comput. Inform. Engin*, 1(2): 137–142, 2007.

[15] M. Wright. School timetabling using heuristic search. *Journal of the Operational Research Society*, 47(3):347–357, 1996.

[16] R. Alvarez-Valdes, E. Crespo, and J. M. Tamarit. Assigning students to course sections using tabu search. *Annals of Operations Research*, 96(1-4):1–16, 2000.

[17] C. R. Glassey and M. Mizrach. A decision support system for assigning classes to rooms. *Interfaces*, 16(5):92–100, 1986.

[18] J. J. Dinkel, J. Mote, and M. Venkataramanan. Or practice-an efficient decision support system for academic course scheduling. *Operations Research*, 37(6):853–864, 1989.

[19] S. D. Bloomfield and M. M. McSharry. Preferential course scheduling. *Interfaces*, 9 (4):24–31, 1979.

[20] S. M. Al-Yakoob, H. D. Sherali, and M. Al-Jazzaf. A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations. *European Journal of Operational Research*, 180(3):1028–1044, 2007.

[21] T. Cura. Timetabling of faculty lectures using simulated annealing algorithm. 2007.

[22] S. C. Sarin, Y. Wang, and A. Varadarajan. A university-timetabling problem and its solution using benders' partitioning-a case study. *Journal of Scheduling*, 13(2): 131–141, 2010.

[23] U. Ü. Uçar. Belirsizlik altında dinamik ders programi Çizelgeleme: Gerçek bir uygulama. Master's thesis, Gazi Üniversitesi, 2015.

[24] S. Abdullah and H. Turabieh. On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems. *Information Sciences*, 191:146–168, 2012.

[25] C. W. Fong, H. Asmuni, B. McCollum, P. McMullan, and S. Omatu. A new hybrid imperialist swarm-based optimization algorithm for university timetabling problems. *Information Sciences*, 283:1–21, 2014.

[26] E. H. Kampke, W. de Souza Rocha, M. C. S. Boeres, and M. C. Rangel. A grasp algorithm with path relinking for the university courses timetabling problem. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 3(2), 2015.

[27] E. K. Burke, J. Marecek, A. J. Parkes, and H. Rudová. Penalising patterns in timetables: Novel integer programming formulations. In *Operations Research Proceedings 2007*, pages 409–414. Springer, 2008.

[28] M. Eley. Ant algorithms for the exam timetabling problem. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 364–382. Springer, 2006.

[29] R. Martí, H. Lourenço, and M. Laguna. Assigning proctors to exams with scatter search. In *Computing Tools for Modeling, Optimization and Simulation*, pages 215–227. Springer, 2000.

[30] Z. K. Ozturk, G. Ozturk, and M. Sagir. An automated multi-objective invigilator-exam assignment system. *International Journal of Information Technology & Decision Making*, 9(02):223–238, 2010.

[31] Salem M Al-Yakoob, Hanif D Sherali, and Mona Al-Jazzaf. A mixed-integer mathematical modeling approach to exam timetabling. *Computational Management Science*, 7(1):19–46, 2010.

[32] M. M. Kahar and G. Kendall. The examination timetabling problem at universiti malaysia pahang: Comparison of a constructive heuristic with an existing software solution. *European journal of operational research*, 207(2):557–565, 2010.

[33] C. B. Kalayci and A. Gungor. A genetic algorithm based examination timetabling model focusing on student success for the case of the college of engineering at pamukkale university, turkey. *Gazi University Journal of Science*, 25(1):137–153, 2012.

[34] L. De Boeck, J. Beliën, and S. Creemers. A column generation approach for solving the examination-timetabling problemauthor-name: Woumans, gert. *European Journal of Operational Research*, 253(1):178–194, 2016.

[35] M. Battistutta, A. Schaerf, and T. Urli. Feature-based tuning of single-stage simulated annealing for examination timetabling. *Annals of Operations Research*, pages 1–16, 2014.

[36] N. Pillay and W. Banzhaf. An informed genetic algorithm for the examination timetabling problem. *Applied Soft Computing*, 10(2):457–467, 2010.

[37] C. Gogos, P. Alefragis, and E. Housos. An improved multi-staged algorithmic process for the solution of the examination timetabling problem. *Annals of Operations Research*, 194(1):203–221, 2012.

[38] B. McCollum, P. McMullan, A. J. Parkes, E. K. Burke, and R. Qu. A new model for automated examination timetabling. *Annals of Operations Research*, 194(1): 291–315, 2012.

[39] G. N. Beligiannis, C. Moschopoulos, and S. D. Likothanassis. Applying evolutionary computation to the school timetabling problem: The greek case. *Computers & Operations Research*, 35(4):1265–1280, 2008.

[40] D. Abramson. Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Management science*, 37(1):98–113, 1991.

[41] R. Alvarez-Valdes, G. Martin, and J. Tamarit. Constructing good solutions for the spanish school timetabling problem. *Journal of the Operational Research Society*, 47(10):1203–1215, 1996.

