EXAMINATION TIMETABLING PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

By
Berk Şahin
July 2021

EXAMINATION TIMETABLING PROBLEM
By Berk Şahin
July 2021

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

Oya Karaşan(Advisor)

Alper Şen

Serpil Erol

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

EXAMINATION TIMETABLING PROBLEM

Berk Şahin
M.S. in Industrial Engineering
Advisor: Oya Karaşan
July 2021

Examination timetabling is a major challenge in most educational institutions. Since the underlying problem is NP-hard and real-life problems are too hard to solve to optimality, heuristic approaches are adopted as solution methodologies in general. The significance of a fair exam schedule creates a need for exact solutions to the examination timetabling problem. In this thesis, we mainly focus on exact solution approaches for this problem and test their efficacy on well-known benchmark problems from the literature as well as on Bilkent University's data. Existing formulations used for the p-hub median hub location problem and the quadratic assignment problem in the literature are adapted to the examination timetabling problem and various Bender's decomposition and branch and cut methodologies are tailored to these formulations. A novel compact formulation based on individual student schedules with reduced model dimensions is proposed. For the literature instances in which optimal values are not known, we could find effective lower bounds. For higher dimensions, we propose matheuristic approaches based on our proposed formulations. With this study, effective lower bounds are found for unsolved problems and small-scale problems are solved to optimality in short computational times.

Keywords: Examination scheduling, mixed integer programming.

ÖZET

SINAV ZAMAN ÇİZELGELEME PROBLEMİ

Berk Şahin
Endüstri Mühendisliği, Yüksek Lisans
Tez Danışmanı: Oya Karaşan
Temmuz 2021

Sınav zaman çizelgeleme problemi birçok eğitim kurumu tarafından büyük bir sorundur. İlgilenilen problem NP-zor olduğu için ve gerçek problemlerin eniyilenmesi zor olduğu için, birçok akademik çalışma bu problemin çözümü için bulgusal yöntemler kullanmaktadır. Adil bir sınav takviminin önemi sınav zaman çizelgeme problemi için kesin çözümleri gerekli kılmaktadır. Bu çalışmada, bu problem için kesin çözüm yöntemleri araştırılmış ve bu yöntemlerin yeterliği literatürdeki problemler ve Bilkent Üniversitesi problemi üzerinde denenmiştir. Tez kapsamında, p-merkez ortanca aktarma merkezi ve ikinci dereceden atama problemleri kaynaklarındaki formulasyonlar sınav zaman çizelgeleme problemine uyarlanmıştır ve bu formulasyonlara farklı Bender's ayrıştırma metodu ve dal ve kesme metotları uygulanmıştır. Az sayıda zaman aralığı içeren problemler için öğrencilerin potansiyel takvimlerini kullanan yeni bir formulasyon tanımlanmıştır. Büyük ölçekli problemlerde uygulanmak için alt sınır hesaplama metotları ve önerilen formulasyonlara dayalı bulgusal yaklaşımlar önerilmiştir. Çalışmanın sonucunda çözülemeyen problemler için etkili alt sınırlar bulunmuş ve küçük boyutlu problemler kısa zamanda kesin olarak çözülebilmiştir.

Anahtar sözcükler: Sınav çizelgeleme, karma tamsayılı programlama.

Acknowledgement

First of all, I would like to express my deep gratitude to my thesis supervisor Prof. Oya Karaşan for her continuous support, encouragement, and patience. Starting from the very early times of my academic journey, I always feel her guidance in every part of my studies. I am very proud to study under her supervision.

I am genuinely grateful to Prof. Alper Şen and Prof. Serpil Erol for accepting to be on my thesis committee and sharing their valuable recommendations.

I would like to thank all faculty members, staff, and friends that I met in my 7-year Bilkent experience. I spent the best times of my life here. I will always be proud to be a Bilkent alumnus.

I am forever indebted to my parents, Nazan and Ramazan Şahin, for giving me the opportunities and experiences that have made me who I am. I am profoundly grateful for their endless support, motivation, and love. Without them, I would never accomplish my life goals. I would also like to thank my brother and future colleague Mert Şahin for his motivation and invaluable friendship. I wish him a beautiful and productive university life in Bilkent University Industrial Engineering department.

Finally, I would like to thank my high school friends Orkun Ege Köse, Eren Aksun, Ahmet Burak Baytok, and Semih Ersan for their lifelong friendship. I would also like to thank my friends from university, Nida Demirel, Kemal Canpolat, and Sinem Yaşa for their invaluable friendship. I would also like to thank Barış Bilir for his motivation during my thesis writing process.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Examination Timetabling Problem

Examination timetabling is a challenge that has to be faced in many universities and schools. The examination timetabling problem was first defined by Carter et al. [1] with 13 real-world instances. In the examination timetabling problem, it is aimed to assign all the exams to a given number of timeslots such that no conflicting exam pairs are assigned to the same timeslot. A conflicting exam pair consists of two exams that are jointly taken by at least one student. Examination timetabling problem has some hard and soft constraints. As a typical example of a hard constraint, it is forbidden to assign two conflicting exams to the same timeslot to guarantee that a student will not have two exams at the same time. In addition to the conflicting exam constraint, there might be various hard constraints such as:

• prohibiting the exams of the courses given by the same instructor to be assigned to the same timeslot,
• limiting assignment of specific exams to specific timeslots,
• prohibiting consecutive exams for any student,
• prohibiting more than a specific number of exams for any student on a particular day, etc.

As Qu et al. [2] stated, in general, a solution satisfying all hard constraints is called a feasible solution. In addition to hard constraints, there are some soft constraints, which are aimed to be satisfied as much as possible in the assignment. As can be seen in Toronto dataset [1], Nottingham dataset [3] and Yeditepe dataset [4], these soft constraints are generally imposed so as to spread the exams as much as possible for any student so that students have enough time to prepare for their exams. Soft constraints are included in the problem with a penalty cost, and the aim of the problem is to schedule the exams such that all hard constraints are satisfied and the objective function representing the penalty for the violation of soft constraints is minimized.

The examination timetabling problem is a challenging issue for most educational institutions due to the size of the problem even if there are reasonable numbers of students and courses. It is shown by Even et al. [5] that the examination timetabling problem is NP-complete. Therefore, most real-life instances are hard to solve to optimality and ultimately, they are handled by heuristic methods. Since the number of works related to exact solutions of examination timetabling is limited, there might be a need for a study to solve some of the small-sized real-life problems exactly. Since a fair schedule for all students is crucial for a fair educational system, solving the examination timetabling problem becomes a necessity for educational institutions. Moreover, the need for solving real-life problems increases the significance of well-developed exact solution methodologies. Developing exact solution methodologies might require a long-time research. With this motivation, we focus on finding exact solutions for relatively small-scale examination timetabling problems as a first step to construct more developed approaches.

1.2 Bilkent University Examination Timetabling Problem

In Bilkent University, final exams are assigned to timeslots like a typical examination timetabling problem but considering some extra hard and soft constraints. Bilkent University examination timetabling problem is one of the largest timetabling problems in the literature, therefore, it is very challenging to find exact solutions to the problem.

Through this thesis, in addition to different methodologies to solve the examination timetabling problem exactly, Bilkent University problem will be added to the literature. Since the number of courses, the number of timeslots, and some constraints might change from semester to semester, we have used the data of the 2018 Spring semester for our experimentation throughout this thesis. Bilkent University problem has specific hard and soft constraints. Hard constraints of Bilkent University problem are described below:

- no conflicting exams are allowed,
- since some instructors can go abroad to attend conferences each course has a maximum and a minimum timeslot number and it is not allowed to assign exams to timeslots that do not satisfy maximum and minimum timeslot number requirements.
- a student cannot have more than two exams in a day to provide a fair schedule.
- some crowded courses are pre-assigned to some specific timeslots (mostly the earliest ones) to provide an effective schedule.
- some timeslots must not be used due to holidays.

During the 2018 Spring semester, Bilkent University had 6560 students jointly taking 565 courses taught by 570 instructors. The exam period was 14 days with 4 slots per day. 32 courses were pre-assigned, 50 had maximum slot and 21 had minimum slot requirements.

To include the soft constraints in the model, a cost is incurred for each student in the objective function for consecutive exams, exams on the same day, and exams on consecutive days. These costs also differ by the hardness of the exams of courses. Some courses such as most curriculum courses are viewed as harder than elective courses. The categorization of "hard" and "easy" exams of the courses is determined by the departments offering the courses. It is considered that a hard exam will require more time to prepare for than an easy one. Table 1.1 gives costs incurred for each soft constraint and each hardness type of exam per student.

Course	Consecutive timeslot	Same day	Consecutive Day
Hard&Hard	24	20	10
Hard&Easy	22	16	6
Easy&Easy	20	12	4

Table 1.1: Costs used in Bilkent University's problem's objective function

The coefficients described in the table are added to the objective function for each course pair and for each student taking both courses in the pair. For instance, if a student is taking two courses where one of them is hard and the other is easy, and exams of these two courses are scheduled consecutively within a day, the cost incurred for the course pair and the student is 22. Cost values are specific to Bilkent University's problem, and these values might change among institutions. The sum of costs incurred for all students forms the objective function of the Bilkent University examination timetabling problem. The aim of the problem is to minimize the total cost incurred due to the violation of soft constraints while ensuring the satisfaction of hard constraints.

1.3 Overview

In this thesis, we used a span of methodologies that have been applied to different problems in the literature to exactly solve real-world examination timetabling instances. This is the main focus of the study. We used methodologies applied to the p-hub median hub location problem and the quadratic assignment problem. We tried to solve some benchmark instances in the literature and Bilkent University problem. In addition to that, some methodologies are used to find effective lower bounds to some benchmark instances and a matheuristic method that is currently being used to solve Bilkent University problem is formalized and proposed as a method.

Chapter 2 gives a detailed literature review of the studies about examination timetabling. In this chapter, firstly, various datasets are explained. Then, different solution approaches in the examination timetabling literature are described. Finally, solution approaches for the p-hub median hub location problem and the quadratic assignment problem are stated and the similarity between the examination timetabling problem and these problems is highlighted.

In Chapter 3, a typical four-indexed model that is used to solve many problems that have a quadratic objective is introduced for the examination timetabling problem. A four-indexed variable is used to represent the assignment of two items to two possible locations. The problem is proposed as a compact model, and various Benders' decomposition strategies are applied to the proposed model.

In Chapter 4, two different flow-based models used for the p-hub median hub location problem are examined. In the p-hub median hub location problem, specific nodes are selected as hubs and other nodes are assigned to these hubs to decrease the cost incurred for the transfer between nodes. This problem is similar to the examination timetabling problem except that in the p-hub median hub location problem, hubs are selected by the model, whereas in the examination timetabling problem timeslots (the object of the model that is similar to hubs) are known as an input. Two models are proposed in which one of them has two-indexed variables and the other has three-indexed ones. In this chapter, Bender's decomposition is applied to these models.

In Chapter 5, two methodologies in the quadratic assignment literature, Kauffmann-Broeckx formulations and Gilmore-Lawler bounds are explained. In this chapter, some new models based on these two methodologies and their Bender's decompositions are applied to real-world timetabling problems. In the quadratic assignment problem, facilities are allocated to locations to decrease the quadratic cost incurred due to the transfer between facilities. The quadratic assignment problem and the examination timetabling problem have many similar properties. The only difference between these two problems is that in the quadratic assignment problem there is a one-to-one allocation between facilities and locations but in the examination timetabling problem many exams can be assigned to the same timeslot.

In Chapter 6, a combination and a permutation formulation for examination timetabling problems with a small number of timeslots are introduced. These formulations are derived using the potential schedules of each student. The permutations and combinations of courses are used in the model to represent the schedule of each student, and a compact model is proposed. In addition, some dominance relations among students' course lists are used in order to come up with valid inequalities. Such dominance relations allow us to decrease the dimensions of the problems.

In Chapter 7, some lower bound calculation methods and the matheuristic used to solve the Bilkent University examination timetabling problem are introduced. The proposed lower bounds are provided by dividing the problem into two or calculating the minimum cost for each student individually. The matheuristic is applied with assigning some exams to some given timeslots, and solving the problem with only a small proportion of exams.

Finally, in Chapter 8, the conclusion of the study and information about possible future research directions can be found.

Chapter 2

Literature Review

In the literature, very few exact and many heuristic solution methods are tried to solve examination timetabling problems. Although it is generally impossible to optimally solve real-life instances, which have many courses and timeslots leading to large-scale integer programming models, there are some exact methods utilized for small dimensional problems and typically heuristic methods are used to solve the large-scale ones.

2.1 Some Datasets in the Literature

In 1996, Carter et al. [1] introduced a set of 13 real-world examination timetabling problems from 3 Canadian high schools, 5 Canadian, 1 American, 1 UK and 1 mid-east universities. These are well-known Toronto datasets. In addition, some instances in Yeditepe datasets [4], which are problems of real instances of Yeditepe University are used to test the algorithms found in this thesis. Tables 2.1, 2.2, 2.3 describe the specifications of Bilkent dataset, Toronto dataset and Yeditepe dataset, respectively. In these tables, the number of exams, students, enrolments and conflict densities are stated. A student and one of his/her exam connections is called an enrolment. Conflict density is the ratio between the number of conflicting exam pairs that have at least one common student and the total number of exam pairs. In Yeditepe dataset, there are capacity constraints for each timeslot, and they are stated in Table 2.3.