[42] P. Wilke and J. Ostler. Solving the school timetabling problem using tabu search, simulated annealing, genetic and branch & bound algorithms, 2010.

[43] G. Post, S. Ahmadi, S. Daskalaki, J. H. Kingston, J. Kyngas, C. Nurmi, and D. Ranson. An xml format for benchmarks in high school timetabling. *Annals of Operations Research*, 194(1):385–397, 2012.

[44] F. Carmusciano and D. D. L. Cardillo. A simulated annealing with tabu list algorithm for the school timetable problem. In *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling (ICPTAT'95)*, pages 231–243, 1995.

[45] L.-F. Kwok, S.-C. Kong, and Y.-Y. Kam. Timetabling in hong kong secondary schools. *Computers & Education*, 28(3):173–183, 1997.

[46] L.-F. Kwok and S.-C. Kong. A conceptual model of knowledge-based time-tabling system. *Knowledge-Based Systems*, 12(3):81–93, 1999.

[47] M. Veenstra and I. F. Vis. School timetabling problem under disturbances. *Computers & Industrial Engineering*, 95:175–186, 2016.

[48] G. N. Beligiannis, C. N. Moschopoulos, G. P. Kaperonis, and S. D. Likothanassis. A genetic algorithm approach to school timetabling. *Journal of the Operational Research Society*, 60(1):23–42, 2009.

[49] P. Avella, B. D'Auria, S. Salerno, and I. Vasil'ev. A computational study of local search algorithms for italian high-school timetabling. *Journal of Heuristics*, 13(6): 543–556, 2007.

[50] G. Post, J. H. Kingston, S. Ahmadi, S. Daskalaki, C. Gogos, J. Kyngas, C. Nurmi, C. Nurmi, N. Musliu, N. Pillay, H. Santos, et al. Xhstt: an xml archive for high school timetabling problems in different countries. *Annals of Operations Research*, 218(1):295–301, 2014.

[51] A. Kheiri, E. Özcan, and A. J. Parkes. A stochastic local search algorithm with adaptive acceptance for high-school timetabling. *Annals of Operations Research*, pages 1–17, 2014.

[52] G. H. Fonseca and H. G. Santos. Variable neighborhood search based algorithms for high school timetabling. *Computers & Operations Research*, 52:203–208, 2014.

[53] Á. P. Dorneles, O. C. de Araújo, and L. S. Buriol. A fix-and-optimize heuristic for the high school timetabling problem. *Computers & Operations Research*, 52:29–38, 2014.

[54] M. W. Carter, G. Laporte, and S. Y. Lee. Examination timetabling: Algorithmic strategies and applications. *Journal of the Operational Research Society*, 47(3): 373–383, 1996.

[55] W. L. Winston and J. B. Goldberg. *Operations research: applications and algorithms*, volume 3. Duxbury press Belmont, CA, 2004.

[56] E. A. Akkoyunlu. A linear algorithm for computing the optimum university timetable. *The Computer Journal*, 16(4):347–350, 1973.

[57] J. Csima and C. Gotlieb. Tests on a computer method for constructing school timetables. *Communications of the ACM*, 7(3):160–163, 1964.

[58] N. L. Lawrie. An integer linear programming model of a school timetabling problem. *The Computer Journal*, 12(4):307–316, 1969.

[59] A. Tripathy. A lagrangean relaxation approach to course timetabling. *Journal of the Operational Research Society*, 31(7):599–603, 1980.

[60] M. Dimopoulou and P. Miliotis. Implementation of a university course and examination timetabling system. *European Journal of Operational Research*, 130(1): 202–213, 2001.

[61] M. S. Ozdemir and R. N. Gasimov. The analytic hierarchy process and multiobjective 0–1 faculty course assignment. *European Journal of Operational Research*, 157 (2):398–408, 2004.

[62] N. A. Ismayilova, M. Sagir, and R. N. Gasimov. A multiobjective faculty–course–time slot assignment problem with preferences. *Mathematical and Computer Modelling*, 46(7):1017–1029, 2007.

[63] S. M. Al-Yakoob and H. D. Sherali. Mathematical programming models and algorithms for a class–faculty assignment problem. *European Journal of Operational Research*, 173(2):488–507, 2006.

[64] N. Boland, B. D. Hughes, L. T. Merlot, and P. J. Stuckey. New integer linear programming approaches for course timetabling. *Computers & Operations Research*, 35(7):2209–2233, 2008.

[65] S. Ribiu and S. Konjicija. A two phase integer linear programming approach to solving the school timetable problem. In *Proceedings of the ITI*, pages 21–24, 2010.

[66] S. Kristiansen and T. R. Stidsen. A comprehensive study of educational timetabling-a survey. Technical report, Department of Management Engineering, Technical University of Denmark, 2013.

[67] M. Sørensen and F. H. Dahms. A two-stage decomposition of high school timetabling applied to cases in denmark. *Computers & Operations Research*, 43:36–49, 2014.