Problem	Exams	Students	Enrolments	Conflict Density	Timeslots
Bilkent	565	6560	30914	0.88	56

Table 2.1: Bilkent 2018 Spring Semester Problem

Problem	Exams	Students	Enrolments	Conflict Density	Timeslots
EAR83	190	1125	8109	0.27	24
HEC92	81	2823	10632	0.42	18
LSE91	381	2726	10918	0.06	18
STA83	139	611	5751	0.14	13
TRE92	261	4360	14901	0.18	23
UTE92	184	2750	11793	0.08	10
YOR83	181	941	6034	0.29	21

Table 2.2: Toronto Dataset

Problem	Exams	Students	Enrolments	Conflict Density	Timeslots	Capacity
yue20011	126	559	3486	0.18	18	450
yue20023	38	420	790	0.2	6	150

Table 2.3: Yeditepe Dataset

The optimal solution values are not available for any instance of Toronto dataset. However, it is stated by Parkes and Ozcan [4] that $yue20023$ is the largest problem that can be solved exactly in Yeditepe problems. In this thesis, in addition to these problems, Bilkent University examination timetabling problem *Bilkent* and its simplified version *Bilkentx* are introduced to the literature. *Bilkentx* represents the Bilkent University problem with only x number of exams to assign, and the assignments of remaining exams are taken as an input and do not change. These fixed assignments are taken from a good solution to the problem and guaranteed to stay assigned to the same timeslots using proper constraints. In addition to the stated datasets, there are well-known Nottingham and Melbourne datasets used in the literature.

2.2 Solution Approaches for the Examination Timetabling Problem

As stated earlier, the examination timetabling problem was first introduced by Carter et al. [1]. This study introduces Toronto dataset. Many heuristic methods were used to find good feasible solutions to *Toronto* instances stated in that work. One of the methods used for examination timetabling is the memetic algorithm. The memetic algorithm is an evolutionary algorithm that uses mutation of an existing solution and applies local search to find solutions after mutation. In 1996, Burke et al. [3] used hill-climbing after each mutation and tried to solve instances in Nottingham and Toronto datasets using an evaluation function consisting of events, unscheduled exams and conflicts. In this study, some violations of some hard constraints such as scheduling all exams are permitted. Hill-climbing is a local search methodology that uses incremental changes in the solution in each iteration to find the local optima. In 1999, Burke and Newall [6] used a memetic algorithm with hill-climbing to subset problems that are created by the original problem. These studies demonstrate that good solutions for *Nottingham* and *Toronto* can be achieved by violating some hard constraints.

In 1996, Ross et al. [7] applied genetic algorithm to timetabling problems. In this study, a genetic algorithm and a stochastic hillclimber are used and compared for generated instances. In 2007, Ülker et al. [8] used linear linkage encoding (LLE) to solve *Toronto* instances. LLE is one of the methods that is used for data clustering. LLE uses arrays to create a linear path among nodes to create an encoding. In this paper, the examination timetabling problem is represented as a grouping problem and LLE is used to solve the corresponding grouping problem. The method works well on $Toronto$ instances.

Tabu search is a metaheuristic that uses local search techniques but it allows going to worse neighbors at each iteration. A list of forbidden solutions are created in the algorithm to block the repetitive moves to the same solution. In 2001, Gaspero and Schaerf [9] applied tabu search to the examination timetabling problem. $Toronto$ and $Nottingham$ problems are solved and comparisons are made. In 2004, White et al. [10] used another tabu search algorithm by using a flexible longer-term tabu list. In this way, forbidden nodes can be eliminated from the tabu list when a given number of nodes are in the list.

In 1998, Thompson and Dowsland [11] used simulated annealing and applied the algorithm to some real-life instances. Simulated annealing is a method that mimics the cooling procedure of a material. At each iteration, the temperature (probabilities) is changed and the probability of moving to a worse solution decreases in time. In 2003, Merlot et al. [12] used a three-phase algorithm with a constraint programming method to find an initial solution, a simulated annealing approach to improve the solution, and a hill-climbing method to find the best solution. This three-phase algorithm is applied to $Toronto$, $Nottingham$ and $Melbourne$ instances.

In 2005, Dowsland and Thompson [13] used an ant algorithm to solve the problem. Ant algorithms are evolutionary methods that are based on the principles of the cooperation among ants to find the closest food source. In this study, $Toronto$ instances are solved with this algorithm, and effective solutions were found.

Hyper-heuristics, which are heuristics to choose other low-level heuristics are also used for the examination timetabling. There are high-level and low-level heuristics and the high-level heuristics are used to select which low-level heuristics will be applied. In 2005, Kendall and Hussin [14] used a tabu search-based hyper-heuristic. In this paper, low-level heuristics are used with a tabu list and a tabu search decides which heuristic to be applied. *Toronto* instances were used for testing and results were given. In 2009, Qu and Burke [15] used low-level graph coloring heuristics with high-level local-search algorithms to determine the low-level heuristic to be used. The hybrid algorithm is tested on *Toronto* instances.

Lastly, in 2016, Arbaoui et al. [16] used a matheuristic to solve the examination timetabling problem. Table 2.4 gives the results for heuristic approaches in the literature that aim to minimize the cost for *Toronto* datasets. The values show the average cost per student.

	Gaspero and Schaerf [9]	White et al. [10]	Merlot et al. [12]	Kendall and Hussin [14]	Qu and Burke [15]
EAR83	45.7	45.8	35.1	40.18	35.54
HEC92	12.4	12.9	10.6	11.86	12.23
LSE91	15.5	14.7	10.5	-	12.71
STA83	160.8	158	157.3	157.38	159.2
TRE92	10.0	8.94	8.4	8.39	8.67
UTE92	29.0	29.0	25.1	27.60	30.32
YOR83	41.0	42.3	37.4	-	40.24

Table 2.4: Heuristic Results

In addition to the heuristic approaches, in 2011, exact algorithms are tried to solve the problem by McCollum et al. [17]. In the paper, three types of constraints, strict-hard, relaxable-hard and soft ones, are used. Some hard constraints are relaxed and the problem is solved without these hard constraints. When an infeasibility is detected, it is fixed by changing the assignment. A variable x_{ik} is used for the scheduling of an exam i to timeslot t. Moreover, it is stated by Parkes and Özcan [18] that the model used by McCollum et al. [17] was able to solve small instances of Yeditepe datasets.

Although there are not many exact solution methods for the examination timetabling problem, there are some for the quadratic assignment problem and the p-hub median hub location problem that might be useful to solve the examination timetabling problem since the characteristics of these problems are similar. Later sections give the literature review for these problems.

2.3 Solution Approaches for the P-Hub Median Hub Location Problem

As Ernst and Krishnamoorthy [19] stated, in the p-hub median hub location problem, given a complete graph, p nodes are selected to be hubs and other nodes are allocated to these hubs. As it is stated by Lin et al. [20] the problem aims to minimize the cost of the traffic among non-hub nodes which are assigned to hubs as transit stations and a total of p hubs are allowed. There are two variants of the p-hub median hub location problem. The first one is the single allocation version, where a non-hub node must be assigned to exactly one hub. The other one is the multiple allocation version and in this version, a non-hub node can be assigned to multiple hubs. In addition, there might be capacities for each hub for the traffic that passes through the hub. The examination timetabling problem is similar to the single allocation p-hub median hub location problem since it uses timeslots as transit stations among exams where students create the traffic among exams. In the p-hub median hub location problem, hubs are not given, whereas in the examination timetabling problem, timeslots are given and only the assignment of courses to timeslots are selected among possible assignments.

In 1996, Skorin-Kapov et al. [21] described some linear relaxations and a four-indexed formulation for the uncapacitated single and multiple allocation p-hub median problems. This formulation uses four-indexed variables to represent the assignment of two items to two hubs. The binary variable $x_{c_1c_2t_1t_2}$ is 1 if items c_1 and c_2 are assigned to locations (hubs) t_1 and t_2, and 0 otherwise. In 1996, Ernst and Krishnamoorthy [19] proposed a three-indexed flow formulation for the hub location problem. Three-indexed variables are used in this study to describe the traffic that stems from the assignment of an item to a location. These variables represent the traffic created by a specific item between its assigned location and any other location. In 2004, Labbe et al. [22] proposed a two-indexed flow formulation and used a branch and cut method to solve the problem. In this paper, a non-compact model is used and solved. According to the solution of the problem, violated constraints are added to the branch and bound scheme at each iteration. The violated constraints are added for subsets of node pairs that have traffic which creates cost and is not included in the current model. In 2018, Mokhtar et al. [23] applied Bender's decomposition to the four-indexed formulation by changing the solution values of the master problem in each iteration. Unlike a typical Bender's decomposition algorithm, at each iteration, variables with zero values are changed to small values since it is believed that it eliminates degeneracy. With this modified algorithm, some unsolved instances with 150 and 200 number of nodes were able to be solved.

2.4 Solution Approaches for the Quadratic Assignment Problem

In 1957, Koopmans and Beckmann [24] introduced the quadratic assignment problem for the first time. In the problem, n plants are assigned to n locations such that the quadratic objective function created by the traffic among plants is minimized. In the quadratic assignment problem, there is traffic among plants and this traffic goes from a plant to a location, a location to another location and the final location to another plant. The aim of the problem is to minimize the total cost of the traffic. The problem is similar to the examination timetabling problem. The only difference between two problems is that in the quadratic assignment problem, exactly one plant is assigned to exactly one location, however, in the examination timetabling problem, many exams can be assigned to the same timeslot.

In 1962-1963, Gilmore [25] and Lawler [26] presented a lower bound calculation method using single assignments. Then, in 1978, Kaufman and Broeckx [27] created a two-indexed linearization for the problem and applied a Bender's decomposition algorithm for that formulation. In 2006, Xia and Yuan [28] defined two new linearizations for the problem and used them to create two new formulations for the problem. In the work, it was proven that both formulations are stronger than Kaufman-Broeckx linearization and one of the formulations is stronger than the other. Moreover, they introduced a decomposition on the strongest model out of the three formulations. In 2012, Zhang et al. [29] introduced a new linearization that overperforms Kaufman-Broeckx formulation. In 2012, Fischetti et al. [30] suggested three significant factors to solve the quadratic assignment problem efficiently, which are taking advantage of symmetry, creating mixed-integer linear models and algorithms, and taking advantage of a facility-flow schedule by creating two sets of nodes so that the flow between two sets is minimized. The last factor is used to divide the problem, to solve two smaller problems, and to combine them to get a lower bound. In this paper, they introduced a linearization technique and two branch and cut methods, and proposed a method to find effective lower bounds by dividing the cost matrix into two. With these lower bounds, they could solve the largest optimally solved instance in the literature, which has 128 nodes.

This thesis is written in order to take the first step of finding exact solutions to the examination timetabling problem. We adapted the exact solution approaches applied to the p-hub median hub location problem and the quadratic assignment problem to the examination timetabling problem. In addition, methods that are used to find effective lower bounds for these problems are applied to the examination timetabling problem. Exact formulations might be used to solve small instances of the examination timetabling problem. For the large-sized instances, lower bound methodologies might be helpful to find a good feasible solution and to evaluate the current best solution found. We believe that development on these formulations and applications might lead to exact solutions for the real-world examination timetabling instances in future.

Chapter 3

Four-Indexed Formulation and Its Bender's Decomposition Algorithm

This chapter first gives details of the shrinking operation applied to decrease the problem's dimensions. A formulation that was applied to the p-hub median hub location problem is suggested and Bender's decomposition is applied to the suggested model. To decrease the dimension of the subproblem, the subproblem is separated and solved for each course pair that has flow (common students).

3.1 Shrinking

As a starting point, we used shrinking operation to the instances that are used in this thesis. The shrinking operation is used for Bilkent University examination timetabling problem by aggregating the students taking exactly the same courses and multiplying their weight. The same operation might be applied to other instances and it can decrease the number of variables significantly. As an example, if students $s_1, s_2 \in S$ are taking exactly the same courses, we only regard student s_1 in the formulation by setting the weight of student s_1 as 2. Through this operation, problems with smaller number of students and variables can be created. With this operation, changes in the number of students for *Toronto* and *Yeditepe* can be found in Table 3.1. Since *Bilkent* problem is given as shrunk the information of the number of students in the original instance is not known and not shown in the table.

Problem	# of students (original)	# of students after shrinking	Percent decrease
EAR83	1125	1051	6%
HEC92	2823	1499	46%
LSE91	2726	1791	34%
STA83	611	316	48%
TRE92	4360	3653	16%
UTE92	2750	861	68%
YOR83	941	910	3%
yue20011	559	341	38%
yue20023	420	175	58%

Table 3.1: The number of students in the original instances and after shrinking operation

3.2 Four-Indexed Model and Bender's Decomposition

Skorin-Kapov et al. [21] linearly formulated the p-hub median hub location problem using four-indexed $x_{c_1c_2t_1t_2}$ variables which take value 1 if demand nodes c_1, c_2 are assigned to hubs t_1, t_2, respectively. This formulation might be applied to the examination timetabling problem using the following sets, parameters, and decision variables.

Sets

- ST : set of students.
- C : set of courses.
- T : set of timeslots.
- T_c : set of timeslots that are allowed for the exam of course $c \in C$. This set is used for each course since some courses have different minimum slot, maximum slot and pre-assigned slot requirements.
- C_s : set of courses taken by student $s \in ST$.
- ST_c : set of students taking course $c \in C$.
- S : set of course pairs that have at least one common student ie. $S = \{(c_1, c_2) : c_1, c_2 \in C, c_1 < c_2, ST_{c_1} \cap ST_{c_2} \neq \emptyset\}$.

Parameters

- $a_{c_1c_2}$: the number of students taking both courses $(c_1, c_2) \in S$.
- $c_{c_1c_2t_1t_2}$: cost of assigning exams of course pairs $(c_1, c_2) \in S$ to timeslots $t_1 \in T_{c_1}, t_2 \in T_{c_2}$, respectively.

Decision Variables

- $x_{c_1c_2t_1t_2}$: 1 if course pair $(c_1, c_2) \in S$ is assigned to timeslots $t_1 \in T_{c_1}$, $t_2 \in T_{c_2}$, respectively, 0 otherwise.
- y_{ct} : 1 if course $c \in C$ is assigned to timeslot $t \in T_c$, 0 otherwise.

With these parameters and decision variables, the model can be formulated as below:

(P)

$$
\begin{array}{llll}
\min & \sum_{(c_1,c_2)\in S, t_1\in T_{c_1} t_2\in T_{c_2}} c_{c_1c_2t_1t_2} x_{c_1c_2t_1t_2} & & \\
\text{s.t.} & & & \\
\sum_{t\in T_c} y_{ct} & = 1 & \forall c \in C & (3.1)\\
\sum_{t_2\in T_{c_2}} x_{c_1c_2t_1t_2} & = y_{c_1t_1} & \forall (c_1,c_2)\in S, t_1\in T_{c_1} & (3.2)\\
\sum_{t_1\in T_{c_1}} x_{c_1c_2t_1t_2} & = y_{c_2t_2} & \forall (c_1,c_2)\in S, t_2\in T_{c_2} & (3.3)\\
x_{c_1c_2tt} & = 0 & \forall (c_1,c_2)\in S, t\in T_{c_1}\cap T_{c_2} & (3.4)\\
x_{c_1c_2t_1t_2} & \leq 1 & \forall (c_1,c_2)\in S, t_1\in T_{c_1}, t_2\in T_{c_2} & (3.5)\\
x_{c_1c_2t_1t_2} & \geq 0 & \forall (c_1,c_2)\in S, t_1\in T_{c_1}, t_2\in T_{c_2} & (3.6)\\
y_{ct} & \in \{0,1\} & \forall c\in C, t\in T_c & (3.7)
\end{array}
$$

Constraint (3.1) guarantees that each exam is assigned to exactly one timeslot. Constraints (3.2) and (3.3) relate $x_{c_1c_2t_1t_2}$ variables to y_{ct} variables for timeslots t_1 and t_2, respectively. Constraint (3.4) blocks the conflicting exams. Other constraints are bounding constraints. Although $x_{c_1c_2t_1t_2}$ variables are binary in the formulation, even if they are relaxed, the binary nature of y_{ct} variables will force them to be binary. Therefore, one can apply Bender's decomposition by sending continuous $x_{c_1c_2t_1t_2}$ variables to the subproblem and keeping binary y_{ct} variables in the master problem.

To apply Bender's decomposition, the compact model is decomposed in such a way that the master problem consists of only integer and binary variables. Continuous variables, the constraints consisting of continuous variables, and the part of the objective function consisting of continuous variables are sent to the subproblem. The algorithm starts with a solution for the master problem. Then, according to the solution gained by the master problem, the dual of the subproblem is solved. If an optimal solution can be found for the dual of the subproblem, an optimality cut guaranteeing optimality is added to the master problem as a constraint. If the dual of the subproblem is unbounded, which means that the subproblem is infeasible, a feasibility cut is added to the master problem using the extreme ray, which is the direction causing the dual of the subproblem being unbounded. Then, the master problem is solved again with additional constraints. At each iteration, new feasibility or optimality cuts are added to the master problem. When there is no cut to add, the algorithm stops with optimality. A formal algorithm, namely Algorithm 1, is provided after stating the master problem, the subproblem, and the dual of the subproblem. For the master problem, a variable α is used to calculate the cost. The master problem is formulated as:

(MP)

$$\begin{array}{llll}
\min & \sum_{(c_1,c_2)\in S} \alpha_{c_1c_2} & & \\
\text{s.t.} & & & \\
\sum_{t\in Tc} y_{ct} & = \ 1 & \forall c \in C & (3.8)\\
y_{c_1t} + y_{c_2t} & \leq \ 1 & \forall (c_1,c_2) \in S, t \in T & (3.9)\\
y_{ct} & \in \ \{0,1\} & \forall c \in C, t \in T_c & (3.10)
\end{array}$$

The subproblem is modeled below:

(SP)

$$\begin{array}{lllll}
\min & \sum_{(c_1c_2)\in S, t_1t_2\in T} c_{c_1c_2t_1t_2} x_{c_1c_2t_1t_2} & & & \\
\text{s.t.} & & & & \\
\sum_{t_2\in T_{c_2}} x_{c_1c_2t_1t_2} & = \ \hat{y}_{c_1t_1} & \forall (c_1,c_2) \in S, t_1 \in T_{c_1} & (3.11) & (\mu_{c_1c_2t_1})\\
\sum_{t_1\in T_{c_1}} x_{c_1c_2t_1t_2} & = \ \hat{y}_{c_2t_2} & \forall (c_1,c_2) \in S, t_2 \in T_{c_2} & (3.12) & (v_{c_1c_2t_2})\\
x_{c_1c_2t_1t_2} & \leq \ 1 & \forall (c_1,c_2) \in S, t_1 \in T_{c_1}, t_2 \in T_{c_2} & (3.13) & \\
x_{c_1c_2t_1t_2} & \geq \ 0 & \forall (c_1,c_2) \in S, t_1 \in T_{c_1}, t_2 \in T_{c_2} & (3.14) &
\end{array}$$

where $\hat{y}$ corresponds to a solution of the master problem MP. It can be observed that the subproblem is always feasible for any feasible solution of the master problem, hence, there is no need to add feasibility cuts to the master problem. Moreover, it is not needed to add constraint (3.13) since $x_{c_1c_2t_1t_2}$ variables cannot be greater than 1 due to constraints (3.11) and (3.12).The dual variables are $\mu_{c_1c_2t_1}$ for constraint (3.11) and $v_{c_1c_2t_2}$ for constraint (3.12). Hence, the dual of the subproblem becomes:

(DSP)

$$
\begin{aligned}
&\max && \sum_{(c_1,c_2)\in S, t_1\in T_{c_1}, t_2\in T_{c_2}} \hat{y}_{c_1t_1}\mu_{c_1c_2t_1} + \hat{y}_{c_2t_2}v_{c_1c_2t_2} \\
&\text{s.t.} && \\
&\mu_{c_1c_2t_1} + v_{c_1c_2t_2} \leq c_{c_1c_2t_1t_2} && \forall (c_1,c_2)\in S, t_1\in T_{c_1}, t_2\in T_{c_2} \quad (3.15)
\end{aligned}
$$

Since there are large number of variables in DSP, Mokhtar et al. [23] solved subproblems for each course pairs $(c_1, c_2) \in S$. Therefore, for each $(c_1, c_2) \in S$, $DSP_{c_1c_2}$ is formulated as below:

$(DSP_{c_1c_2})$

$$
\begin{aligned}
&\max && \sum_{t_1\in T_{c_1}, t_2\in T_{c_2}} \hat{y}_{c_1t_1}\mu_{c_1c_2t_1} + \hat{y}_{c_2t_2}v_{c_1c_2t_2} \\
&\text{s.t.} && \\
&\mu_{c_1c_2t_1} + v_{c_1c_2t_2} \leq c_{c_1c_2t_1t_2} && \forall t_1\in T_{c_1}, t_2\in T_{c_2} \qquad (3.16)
\end{aligned}
$$

A solution for $DSP_{c_1c_2}$ for each course pair $(c_1, c_2) \in S$ is found in each iteration of Bender's algorithm. Using $DSP_{c_1c_2}$ makes solving the subproblem easier since DSP has so many variables and can be challenging if the number of courses is high. With $DSP_{c_1c_2}$, the model is separated into smaller problems. In addition, it allows us to add more effective optimality cuts to the master problem since in this case, a total of $|S|$ small cuts are added instead of one large cut.

At each iteration of the Bender's decomposition algorithm, an optimality cut of form

$$
\alpha_{c_1c_2} \geq \sum_{t_1t_2\in T} y_{c_1t_1}\hat{\mu}_{c_1c_2t_1} + y_{c_2t_2}\hat{v}_{c_1c_2t_2} \quad \forall (c_1,c_2)\in S \quad (3.17)
$$

is added to the problem. Constraint (3.17) is not directly added to MP since it increases the size of the model extremely, however, it is added when it is violated.

For each solution of the MP, there are many alternate optimal solutions for the subproblem. To add more beneficial cuts, an optimal solution that will lead to an effective cut, out of many optimal solutions should be found. To use a good optimal solution to add cuts, Mokhtar et al. [23] fixed assignment variables $\hat{y}_{ct}$ to $1.6/|C|$ as the objective coefficient of $\mu_{c_1c_2t_1}$ and to $1.2/|C|$ as the objective coefficient of $v_{c_1c_2t_2}$ variables in the dual of the subproblem, whenever they have 0 values in the solution of the master problem. This approach is used in Bender's decomposition applications in this chapter.

As a different application, constraint (3.9) might be removed from MP and added to the subproblem. In this case, the property of having feasible solutions in the subproblem is lost. Therefore, in addition to the optimality cuts, feasibility cuts must be added to the master problem at each iteration. Adding a new constraint to SP affects $DSP_{c_1c_2}$. Therefore, constraint (3.16) in $DSP_{c_1c_2}$ is separated and written as follows:

$$\begin{aligned} \mu_{c_1c_2t_1} + v_{c_1c_2t_2} &\leq c_{c_1c_2t_1t_2} \quad \forall t_1 \in T_{c_1}, t_2 \in T_{c_2}, t_1 \neq t_2 && (3.18)\\ \mu_{c_1c_2t} + v_{c_1c_2t} + t_{c_1c_2t} &\leq c_{c_1c_2tt} \quad \forall t \in T && (3.19)\end{aligned}$$

In addition, for each unbounded $DSP_{c_1c_2}$, a feasibility cut of form

$$0 \geq \sum_{t_1\in T_{c_1}, t_2\in T_{c_2}} y_{c_1t_1}\hat{\mu}^r_{c_1c_2t_1} + y_{c_2t_2}\hat{v}^r_{c_1c_2t_2} \quad \forall (c_1,c_2) \in S \quad (3.20)$$

should be added to the master problem where $\hat{\mu}^r_{c_1c_2t_1}$ and $\hat{v}^r_{c_1c_2t_2}$ are values of the extreme rays, which are the directions of the problem causing the unboundedness of the subproblem. A formal Bender's decomposition algorithm is stated in Algorithm 1.

3.3 Numerical Results

There are a few options while applying Bender's algorithm. Firstly, adding/not adding feasibility cuts is a decision. If DSP is formulated in such a way that all solutions of the dual problem are feasible to the master problem, there is no need to add the feasibility cuts. The solution with feasibility cuts is also tested since it might decrease the dimension of the master problem.

Algorithm 1 Bender's Decomposition Algorithm

```
1: procedure BENDER'S
2:     Solve MP without constraint (3.17) to start with a feasible solution
3:     for (c1, c2) ∈ S do
4:         Solve DSP_{c1c2} using the solution of MP
5:     end for
6:     if DSP_{c1c2} is solved with optimality then
7:         Add optimality cut, constraint of type (3.17), to MP
8:     else
9:         if DSP_{c1c2} is unbounded for any (c1, c2) ∈ S then
10:            Find the extreme ray making DSP_{c1c2} unbounded and add a feasibility cut, a constraint of type (3.20), to MP
11:        end if
12:    end if
13:    if DSP_{c1c2} is solved with optimality for all (c1, c2) ∈ S and all optimality cuts are satisfied by the solution of MP then
14:        Stop with optimality
15:    else
16:        Solve MP, go to step 4.
17:    end if
18: end procedure
```

Figure 3.1: Bender's Decomposition Algorithm

Secondly, while calculating the cost in the master problem, a single α variable might be used as it is proposed above. Another option might be using $\alpha_{c_1c_2}$ variables that are describing the cost of each course pair in set S. It was observed that using only a single α variable does not give successful results since in that case, optimality cuts used for the total cost created by all of the course pairs include many variables and do not change the feasible region enough. Hence, we decided not to use a single α variable. Instead, we used $\alpha_{c_1c_2}$ variables for each course pairs.