[68] A. E. Phillips, C. G. Walker, M. Ehrgott, and D. M. Ryan. Integer programming for minimal perturbation problems in university course timetabling. *Annals of Operations Research*, pages 1–22, 2016.

[69] J. Havås, A. Olsson, J. Persson, and M. S. Schierscher. Modeling and optimization of university timetabling-a case study in integer programming. 2013.

[70] J. Peck and M. Williams. Algorithm 286: Examination scheduling. *Communications of the ACM*, 9(6):433–434, 1966.

[71] D. J. Welsh and M. B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85–86, 1967.

[72] G. Laporte and S. Desroches. Examination timetabling by computer. *Computers & Operations Research*, 11(4):351–360, 1984.

[73] L. F. Lai, C.-C. Wu, N.-L. Hsueh, L.-T. Huang, and S.-F. Hwang. An artificial intelligence approach to course timetabling. *International Journal on Artificial Intelligence Tools*, 17(01):223–240, 2008.

[74] L. T. Merlot, N. Boland, B. D. Hughes, and P. J. Stuckey. A hybrid algorithm for the examination timetabling problem. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 207–231. Springer, 2002.

[75] B. Bullnheimer. An examination scheduling model to maximize students' study time. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 78–91. Springer, 1997.

[76] E. Burke, Y. Bykov, J. Newall, and S. Petrovic. A time-predefined local search approach to exam timetabling problems. *Iie Transactions*, 36(6):509–528, 2004.

[77] E. Burke, D. Elliman, and R. Weare. A genetic algorithm based university timetabling system. In *Proceedings of the 2nd east-west international conference on computer technologies in education*, volume 1, pages 35–40, 1994.

[78] E. K. Burke, J. P. Newall, and R. F. Weare. A memetic algorithm for university exam timetabling. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 241–250. Springer, 1995.

[79] D. Abramson and J. Abela. A parallel genetic algorithm for solving the school time tabling problem. 15 australian computer science conference. *Hobert, Feb*, 1992.

[80] O. Rossi-Doria, C. Blum, J. Knowles, M. Sampels, K. Socha, and B. Paechter. A local search for the timetabling problem. In *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling, PATAT*, pages 124–127, 2002.

[81] Z. N. Azimi. Hybrid heuristics for examination timetabling problem. *Applied Mathematics and Computation*, 163(2):705–733, 2005.

[82] K. Dowsland and J. Thompson. Ant colony optimization for the examination scheduling problem. *Journal of the Operational Research Society*, 56(4):426–438, 2005.

[83] M. M. Seref, R. K. Ahuja, and W. L. Winston. *Developing spreadsheet-based decision support systems.* Dynamic Ideas, 2007.

[84] R. H. Sprague. A framework for the development of decision support systems. *MIS quarterly*, pages 1–26, 1980.

[85] P. G. Keen, M. Scott, et al. Decision support systems; an organizational perspective. Technical report, 1978.

[86] J. S. Dyer. A time-sharing computer program for the solution of the multiple criteria problem. *Management Science*, 19(12):1379–1383, 1973.

[87] M. M. Koksalan, J. Wallenius, and S. Zionts. *Multiple criteria decision making: from early history to the 21st century.* World Scientific, 2011.

[88] A. Ishizaka and P. Nemery. *Multi-criteria decision analysis: methods and software.* John Wiley & Sons, 2013.

[89] H. G. Sol, A. T. Cees, P. F. de Vries Robbé, et al. *Expert systems and artificial intelligence in decision support systems: proceedings of the Second Mini Euroconference, Lunteren, The Netherlands, 17–20 November 1985.* Springer Science & Business Media, 2013.

[90] Y. M. Lee and E. J. Chen. Basf uses a framework for developing web-based production-planning-optimization tools. *Interfaces*, 32(6):15–24, 2002.

[91] J. Miranda. eclasskeduler: a course scheduling system for the executive education unit at the universidad de chile. *Interfaces*, 40(3):196–207, 2010.

[92] L. R. Foulds and D. G. Johnson. Slotmanager: a microcomputer-based decision support system for university timetabling. *Decision Support Systems*, 27(4):367–381, 2000.

[93] J. Miranda, P. A. Rey, and J. M. Robles. udpskeduler: A web architecture based decision support system for course and classroom scheduling. *Decision Support Systems*, 52(2):505–513, 2012.

[94] E. Şuşnea. Improving decision making process in universities: a conceptual model of intelligent decision support system. *Procedia-Social and Behavioral Sciences*, 76: 795–800, 2013.

[95] T.-P. Liang, C.-C. Lee, and E. Turban. Model management and solvers for decision support. In *Handbook on Decision Support Systems 1*, pages 231–258. Springer, 2008.

[96] S. C. Brailsford, C. N. Potts, and B. M. Smith. Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119(3): 557–581, 1999.

[97] G. White. Constrained satisfaction, not so constrained satisfaction and the timetabling problem. In *A Plenary Talk in the Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling, University of Applied Sciences, Konstanz*, pages 32–47, 2000.