Using $\alpha_{c_1c_2}$ variables gives good results without feasible cuts, however, solving MP becomes much harder after a small number of iterations. To solve this issue, firstly, each subproblem is solved and whenever a subproblem is unbounded, a feasibility cut is added, and whenever a subproblem is solved with optimality, optimality cut is not added to the problem, in contrast with the classical Bender's decomposition. We did not add optimality cuts until we reach the iteration that creates no unbounded subproblems, so no feasibility cuts. Therefore, in the first few iterations of Bender's algorithm, only feasibility cuts were added to MP. When there is no feasibility cut to add, optimality cuts are started to be added. This method increases the duration of the algorithm, however, solving the problem is expected to become easier.

Lastly, $\alpha_{c_1c_2}$ variables are stated as continuous and integer variables in different formulations to try another diversification. Finally, it was found that the continuous version is slightly better.

The compact formulation was also tried to solve the problems in the literature. The compact version with shrinking students taking the same exams is called P. With and without feasibility cut version of Bender's Decomposition is called $BDFC$ and $BDWFC$.

These algorithms were tested using $Bilkent100$ data by starting with a good feasible solution, adding a constraint to guarantee that the assignment of exams that have exam numbers larger than 100 to be assigned to the same timeslot as the starting solution. Since most exact solution methods do not allow us to use the hardness of courses, hardness property of exams of Bilkent University problem is omitted for the exact solution methods in this chapter. The first 100 exams can be assigned to any timeslot if there is no conflict between others.

All of the methods suggested above were able to solve the problem with 100 exams. However, when the formulations are tested on whole $Bilkent$ problem, good cuts could not be added by the solver. Additionally, to test the suggested algorithms with whole instances, $yue20023$, $yue20011$ and $UTE92$ are used. $yue20023$ and $Bilkent100$ problems could be optimally solved using P, $BDWFC$

and $BDFC$. As an illustration, optimal exam-timeslot assignments for $yue20023$ can be found in Figure 3.2. Figure 3.3 demonstrates the optimal schedule for a selected student with student number 16. In Figure 3.2, circles and squares represent exams and timeslots, respectively. In Figure 3.3, squares represent timeslots and values inside them are the exam numbers of exams that are assigned to timeslots.

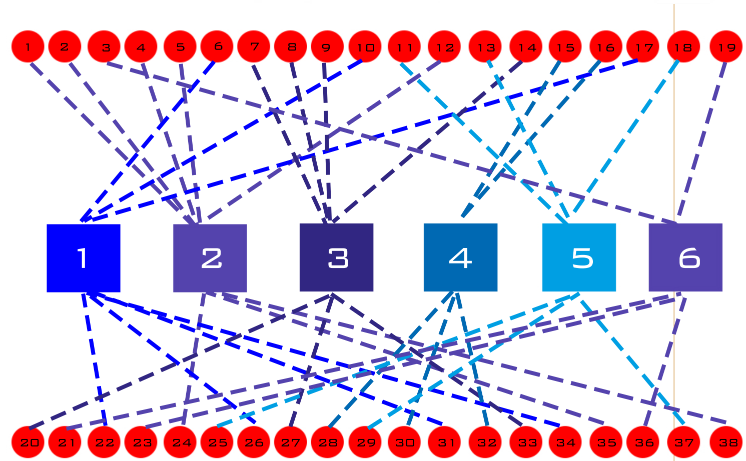

Figure 3.2: Optimal assignments for $yue20023$

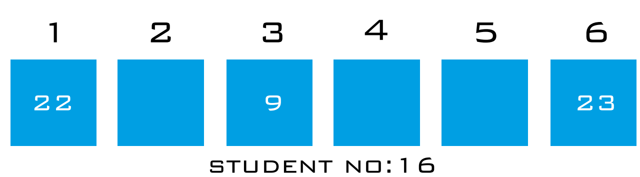

Figure 3.3: Optimal schedule for student 16 in $yue20023$

These three algorithms were also tested on $yue20011$ and $UTE92$ problems and these two relatively small problems could not be solved using any of these algorithms. After some iterations, master problems were not solved or solved to optimality of zero objective value for a large number of iterations. Tables 3.2 and 3.3 show the best-found integer objective values and best lower bounds, respectively when problems are tested with a time limit of 3 hours using CPLEX 12.7 on Java on a UNIX platform provided by Bilkent University. All computations are performed on 3.6 GHz Intel Core i7-3820 CPU with 64 GB memory. Optimally solved problems are stated with a "*". When no bounds or integer solutions can be found, the corresponding cell is stated as "-".

Problem	P	BDFC	BDWFC
Bilkent100	42212*	42212*	42212*
UTE92	85827	-	-
yue20011	-	-	-
yue20023	56*	56*	56*

Table 3.2: Best found objective values for four-indexed formulation

Problem	P	BDFC	BDWFC
Bilkent100	42212*	42212*	42212*
UTE92	1131	-	-
yue20011	-	-	-
yue20023	56*	56*	56*

Table 3.3: Best lower bounds for four-indexed formulation

It can be seen from the results that P gives some lower bounds and Bender's decomposition formulations do not give a positive lower bound. Bender's decomposition for P does not reach successful solutions for larger problems. However, these approaches could solve small problems in 3 hours. However, since solving the dual of the subproblem and getting the solution at each iteration for Bender's decomposition take time, P was able to solve the small instances in shorter times. Out of the unsolved problems, since $UTE92$ problem is more likely to be solved according to the results, tests in this thesis will be mostly applied to this instance in later parts.

Chapter 4

Flow-Based Formulations

This chapter proposes two flow formulations that were used for the p-hub median hub location problem adopted to solve the examination timetabling problem. The first formulation includes a three-indexed variable to represent the cost of the traffic between course $c \in C$, which is assigned to timeslot $t_1 \in T_c$, and any other course that is assigned to timeslot $t_2 \in T$. The formulation is presented and some changes in the constraints are made. Bender's decomposition algorithm is applied to the formulation. In the subproblem, some changes are made to cope with degeneracy. A different flow formulation is proposed with two-indexed variables to calculate the cost incurred between two timeslots. These approaches are tested on some instances in the literature.

4.1 Three-Indexed Flow Formulation

In the p-hub median hub location problem, given a specific number of nodes, some nodes are selected as hubs and the remaining nodes are assigned to the selected hubs. The aim of the problem is to minimize the total cost created by the transfer between any two nodes. The cost in the problem can be approached similarly with the cost created by the common students in each course pair in the examination timetabling problem.

Ernst and Krishnamoorthy [19] proposed a three-indexed flow formulation for the p-hub median hub location problem. In their work, $x_{ct_1t_2}$ variables are used to calculate the cost created by the traffic from node $c \in C$ to any other node between hubs $t_1, t_2 \in T$. This cost can be considered as the total number of students taking both course $c \in C$, which is assigned to timeslot $t_1 \in T_c$ and any other course assigned to timeslot $t_2 \in T$. Figure 4.1 demonstrates the flow variable $x_{ct_1t_2}$.

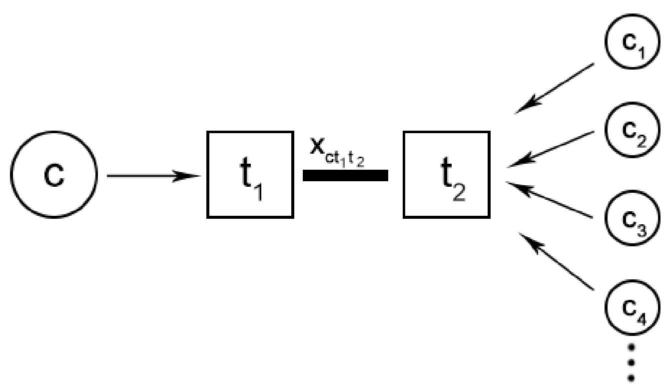

Figure 4.1: Three-indexed flow formulation graph

With the same logic, the examination timetabling problem might also be modeled through three-indexed flow variables. Additional parameters and the decision variables required for the mathematical model are listed as:

Parameters

- m_{c_1} : the total number of course intersections of students taking course $c_1 \in C$, ie. $m_{c_1} = \sum_{c_2 \in C | c_1 \neq c_2} a_{c_1c_2}$.
- $d_{t_1t_2}$: the cost of having exams at timeslot $t_1, t_2 \in T$.

Decision Variables

- $x_{c\,t_1t_2}$: total flow of course $c \in C$ created by the assignment of other courses assigned to timeslot $t_2 \in T$ when course $c \in C$ is assigned to timeslot $t_1 \in T_c$.

The three-indexed model for the examination timetabling problem is formulated below:

(FP)

$$
\begin{array}{llll r}
\min & \sum_{t_1 \in T_c, t_2 \in T, t_1 \neq t_2, c \in C} d_{t_1t_2} x_{c\,t_1t_2} & & & \\
\text{s.t.} & & & & \\
\sum_{t_2 \in T} x_{c\,t_1t_2} & = & m_c y_{ct_1} & \forall c \in C, t_1 \in T_c & (4.1) \\
\sum_{t_1 \in T_{c_1}} x_{c_1t_1t_2} & = & \sum_{c_2 \in C} a_{c_1c_2} y_{c_2t_2} & \forall c_1 \in C, t_2 \in T_{c_2} & (4.2) \\
\sum_{t \in Tc} y_{ct} & = & 1 & \forall c \in C & (4.3) \\
y_{c_1t} + y_{c_2t} & \leq & 1 & \forall (c_1, c_2) \in S, t \in T & (4.4) \\
x_{c\,t_1t_2} & \leq & m_c & \forall c \in C, t_1 \in T_c, t_2 \in T & (4.5) \\
x_{c\,t_1t_2} & \geq & 0 & \forall c \in C, t_1 \in T_c, t_2 \in T & (4.6) \\
y_{ct} & \in & \{0, 1\} & \forall c \in C, t \in T_c & (4.7)
\end{array}
$$

The compact model FP can be used to solve the examination timetabling problem. Constraint (4.1) guarantees that the total flow of a course is equal to its total intersection. Constraint (4.2) satisfies the same equality for other courses by making the total flow equal to the number of intersections between course c_1 to any other courses. Constraint (4.3) guarantees that each exam is assigned to exactly one timeslot. Constraint (4.4) blocks the conflicts. Constraint (4.5) is a valid inequality used to bound x variables. Other constraints are nonnegativity and binary constraints.

As another approach, since only a proportion of timeslot pairs create cost ($d_{t_1t_2}$

is greater than 0 for these timeslot pairs and 0 for others), we might eliminate $x_{c\,t_1t_2}$ variables for the timeslot pairs that do not create cost in order to decrease the problem size. Let $L_{t_1t_2}$ be 1 if timeslots (t_1, t_2) create cost and 0 otherwise. To eliminate $x_{c\,t_1t_2}$ variables, constraint (4.2) should be changed to satisfy the flow requirements since the flow between timeslot pairs that do not create cost is not included in the model anymore. Therefore, constraint (4.2) becomes:

$$\sum_{t_1 \in T_{c_1} | L_{t_1t_2}=1} x_{c_1t_1t_2} \;\geq\; \sum_{c_2 \in C} a_{c_1c_2} y_{c_2t_2} - \sum_{t_1 \in T_{c_1} | L_{t_1t_2}=0} y_{c_1t_1} m_{c_1} \quad \forall c_1 \in C, t_2 \in T_{c_2} \quad (4.2.1)$$

Moreover, constraint (4.1) must also be changed due to the variable elimination in the model. The sum of $x_{c\,t_1t_2}$ values is now less than or equal to m_c values, instead of the equality that we used in (FP). Therefore, constraint (4.1) must be removed and two constraints below must be added to the model.

$$\sum_{t_2 \in T} x_{c\,t_1t_2} \;\geq\; m_c\,(y_{ct_1} - 1) \quad \forall c \in C, t_1 \in T_c \mid L_{t_1t_2} = 1 \quad (4.1.1)$$
$$\sum_{t_2 \in T} x_{c\,t_1t_2} \;\leq\; m_c\, y_{ct_1} \quad \forall c \in C, t_1 \in T_c \mid L_{t_1t_2} = 1 \quad (4.1.2)$$

If course $c \in C$ is not assigned to timeslot $t_1 \in T_c$, by constraint (4.1.2), $x_{c\,t_1t_2}$ will be 0 for all $t_2 \in T$. To explain constraint (4.1.1), let us define a new variable, $x_{c\,t_10}$, which describes the total flow created by course $c \in C$ and goes from t_1 $\in T_c$ through any timeslot $t_2 \in T$ that does not create cost between timeslot t_1 $(L_{t_1t_2} = 0)$. Clearly, $x_{c\,t_10}$ variables are less than or equal to m_c forall $c \in C$. Hence, the logic behind constraint (4.1) is that the constraint implies that the sum of all flow is equal to the total flow created by course $c \in C$, ie.

$$\sum_{t_2 \in T | L_{t_1t_2}=1} x_{c\,t_1t_2} + x_{c\,t_10} \;=\; m_c y_{ct_1} \quad \forall c \in C, t_1 \in T_c \qquad (4.1.3)$$

Let us make $x_{c\,t_10}$ variables alone in that equation. Therefore, constraints (4.1.1) and (4.1.2) are valid for the model. Obviously, there is no need for constraint (4.1.1) since the other constraints imply it for all x and y variables. Therefore, the model becomes:

$(FP1)$

$$
\begin{array}{llll}
\min & \sum_{c\in C, t_1\in T_c, t_2\in T, t_1\neq t_2} d_{t_1t_2} x_{c\,t_1t_2} & & \\
\text{s.t.} & & & \\
\sum_{t_2\in T|L_{t_1t_2=1}} x_{c\,t_1t_2} + x_{c\,t_1 0} & = m_c y_{ct_1} & \forall c\in C, t_1\in T_c & (4.8)\\
\sum_{t_1\in T_{c_1}|L_{t_1t_2}=1} x_{c_1t_1t_2} & \geq \sum_{c_2\in C} a_{c_1c_2} y_{c_2t_2} - \sum_{t_1|L_{t_1t_2}=0} y_{c_1t_1} m_{c_1} & \forall c_1\in C, t_2\in T_{c_2} & (4.9)\\
\sum_{t\in T_c} y_{ct} & = 1 & \forall c\in C & (4.10)\\
x_{c\,t_1t_2} & \leq m_c & \forall c\in C, t_1\in T_c, t_2\in T & (4.11)\\
y_{c_1t} + y_{c_2t} & \leq 1 & \forall (c_1,c_2)\in S, t\in T & (4.12)\\
x_{c\,t_1t_2} & \geq 0 & \forall c\in C, t_1\in T_c, t_2\in T & (4.13)\\
y_{ct} & \in \{0,1\} & \forall c\in C, t\in T_c & (4.14)
\end{array}
$$

In addition, instead of constraint (4.8), constraint (4.1.2) might be used. The version with constraint (4.8) is called $(FP1)$ and with constraint (4.1.2) is called $(FP2)$. The next section presents Bender's decomposition algorithm applied to these formulations.

4.2 Bender's Decomposition for (FP1) and (FP2)

In the Bender's decomposition algorithm, one can see that the dual of subproblems and master problems for these two formulations are exactly the same. Although these formulations may be different in computational time when the compact form is used, Bender's decomposition can only be applied in one way. To apply Bender's decomposition to flow formulations, y variables are put in the master problem and all other variables are sent to the subproblem. Let (MP), (SP), (DSP) describe the master problem, the subproblem and the dual of the subproblem for $(FP1)$ and $(FP2)$. The master problem is formulated as:

(MP)

$$\begin{array}{llll}
\min & \alpha & & \\
\text{s.t.} & & & \\
\sum_{t\in T_c} y_{ct} & = 1 & \forall c\in C & (4.15)\\
y_{c_1t}+y_{c_2t} & \leq 1 & \forall (c_1,c_2)\in S, t\in T & (4.16)\\
y_{ct} & \in \{0,1\} & \forall c\in C, t\in T_c & (4.17)
\end{array}$$

The subproblem is formulated as:
(SP)

$$\begin{array}{lllll}
\min & \sum_{t_1\in T_c,t_2\in T,t_1\neq t_2,c\in C} d_{t_1t_2}x_{c\,t_1t_2} & & & \\
\text{s.t.} & & & & \\
\sum_{t_2\in T|L_{t_1t_2}=1} x_{ct_1t_2}+x_{ct_10} & = m_c\hat{y}_{ct_1} & \forall c\in C, t_1\in T_c & (4.18) & (\mu_{ct_1})\\
\sum_{t_1\in T_{c_1}|L_{t_1t_2}=1} x_{c_1t_1t_2} & \geq \sum_{c_2\in C} a_{c_1c_2}\hat{y}_{c_2t_2} - \sum_{t_1|L_{t_1t_2}=0}\hat{y}_{c_1t_1}m_{c_1} & \forall c_1\in C, t_2\in T_{c_2} & (4.19) & (v_{c_1t_2})\\
x_{ct_1t_2} & \geq 0 & \forall c\in C, t_1\in T_c, t_2\in T & (4.20) & \\
x_{ct_1t_2} & \leq 1 & \forall c\in C, t_1\in T_c, t_2\in T & (4.21) &
\end{array}$$

where $\hat{y}_{ct}$ is the value of y_{ct} in the solution of the master problem. The dual variables corresponding to constraints (4.18) and (4.19) are μ_{ct_1} and $v_{c_1t_2}$, respectively. The dual of the subproblem is:
(DSP)

$$\begin{array}{llll}
\max & \sum_{c_1\in C,t_1\in T_{c_1}} m_{c_1}\,\hat{y}_{c_1t_1}\mu_{c_1t_1}+\left(\sum_{c_2\in C:(c_1,c_2)\in S} a_{c_1c_2}\hat{y}_{c_2t_1}-\sum_{t_2\in T:T_{t_1t_2}=0}\hat{y}_{c_1t_2}m_{c_1}\right)v_{c_1t_1} & & \\
\text{s.t.} & & & \\
\mu_{ct_1}+v_{ct_2} & \leq d_{t_1t_2} & \forall c\in C, t_1\in T_{c_1}, t_2\in T \mid L_{t_1t_2}=1 & (4.22)\\
\mu_{ct} & \leq 0 & \forall c\in C, t\in T_c & (4.23)\\
v_{ct} & \geq 0 & \forall c\in C, t\in T_c & (4.24)
\end{array}$$

Like the previous chapter, there is no need to add any feasibility cuts to the master problem since the master problem is always feasible for all solutions of the dual problem. This is because the master problem has all of the constraints in the compact model. Bender's decomposition algorithm is applied by solving (MP) and (DSP) iteratively, and optimality cuts below are added in each iteration:

$$\alpha \geq \sum_{c_1 \in C, t_1 \in T_{c_1}} m_{c_1} y_{c_1 t_1} \hat{\mu}_{c_1 t_1} + \left(\sum_{c_2 \in C:(c_1,c_2) \in S} a_{c_1 c_2} y_{c_2 t_1} - \sum_{t_2 | T_{t_1 t_2} = 0} y_{c_1 t_2} m_{c_1} \right) \hat{v}_{c_1 t_1} \quad (4.25)$$

4.3 Two-Indexed Flow Formulation

A different formulation can be derived with variables corresponding to the flow between timeslots $t_1, t_2 \in T$. For the p-hub median hub location problem, Labbe et al. [22] proposed a two-indexed flow formulation. In that formulation, $x_{t_1 t_2}$ variables are used to calculate the total flow between hubs $t_1, t_2 \in T$. The same variable can be used to describe the cost of of students having two exams assigned to timeslots $t_1, t_2 \in T$, in the examination timetabling problem. Figure 4.2 demonstrates the flow variable $x_{t_1 t_2}$.

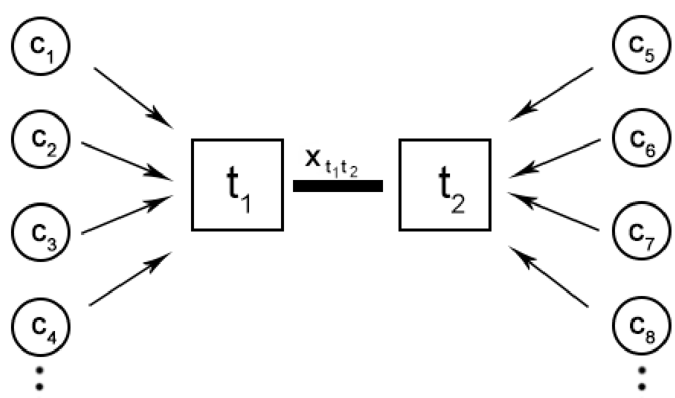

Figure 4.2: Two-indexed flow formulation graph

In the paper, a branch and cut approach is applied to the problem and a constraint ultimately making the problem dimension too large to solve is added at each iteration when it is violated. This formulation can be transformed to be used in the examination timetabling problem and the same branch and cut method can be benefited from. In the transformed formulation, $x_{t_1 t_2}$ variables

are used to calculate the total cost between timeslots $t_1, t_2 \in T$. Therefore, $x_{t_1t_2}$ variables must be equal to the cost created by all course pairs $(c_1, c_2) \in S$ which are assigned to timeslots t_1 and $t_2 \in T$, respectively. In the examination timetabling problem, subsets of course pairs $(c_1, c_2) \in S$ are similar to the subsets of nodes in the p-hub median hub location problem, and the number of students taking both courses $(c_1, c_2) \in S$ are similar to the transfer between these node pairs. In other words, $x_{t_1t_2}$ variables (total flow between timeslots t_1 and t_2) are larger than the total flow among all course pairs where one course is assigned to timeslot $t_1 \in T$ and the other is assigned to timeslot $t_2 \in T$ and this inequality is valid for all subsets of such courses. This is guaranteed in constraint (4.26) in the model below. The compact formulation of the model is:
(LYG)

$$\min \sum_{t_1t_2 \in T, t_1 \neq t_2} d_{t_1t_2} x_{t_1t_2}$$

s.t.

$$x_{t_1t_2} \geq \sum_{(c_1,c_2)\in S'} a_{c_1c_2}(y_{c_1t_1} + y_{c_2t_2} - 1) \quad \forall t_1 \in T_{c_1}, t_2 \in T_{c_2}, S' \subset S \tag{4.26}$$

$$\sum_{t \in T_c} y_{ct} = 1 \quad \forall c \in C \tag{4.27}$$

$$y_{c_1t} + y_{c_2t} \leq 1 \quad \forall (c_1, c_2) \in S, t \in T \tag{4.28}$$

$$x_{t_1t_2} \geq 0 \quad \forall t_1t_2 \in T \tag{4.29}$$

$$y_{ct} \in \{0, 1\} \quad \forall c \in C, t \in T_c \tag{4.30}$$

Since constraint (4.26) is for all subsets of S, adding all of the constraints (4.26) makes the size of the problem too large to solve. Therefore, the problem can be solved by iteratively adding them when it is needed. At each iteration, a separation problem is solved and a constraint of type (4.26) is added to the model for only one $S' \subset S$ for each $t_1, t_2 \in T$ where S' consists of all course pairs $(c_1, c_2) \in S$ assigned to timeslots $t_1, t_2 \in T$, respectively. The separation problem is solved using the current solution of the model and a constraint of type (4.26) for each $t_1, t_2 \in T$ is constructed using all course pairs assigned to timeslots t_1, t_2. Constraints for other subsets are not added for an iteration, and added in future iterations if they are violated in the solution procedure. Hence, the optimal objective value is 0 firstly, and it increases at each iteration. When no inequality of type (4.26) is violated, the algorithm stops.

4.4 Numerical Results

$(FP1)$, $(FP2)$, their Bender's decomposition algorithm $(BDFP)$, and LYG were used to solve $yue20023$, $Bilkent100$, and $UTE92$. For $Bilkent100$, the hardness property of exams is omitted as before. All four approaches were able to solve $yue20023$ and $Bilkent100$ in less than one hour. It was observed that $FP1$ and $FP2$ are better than other two formulations since they are compact formulations. However, $UTE92$ problem could not be solved exactly. Another approach, adding both flow Bender's optimality cuts and four-indexed Bender's optimality cuts stated in Chapter 3 were also tested on this instance, however, this approach was also inefficient. Tables 4.1 and 4.2 show the best-found integer objective values and best lower bounds when problems are tried to be solved with a time limit of 3 hours using CPLEX 12.7 on Java on a UNIX platform provided by Bilkent University.

Problem	FP1	FP2	BDFP	LYG
Bilkent100	42212*	42212*	42212*	42212*
UTE92	81686	83206	-	-
yue20023	56*	56*	56*	56*

Table 4.1: Best found integer objective values for flow formulations

Problem	FP1	FP2	BDFP	LYG
Bilkent100	42212*	42212*	42212*	42212*
UTE92	900	844	-	-
yue20023	56*	56*	56*	56*

Table 4.2: Best found lower bounds for flow formulations

LYG and $BDFP$ were not effective in solving $UTE92$ problem. In addition, it can be seen that $FP1$ gives better results than $FP2$ in terms of both best-found lower bounds and best-found integer objective values.

Chapter 5

Kauffmann-Broeckx Formulation and Its Applications

This chapter gives the details of the quadratic assignment problem and proposes a lower bound calculation method. An efficient formulation for the quadratic assignment problem is stated and transformed to solve the examination timetabling problem. Bender's decomposition is applied to the proposed formulation and results for the instances in the literature are stated.

5.1 Gilmore-Lawler Lower Bound

In the quadratic assignment problem, a given number of plants are assigned to the same number of locations such that the traffic between plants is minimized. Since the objective function is quadratic and depends on the assignment of plants, the problem is similar to the examination timetabling problem. The only difference between the two problems is that in the quadratic assignment problem, the assignments are one-to-one, whereas, in the examination timetabling problem, many courses can be scheduled to the same timeslot.

The quadratic assignment problem is similar to the linear assignment problem which includes the assignment of plants to locations with a linear objective function. In the linear assignment problem, the cost is created by each assignment. However, the cost in the quadratic assignment problem is created by the transfer between plants through their location that they assigned.

Kauffmann and Broeckx [27] created a linear formulation for the quadratic assignment problem using x_{ct} variables to calculate the cost created by the assignment of facility $c \in C$ to location $t \in T$. This approach might also be useful for the examination timetabling problem. x_{ct} variables in the formulation can be considered as the cost of assigning the exam of course $c \in C$ to timeslot $t \in T$. In addition to the same parameters and decision variables used previously, x_{ct} variables are used to calculate the cost created by assignment of exam $c \in C$ to timeslot $t \in T$. x_{ct} variable can be nonlinearly described as:

$$x_{c_1t_1} = y_{c_1t_1} \sum_{c_2 \in C, t_2 \in T} c_{c_1c_2t_1t_2} y_{c_2t_2} \quad \forall c_1 \in C, t_1 \in T_{c_1}$$

The proposed linear formulation is:
(KBL)

$$\begin{array}{llll}
\min & \sum_{c \in C, t \in T_c} x_{ct} & & \\
\text{s.t.} & & & \\
\sum_{t \in Tc} y_{ct} & = 1 & \forall c \in C & (5.1) \\
x_{c_1t_1} & \geq \sum_{c_2 \in C, t_2 \in T_{c_2}} c_{c_1c_2t_1t_2} y_{c_2t_2} - t_{c_1t_1}(1 - y_{c_1t_1}) & \forall c_1 \in C, t_1 \in T_{c_1} & (5.2) \\
y_{c_1t} + y_{c_2t} & \leq 1 & \forall (c_1, c_2) \in S, t \in T & (5.3) \\
x_{ct} & \geq 0 & \forall c \in C, t \in T_c & (5.4) \\
y_{ct} & \in \{0, 1\} & \forall c \in C, t \in T_c & (5.5)
\end{array}$$

where

$$t_{c_1t_1} = \sum_{c_2 \in C, t_2 \in T_{c_2}} c_{c_1c_2t_1t_2} \quad \forall c_1 \in C, t_1 \in T_{c_1}$$

is used as a big-M variable for the linearization. Constraint (5.1) makes each course assigned to exactly one timeslot. Constraint (5.2) calculates the cost of an assignment. Constraint (5.3) blocks the conflicts. Other constraints are nonnegativity and binary constraints.

A useful bound for the quadratic assignment problem is Gilmore-Lawler lower bound. Gilmore-Lawler lower bound was introduced by Gilmore [25] and Lawler [26] by solving linear assignment problems for each plant-location assignment. The bound is derived by calculating the total minimum cost created by assigning any course to any timeslot. More precisely, Gilmore-Lawler bound might be found by solving the linear assignment problem for each $c_1 \in C$ and $t_1 \in T_{c_1}$ below:

$(GLB_{c_1t_1})$

$$\begin{array}{llll}
\min & \displaystyle\sum_{c_2 \in C:(c_1,c_2)\in S, t_2 \in T_{c_2}} c_{c_1c_2t_1t_2} y_{c_2t_2} & & \\
\text{s.t.} & & & \\
\displaystyle\sum_{t_2 \in Tc_2} y_{c_2t_2} & = \;\; 1 & \forall c_2 \in C & (5.6) \\
y_{c_2t_2} + y_{c_3t_2} & \leq \;\; 1 & \forall (c_2, c_3) \in S, t_2 \in T & (5.7) \\
y_{c_2t_2} & \in \;\; \{0,1\} & \forall c_2 \in C, t_2 \in T_{c_2} & (5.8)
\end{array}$$

Let I_{ct} be the optimal value of (GLB_{ct}) for an arbitrary $c \in C$ and $t \in T$ assignment. Then, I_{ct} is a lower bound for the x_{ct} variable in (KBL). There are many formulations proposed in the quadratic assignment literature based on Gilmore-Lawler bounds. The next section states a formulation that uses Gilmore-Lawler bounds and it gives good results for the instances in the quadratic assignment problem literature. This formulation is based on (KBL) and was stated by Xia and Yuan [28]. In this formulation, lower bounds are derived from (GLB) for each x_{ct} variable. A big-M value, which is the maximization version of (GLB) is used to derive an upper bound for x_{ct} variables.

5.2 A GLB Based Linearization on KBL

In 2006, Xia and Yuan [28] introduced a formulation for the quadratic assignment problem using Gilmore-Lawler bounds to linearize the problem. The formulation can be tranformed to be used for the examination timetabling problem as:
(XYL)

$$\min \sum_{c\in C, t\in T} x_{ct}$$

s.t.

$$
\begin{array}{llll}
\sum_{t\in Tc} y_{ct} & = 1 & \forall c \in C & (5.9)\\
x_{c_1t_1} & \geq \sum_{c_2\in C, t_2\in T_{c_2}} c_{c_1c_2t_1t_2} y_{c_2t_2} - t_{c_1t_1}(1-y_{c_1t_1}) & \forall c_1 \in C, t_1 \in T_{c_1} & (5.10)\\
x_{ct} & \geq I_{ct} y_{ct} & \forall c \in C, t \in T_c & (5.11)\\
y_{c_1t} + y_{c_2t} & \leq 1 & \forall (c_1, c_2) \in S, t \in T & (5.12)\\
x_{ct} & \geq 0 & \forall c \in C, t \in T_c & (5.13)\\
y_{ct} & \in \{0,1\} & \forall c \in C, t \in T_c & (5.14)
\end{array}
$$

where I_{ct} is the optimal objective value of $(GLB)_{ct}$ and t_{ct} is the optimal objective value of the maximization version of $(GLB)_{ct}$, with the same objective function and constraints. I_{ct} is used to generate lower bounds for the variables and t_{ct} is used as a big-M representing the maximum cost of a given x_{ct} variable. In the model above, the objective function calculates the total cost created, constraint (5.10) determines the cost when a course is assigned to a timeslot and by constraint (5.11), Gilmore-Lawler lower bounds are used as lower bounds. Constraint (5.12) is the conflict constraint and other constraints are nonnegativity and binary constraints.

5.3 Bender's Decomposition for XYL

It is possible to apply Bender's algorithm to (XYL) formulation. In that case, x_{ct} variables can be transferred to the subproblem and the master problem consists of binary y_{ct} variables. The master problem is formulated as:

(MP)

$$\begin{array}{llll}
\min & \sum_{c\in C, t\in T_c} \alpha_{ct} & & \\
\text{s.t.} & & & \\
\sum_{t\in T_c} y_{ct} & = 1 & \forall c \in C & (5.15)\\
y_{c_1 t} + y_{c_2 t} & \leq 1 & \forall (c_1, c_2) \in S, t \in T & (5.16)\\
y_{ct} & \in \{0,1\} & \forall c \in C, k \in T_c & (5.17)
\end{array}$$

The subproblem of (XYL) is formulated below:
(SP)

$$\begin{array}{lllll}
\min & \sum_{c\in C, t\in T} x_{ct} & & & \\
\text{s.t.} & & & & \\
x_{c_1t_1} & \geq \sum_{c_2\in C, t_2\in T_{c_2}} c_{c_1c_2t_1t_2}\hat{y}_{c_2t_2} - t_{c_1t_1}(1-\hat{y}_{c_1t_1}) & \forall c_1 \in C, t_1 \in T_{c_1} & (5.18) & (v_{c_1t_1})\\
x_{ct} & \geq I_{ct}\hat{y}_{ct} & \forall c \in C, t \in T_c & (5.19) & (\mu_{ct})\\
x_{ct} & \geq 0 & \forall c \in C, t \in T_c & (5.20) &
\end{array}$$

where $\hat{y}_{c_1t_1}$ is the value of y_{ct} in the solution of the master problem. The dual variables corresponding to constraints (5.18) and (5.19) are $v_{c_1t_1}$ and μ_{ct}, respectively. The dual of the subproblem of (XYL) for each $c_1 \in C$, $t_1 \in T_{c_1}$ can be formulated as:
$(DSP)_{c_1t_1}$

$$\begin{array}{lll}
\max & I_{c_1t_1}\hat{y}_{c_1t_1}\mu_{c_1t_1} + t_{c_1t_1}\hat{y}_{c_1t_1} - t_{c_1t_1} + \sum_{c_2\in C, t_2\in T} c_{c_1c_2t_1t_2}\hat{y}_{c_2t_2}v_{c_1t_1} & \\
\text{s.t.} & & \\
\mu_{c_1t_1} + v_{c_1t_1} & \leq 1 & (5.21)\\
\mu_{c_1t_1} & \geq 0 & (5.22)\\
v_{c_1t_1} & \geq 0 & (5.23)
\end{array}$$

At each iteration, after solving the master problem, subproblems for each $c_1 \in C$, $t_1 \in T_{c_1}$ are solved and the constraint below for each $c_1 \in C$ and $t_1 \in T$ are evaluated:

$$\alpha_{c_1t_1} \geq I_{c_1t_1}y_{c_1t_1}\hat{\mu}_{c_1t_1} + t_{c_1t_1}y_{c_1t_1} - t_{c_1t_1} + \sum_{c_2\in C, t_2\in T_{c_2}} c_{c_1c_2t_1t_2}y_{c_2t_2}\widehat{v_{c_1t_1}} \quad \forall c_1 \in C, t_1 \in T_{c_1} \quad (5.24)$$

If any of constraints (5.24) is violated, it is added to the master problem. When there are no constraints of type (5.24) to add (all constraints of type (5.24) are satisfied in the master problem), the algorithm stops with optimality. As observed, since the master problem is feasible for all feasible solutions for the subproblem, feasibility cuts are not used like the Bender's decomposition applications in Chapter 3 and Chapter 4.

5.4 Numerical Results

(XYL) and its Bender's decomposition could solve $yue20023$ and $Bilkent100$ instances. Since Bender's applications take time, using compact (XYL) formulation seems better for these instances. For $Bilkent100$, the hardness property of exams is omitted as before. For $UTE92$, (XYL) could create slightly better cuts than the aforementioned formulations, however, it was not able to solve the problem. For $yue20011$, (XYL) could not create a positive lower bound although this problem has smaller dimensions than $UTE92$. We believe that this is because the number of conflicts in $yue20011$ is too small. No better improvement could be achieved via Bender's decomposition. $yue20023$ and $Bilkent100$ could be solved, however, the other two problems always gave zero objective value in the master problem for all iterations in a 3-hour run. Tables 5.1 and 5.2 show the best-found integer objective values and best lower bounds when problems are tried to be solved with a time limit of 3 hours using CPLEX 12.7 on Java on a UNIX platform provided by Bilkent University.

Problem	XYL	BDXYL
Bilkent100	42212*	42212*
UTE92	86057	-
yue20011	362	-
yue20023	56*	56*

Table 5.1: Best found integer objective values for K-B based formulations

Problem	XYL	BDXYL
Bilkent100	42212*	42212*
UTE92	2911	-
yue20011	0	-
yue20023	56*	56*

Table 5.2: Best found lower bounds for K-B based formulations

Chapter 6

Combination and Permutation Formulations

This chapter proposes two new formulations based on potential schedules of students. The first model uses the combinations of potential schedules of each student to calculate the total cost. After proposing the compact formulation, Bender's decomposition is applied to the model. As an extension, students whose set of exams are a subset of any other student's set of exams are determined and valid inequalities are added to the compact formulation. New variables are introduced to use these valid inequalities. Another formulation that uses the permutation of the students' potential schedules is proposed. Some branch and cut methods are applied to the formulation. Finally, the results are tested on $UTE92$ problem.

6.1 Combination Formulation

In this section, a new formulation that uses the potential schedules of students is proposed to solve the examination timetablig problem. Since each student must have a unique schedule where no two courses are assigned to the same timeslot, it is possible to use variables for the unique schedules of the exams taken by each student. If this methodology is used, exactly one schedule for each student must be selected out of the possible schedules that the student can have. A new model can be created considering all the possibilities of schedules for each student. Therefore, the objective function will be the total cost of all selected schedules for all students. The new model can be formulated using these additional sets, parameters and decision variables below:

Sets

- SC_s : set of schedules for student $s \in ST$.

Parameters

- E_{skt} : 1 if in k'th schedule student $s \in ST$ has a course assigned to timeslot $t \in T$, 0 otherwise.
- d_{sk} : the cost of schedule $k \in SC_s$ for student $s \in ST$.

Decision Variables

- z_{sk} : 1 if schedule $k \in SC_s$ is selected for student $s \in ST$, 0 otherwise.

Obviously, the number of elements in SC_s for a student $s \in ST$ is equal to the combination of $\binom{|T|}{|C_s|}$ since the matrix of possible schedules consists of zeros and ones where one in column t represents an assignment of a course taking by student s to timeslot t. Therefore, possible schedules for students taking the same number of courses are the same. For example, since $UTE92$ problem has 10 timeslots, a student taking three courses will have $\binom{10}{3}$ possible combinations and these combinations consist of three ones and seven zeros. In addition, since possible schedules for each student are determined by a combination, the number of schedules and the number of variables are directly affected by the number of timeslots in the problem. Therefore, it might be claimed that this model is more beneficial when it is applied to the problems having a small number of timeslots (eg. $UTE92$). The combination model is defined below:

(CP)

$$
\begin{array}{llll}
\min & \displaystyle\sum_{s \in ST, k \in SC_s} d_{sk} z_{sk} & & \\
\text{s.t.} & & & \\
\displaystyle\sum_{c \in C_s} y_{ct} = & \displaystyle\sum_{k \in SC_s | E_{skt}=1} z_{sk} & \forall s \in ST, t \in T & (6.1) \\
\displaystyle\sum_{k \in SC_s} z_{sk} = & 1 & \forall s \in ST & (6.2) \\
\displaystyle\sum_{t \in T_c} y_{ct} = & 1 & \forall c \in C & (6.3) \\
y_{c_1 t} + y_{c_2 t} \leq & 1 & \forall (c_1, c_2) \in S, t \in T & (6.4) \\
z_{sk} \geq & 0 & \forall s \in ST, k \in SC_s & (6.5) \\
y_{ct} \in & \{0,1\} & \forall c \in C, t \in T & (6.6)
\end{array}
$$

Constraint (6.1) relates y_{ct} variables to z_{sk} variables by making one of the z_{sk} variables 1 for each student $s \in ST$ having an exam scheduled to timeslot $t \in T$. The selected z_{sk} variable is the one that is consistent with the assignment of the exams that the student is taking. Constraint (6.2) ensures that each student must have a unique schedule. Constraint (6.3) guarantees that each exam is assigned to exactly one timeslot and constraint (6.4) blocks the conflict between exams. In addition, z_{sk} variables are selected to be continuous to lessen the number of binary variables. We can do that since constraint (6.1) guarantees that z_{sk} variables will be integer-valued. In addition, constraints (6.3) and (6.4) can be removed from the model since constraint (6.2) blocks the possible exam conflicts. Similarly, having constraints (6.3) and (6.4) also eliminates the necessity for constraint (6.2). Moreover, this model can be applied to the problem by making z_{sk} variables binary and y_{it} variables continuous. Therefore, one advantage of this model is that it gives us higher flexibility in terms of integer variable and constraint selection.

In addition, it is also possible to apply Bender's decomposition to that model. While doing that, for simplicity, y_{ct} variables are relaxed as continuous variables and z_{st} variables are made binary. In addition, constraints (6.3) and (6.4) are removed from the model, and constraint (6.1) is sent to the subproblem. Moreover, the constraints that makes y_{ct} variables less than or equal to 1 are not included to the subproblem since constraint (6.1) satisfies it naturally. Therefore, the the master problem, the subproblem, and the dual of it become:
(MP)

$$
\begin{array}{llll}
\min & \sum_{s \in ST, k \in SC_s} d_{sk} z_{sk} & & \\
\text{s.t.} & & & \\
\sum_{k \in SC_s} z_{sk} = & 1 & \forall s \in ST & (6.7) \\
z_{sk} \in & \{0,1\} & \forall s \in ST, k \in SC_s & (6.8)
\end{array}
$$

(SP)

$$
\begin{array}{lllll}
\min & 0 & & & \\
\text{s.t.} & & & & \\
\sum_{c \in C_s} y_{ct} = & \sum_{k \in SC_s | E_{skt}=1} \hat{z}_{sk} & \forall s \in ST, t \in T & (6.9) & (\lambda_{st})
\end{array}
$$

(DSP)

$$
\begin{array}{llll}
\max & \sum_{s \in ST, t \in T} \lambda_{st} \left(\sum_{k \in SC_s | E_{skt}=1} \hat{z}_{sk} \right) & & \\
\text{s.t.} & & & \\
\sum_{s \in ST | S_c = 1} \lambda_{st} \leq & 0 & \forall c \in C, t \in T & (6.10)
\end{array}
$$

λ_{st} in the SP is the dual variable corresponding to constraint (6.9). The master problem gives exactly one schedule for each student, however, these schedules do not satisfy the dependence between z_{sk} variables and y_{ct} variables. Hence, the subproblem SP will always be infeasible and the dual of the subproblem DSP will always be unbounded unless there is a feasible and optimal schedule in the master problem solution. Therefore, we need to find an extreme ray that causes unboundedness and add the constraint that restricts this ray to be positive. At each iteration, a feasibility cut of form

$$0 \geq \sum_{s \in ST, t \in T} \hat{\lambda}^r_{st} \Big(\sum_{k \in SC_s | E_{skt}=1} z_{sk} \Big) \quad (6.11)$$

must be added to the master problem.

6.2 An Extension

In some problems, there are some students who are taking a subset of the courses taken by another student. A student $s \in ST$ taking at least all of the courses taken by another student is called a superset student, and some other students taking only courses in elements of C_s are called subset students of s. We might use some constraints as valid inequalities to guarantee that the selected schedules for a superset student and the selected schedules for their subset students are coherent. To state the constraint, two new parameters are defined:

Parameters

• $subset_{s_1 s_2}$: 1 if the exam list of student $s_1 \in ST$ is a subset of the exam list of student $s_2 \in S$, 0 otherwise. ie. $subset_{s_1 s_2} = 1$ if $C_{s_1} \subset C_{s_2}$
• $app_{s_1 s_2 k_1 k_2}$: 1 if schedule $k_1 \in SC_{s_1}$ is a subset schedule of schedule $k_2 \in SC_{s_2}$ (eg. if $E_{s_1 k_1 t} = 1$ then $E_{s_2 k_2 t} = 1$ forall $t \in T$), 0 otherwise.

The subset constraint can be written for each z_{sk} variable for each superset/subset pairs and their potential schedules such as:

$$z_{s_1 k_1} \leq \sum_{k_2 \in SC_{s_2} | app_{s_1 s_2 k_1 k_2}=1} z_{s_2 k_2} \quad \forall s_1 s_2 \in ST \mid subset_{s_1 s_2} = 1, k_1 \in SC_{s_1} \quad (6.12)$$

However, the LP relaxation of the model becomes harder to solve when all these constraints are added. To use this extension with variables, we might use an additional variable for each student and for each timeslot. Hence, instead of deriving the model with a variable for an exam-timeslot assignment, we can use a variable for a student-timeslot assignment. To do that, another variable x_{st} is added. x_{st} is 1 if student $s \in ST$ taking an exam assigned to timeslot $t \in T$, 0 otherwise.

It is clear that x_{st} variables are either 0 or 1, and they are directly associated with the values of z_{sk} and y_{ct} variables, hence, there is no requirement to state them as a binary variable. However, making x_{st} variables binary will eliminate the necessity of making z_{sk} and y_{ct} variables binary. This can help us reduce the dimension of the problem since the number of timeslots is far less than the number of possible schedules. With the help of x_{st} variables, it is much easier to add subset valid inequalities to the model. To do that, constraints (6.1), (6.3), (6.4) and (6.5) are removed from (CP) and a new formulation is defined as:
$(CP2)$

$$
\begin{array}{llll}
\min & \sum_{s\in ST, k\in SC_s} d_{sk} z_{sk} & & \\
\text{s.t.} & & & \\
x_{st} & = \sum_{k\in SC_s | E_{skt}=1} z_{sk} & \forall s \in ST, t \in T & (6.13)\\
\sum_{c\in C_s} y_{ct} & = x_{st} & \forall s \in ST, t \in T & (6.14)\\
\sum_{t\in T} x_{st} & = |C_s| & \forall s \in ST & (6.15)\\
\sum_{k\in SC_s} z_{sk} & = 1 & \forall s \in ST & (6.16)\\
y_{ct} & \geq 0 & \forall c \in C, t \in T & (6.17)\\
z_{sk} & \geq 0 & \forall s \in ST, k \in SC_s \mid E_{skt} = 1 & (6.18)\\
x_{st} & \in \{0,1\} & \forall s \in ST, t \in T & (6.19)
\end{array}
$$

In addition, subset valid inequalities of form

$$x_{s_1 t} \leq x_{s_2 t} \quad \forall s_1, s_2 \in ST \mid subset_{s_1 s_2} = 1, t \in T \quad (6.20)$$

can also be added to the model and such constraints do not complicate the solution of the LP relaxation of the model.

6.3 Permutation Formulation

As a different approach, permutations (instead of combinations) of assignments can be used as variables for each student. Therefore, rather than using zeros and ones in E_{skt} parameter, we need to know which course is assigned to a given timeslot in each permutation of each schedule of each student. For convenience, the term "schedule" is used to describe the same concept in Section 6.1 and for each schedule, we will have $|C_s|!$ permutations. A schedule is the list consisting of $|T|$ elements where positive numbers imply that the student is taking an exam in that timeslot and zeros imply the student has no exam in that timeslot. In the permutation list, instead of using any positive value or zero, positive values imply the course number and zeros imply that the student has no exam in that timeslot. Therefore, permutation formulation has $\binom{T}{|C_s|} \times |C_s|!$ variables for each student. As an illustration, the table below gives schedule 2 (out of 120 schedules) and all 6 permutations of schedule 2 for student 3 who takes courses 56, 60 and 61 in $UTE92$.

Schedule 2	1	1	0	1	0	0	0	0	0	0
Permutation 1	56	60	0	61	0	0	0	0	0	0
Permutation 2	56	61	0	60	0	0	0	0	0	0
Permutation 3	60	56	0	61	0	0	0	0	0	0
Permutation 4	60	61	0	56	0	0	0	0	0	0
Permutation 5	61	56	0	60	0	0	0	0	0	0
Permutation 6	60	61	0	56	0	0	0	0	0	0

Table 6.1: A Combination/Permutation Schedule Example

The necessary sets, parameters and decision variables for the permutation formulation can be defined below:

Sets

- SC_s : set of schedules for student $s \in ST$.
- C_s : set of courses that student $s \in ST$ is taking.
- SV_{sk} : set of permutations of schedule $k \in SC_s$ for student $s \in ST$.

Parameters

- E_{skvt} : The course number of course assigned to timeslot $t \in T$ in permutation $v \in SV_{sk}$ in schedule $k \in SC_s$ for student $s \in ST$.
- d_{skv} : the cost of permutation $v \in SV_{sk}$ in schedule $k \in SC_s$ for student $s \in ST$.

Decision Variables

- z_{skv} : 1 if permutation $v \in SV_{sk}$ in schedule $k \in SC_s$ is selected for student $s \in ST$, 0 otherwise.

Then, the model can be formulated as:

(PP)

$$
\begin{array}{llll}
\min & \displaystyle\sum_{s\in ST,k\in SC_s,v\in SV_{sk}} d_{skv}z_{skv} & & \\
\text{s.t.} & & & \\
\displaystyle\sum_{c\in C_s} y_{ct} & = \displaystyle\sum_{k\in SC_s,v\in SV_{sk}|E_{skvt}>0} z_{skv} & \forall s \in ST, t \in T & (6.21)\\
\displaystyle\sum_{k\in SC_s,v\in SV_{sk}} z_{skv} & = 1 & \forall s \in ST & (6.22)\\
\displaystyle\sum_{t\in T_c} y_{ct} & = 1 & \forall c \in C & (6.23)\\
y_{c_1t} + y_{c_2t} & \leq 1 & \forall (c_1, c_2) \in S, t \in T & (6.24)\\
z_{skv} & \geq 0 & \forall s \in ST, k \in SC_s, v \in SVsk & (6.25)\\
y_{ct} & \in \{0, 1\} & \forall c \in C, t \in T & (6.26)
\end{array}
$$

The logic behind this model is exactly the same as that in Section 6.1. Constraint (6.21) builds the relation between y_{ct} and z_{skv} variables. Constraint (6.22) makes each student have exactly one schedule and exactly one permutation. Constraint (6.23) guarantees the assignment of each exam. Constraint (6.24) blocks conflicts among exams. Other constraints are nonnegativity and binary constraints.

Although (PP) has many more variables than (CP), with the extra information, it can be used to make the problem easier to solve. To add more beneficial cuts, a new dummy binary variable x_{sct} is created for all $s \in ST, c \in C_s, t \in T$ which is 1 if student $s \in ST$ takes the exam of course $c \in C_s$ at timeslot $t \in T$, 0 otherwise. Then, constraints below are added to the model:

$$\begin{aligned}
x_{sct} &= y_{ct} && \forall s \in ST, c \in C_s, t \in T_c && (6.27)\\
\sum_{c \in C_s} x_{sct} &= 1 && \forall s \in ST, t \in T_c && (6.28)\\
\sum_{t \in T_c} x_{sct} &= 1 && \forall s \in ST, c \in C_s && (6.29)\\
\sum_{s \in ST_s} x_{sct} &= m_c y_{ct} && \forall c \in C, t \in T_c && (6.30)
\end{aligned}$$

Constraint (6.27) guarantees that x_{sct} is one if the exam of the course $c \in C$ is assigned to timeslot $t \in T_c$. Constraint (6.28) blocks the conflicting exams. Constraint (6.29) satisfies the single assignment of each course. Constraint (6.30) is a valid inequality making the sum of x_{sct} variables equal to the number of students taking course $c \in C$, if the course is assigned to timeslot $t \in T_c$. In addition, subset constraints below are added to the model using x_{sct} variables to make the coherent assignment for subset students similarly in Section 6.2:

$$x_{s_1ct} = x_{s_2ct} \quad \forall s_1, s_2 \in ST \mid subset_{s_1s_2} = 1, c \in C, t \in T \mid c \in C_{s_1}, c \in C_{s_2} \quad (6.31)$$

The most significant benefit of this formulation is that it allows us to know a superset student's subset students' selected schedules and permutations when a permutation of a schedule is selected for a superset student. Therefore, it is possible to totally eliminate the subset students from the model and solve the model for only the superset students. Before doing that, the cost of subset students' permutations must be added to the cost of superset students' corresponding permutations. Then, the problem might be solved for fewer students, hence fewer variables.

Firstly, students taking 5 or 6 courses are marked as a superset and their subset students are selected and eliminated. Therefore, the number of students in $UTE92$ can be reduced from 861 to 585. Each of the 585 students is either a superset student or a student that is not a subset of any selected student. In addition, there is another option to choose the selected students as the minimum-length set of students covering all the courses. In this case, the set of superset students consists of some students that set of all courses taken by these students is equal to the set of courses C. However, it is observed that for $UTE92$, selecting all students taking 5 or 6 courses decreases the number of variables more effectively.

The formulation was tested on $UTE92$ using dummy variables x_{sct} and other variables in the model for the superset students and students that are not a subset of any selected student. However, there was a feasibility problem since the selected permutations for the students in the model are not coherent for students that are eliminated. To remove the infeasibility, x_{sct} variables are created for all $s \in ST$, $c \in C_s$, $t \in T$, rather than creating these variables only for students which are not subset of another student. Let the set of used students (selected students and the students that are not a subset of any selected student) be US and the parameter us_s shows the superset student number of eliminated student $s \in ST - US$. Then, constraint (6.21) is written as:

$$\sum_{c \in C_s} x_{sct} = \sum_{k \in SC_s, v \in SVsk | E_{skvt} > 0} z_{skv} \qquad \forall s \in US, t \in T \qquad (6.32)$$

$$\sum_{c \in C_s} x_{sct} = \sum_{k \in SC_{us_s}, v \in SVus_sk | E_{us_skvt} > 0} z_{us_skv} \qquad \forall s \in ST - US, t \in T \qquad (6.33)$$

That version solves the feasibility problem, however, solution procedure is much more inefficient compared to the model in Section 6.1. Another possibility to eliminate the feasibility problem is to add the constraint below to the model for each $c \in C$ and $t \in T$:

$$m_c y_{ct} = \sum_{s \in ST_c, k \in SC_s, v \in SVsk | E_{skvt} = c} z_{skv} \quad \forall c \in C, t \in T \quad (6.34)$$

The constraint above is added to the model and constraint (6.21) is used instead of constraints (6.32) and (6.33) for simplicity. This version of the problem is tested for all students and it could not be solved since solving the LP relaxation is hard. To try another approach, constraint (6.21) is removed from the model since constraint (6.34) may guarantee that constraint (6.21) is satisfied. With this approach, an inefficient solution procedure is reached and feasibility is observed.

Since either constraint (6.21) or (6.34) is enough to satisfy the feasibility requirements, only one of them can be added directly or in a branch and cut manner. These attempts were not successful since branch and cut operation blocks the dual presolve in the CPLEX solver and eliminating some unnecessary variables in the model.

6.4 Numerical Results

Models in this chapter can create much more effective cuts than the other suggested models in this thesis. In addition, in this formulation, it is possible to make only one of z_{sk}, x_{st} or y_{ct} variables binary. All three possibilities are tested, and it is concluded that the best results could be achieved by making only y_{ct} variables binary. A starting solution of the best solution found (objective value of 68090) in the literature is given since the optimality gap is too small. Table 6.2 shows the best-found integer objective values and best lower bounds when $UTE92$ problem is tried to be solved with a time limit of 3 hours using CPLEX 12.7 on Java on a UNIX platform provided by Bilkent University.

Problem	CP	BDCP	PP
Best objective	68090	-	68090
Best bound	64022	-	65648

Table 6.2: Best found integer objective values and lower bounds for combination/permutation formulations

It is observed that Bender's decomposition is not efficient for the permutation formulation. CP and PP give the best lower bounds in comparison with other formulations. In addition, for $UTE92$ problem, PP finds slightly better lower bounds. With PP, in longer runs, a lower bound of 66849.06 was derived and it is proven that the best solution found so far for $UTE92$ problem has an optimality gap of 1.82%. One of the main contributions of this thesis is providing effective lower bounds via these formulations.

Chapter 7

Lower Bounds and Matheuristics

7.1 A Lower Bound for the Examination Timetabling Problem

Since solving the examination timetabling problem to optimality can be hard for many real-world instances, finding lower bounds might be beneficial. Fischetti et al. [30] stated a theorem that can be used to find a lower bound for the quadratic assignment problem by dividing the flow (cost) matrix a, which is the same as the conflict matrix in the examination timetabling problem, into two. The conflict matrix a is the matrix consists of the number of students taking both courses. For example, $a_{c_1 c_2}$ describes the number of students taking both courses $c_1 \in C$ and $c_2 \in C$.Hence, smaller problems might be created and solved with this approach. In the paper, it is proven that the sum of the objective values of these two problems is a lower bound for the original problem. The formal theorem is given below:

Theorem 1 (Fischetti) *Let a^1 and a^2 be nonnegative matrices creating a partition of the cost matrix a (eg. $a^1 + a^2 = a \mid a^1, a^2 \geq 0$) and opt(a) be the objective value of a particular examination timetabling problem for cost matrix a.*

Then, $opt(a^1) + opt(a^2) \leq opt(a)$ *.*

Using this method, the problem might be partitioned into two smaller problems, and by solving these smaller problems we can get a lower bound for the original problem. Therefore, we can have an idea about the optimality gap for the best-found solution. In addition, this lower bound can be used to prove optimality if a feasible solution has the same objective value with the lower bound that is found using this way.

In the examination timetabling problem, a matrix consists of elements as the number of common students between each course pair. The original problem can be divided into two and be solved for two different exam sets using the formulation in Section 3.2. In addition, it is also possible to apply this lower bound calculation method to the formulation in Section 6.1 by dividing the set of students. This lower bound method is used for $UTE92$ problem and it was hard to solve the problem even if the conflict matrix a was divided into two. Then, this method was tried to solve $yue20023$ problem to test it. Many different values were tried as a^1. As an option, the values in the matrix a^1 can be selected as the half of a matrix. In this case, a^1 is defined as $a^1_{c_1c_2} = a^2_{c_1c_2} = 0.5a_{c_1c_2} \forall (c_1, c_2) \in S$. Therefore, if 5 students are taking both courses $c_1 \in C$ and $c_2 \in C$, we create two problems with $a^1_{c_1c_2} = a^2_{c_1c_2} = \frac{5}{2}$. Another option is to randomly select a given number of courses and construct a^1 in such a way that only the values between the selected courses have positive values in matrix a^1 and all other values are represented in a^2. In addition, as Fischetti et al. [30] proposed, support of matrix a is also tried ($a^1_{c_1c_2} = \min\{a_{c_1c_2}, 1\}$ and $a^2_{c_1c_2} = a_{ij} - a^1_{c_1c_2}$). In this case, if 5 students are taking both courses $c_1 \in C$ and $c_2 \in C$, $a^1_{c_1c_2}$ is taken as 1 and $a^2_{c_1c_2}$ is taken as 4. Using support of a, the best lower bound obtained through splitting was 50 for $yue20023$ instance for which the optimal solution was 56. In addition, the best results are found by selecting a^1 as the support of a. It is also possible to divide the a matrix by three or more but this gives worse results due to the intersection between separated problems.

7.2 Special Attempts to Find Lower Bounds for UTE92 Problem

After the shrinking operation that we used before, $UTE92$ problem has 861 students. When there are not many timeslots and the average number of courses taken by a student is high with respect to the number of timeslots in a specific problem, it is easier to calculate a lower bound just by scheduling the exams taken by the students who take a large number of exams. In $UTE92$ problem, there are 10 timeslots and students take courses up to 6. Therefore, just calculating the minimum costs for each student, it is possible to find a minimum assignment strategy for students who take 1 to 6 courses in $UTE92$ problem. For instance, it is calculated that a student taking 6 courses has a cost of 59 in the best-case scenario. Therefore, by summation of these minimum best-case scenario costs of all students, it is possible to have a cost of 59160, which is a lower bound for the problem.

Since this lower bound is calculated by considering the feasibility for only one student, it might be wiser to find a lower bound considering the feasibility of all students while calculating the cost created by only one student. A compact formulation of all students was tried iteratively to minimize the cost of any student. This attempt was not successful since it was detected that the minimum cost calculated for any student can be achieved in a feasible assignment for all students. Namely, a feasible assignment can be created for all students if our aim is to minimize the cost related to a given student. The same method was used to calculate the cost created by only a small subset of students (eg. 20 students), however, this attempt was not successful.

7.3 A Matheuristic for the Examination Timetabling Problem

A matheuristic is designed for Bilkent University examination timetabling problem. The inspiration of the matheuristic is sourced by the notice that most real-world examination timetabling problems can be solved with a fewer number of variables when some exams are assigned to given timeslots to decrease the number of assignments. When some variables are fixed to some feasible values it becomes much easier to find the optimal solution for unfixed variables since this operation decreases the number of variables and reduces the feasible region. Then, fixing some variables, other variables can be optimized in a small amount of time. In addition, for the cases that are hard to optimize in a short duration, since the number of variables is less, a good improvement on the solution can be achieved easily. A very similar approach is applied to well-known examination timetabling instances by Arbaoui et al. [16].

To apply the matheuristic, firstly, a feasible solution is found as a starting point. Then, some variables in the model are fixed and the problem is solved for the unfixed variables. Solving iteratively, the objective value is improved at each iteration. When the difference between the objective values is less than a specific value, or the number of iterations exceeds a certain number, the process is done. A formal pseudo-code of the matheuristic for a problem having n courses, m timeslots and so $m \times n$ binary variables is proposed in Algorithm 2.

The pseudo-code explains how the algorithm can be applied to the formulation in section 3.2. In addition, this matheuristic can also be applied to another mathematical model as it is done by Arbaoui et. al. [16]. In that model, x_{st}, the cost of student $s \in ST$ that is created by the assignment of any course taken by student $s \in ST$ to timeslot $t \in T$, is used as a decision variable. Logically, the cost can be positive for a student and a timeslot if the student has an exam in that timeslot. Including others introduced in the thesis above, the additional

Algorithm 2 Examination Timetabling Matheuristic

Require: $P \subset C$, $l \geq 0$, $|P| \geq 0$

1: Find a feasible solution S
2: **for** $c \in C$, $t \in T$ **do**
3:     **if** $c \in C$ is assigned to timeslot $t \in T$ in solution S **then**
4:         $z_{ct} \leftarrow 1$
5:     **else**
6:         $z_{ct} \leftarrow 0$
7:     **end if**
8: **end for**
9: **while** Iteration no $\leq k$ **do**
10:     **for** $p \in P$, $t \in T$ **do**
11:         Add an additional constraint $x_{pt} = z_{pt}$
12:     **end for**
13:     Solve the assignment problem for l seconds and find solution S'
14:     **if** A good improvement is achieved **then**
15:         $S \leftarrow S'$
16:         Remove all additional constraints
17:         Keep $|P|$, change P and go to step 2
18:     **else**
19:         Remove all additional constraints
20:         Increase $|P|$, change P and go to step 2
21:     **end if**
22:     Iteration no $\leftarrow$ Iteration no + 1
23: **end while**

Figure 7.1: Examination Timetabling Matheuristic

parameters and decision variables needed for the model are described below:

Parameters

- $d_{t_1 t_2}$: the cost incurred between timeslots $t_1 t_2 \in T$.

Decision Variables

• x_{st} : the cost of student $s \in ST$ created by timeslot $t \in T$.

• y_{ct} : 1 if course $c \in C$ is assigned to timeslot $t \in T$, 0 otherwise.

The mathematical model is introduced below:

(STP)

$$
\begin{array}{llllr}
\min & \sum\limits_{s\in ST, t\in T} x_{st} & & & \\
\text{s.t.} & & & & \\
\sum\limits_{t\in T_c} y_{ct} & = & 1 & \forall c \in C & (7.1)\\
x_{st_1} & \geq & \sum\limits_{c\in C_s, t_2\in T | t_2\neq t_1} d_{t_1t_2} y_{ct_2} + M((\sum\limits_{c\in C_s} y_{ct_1}) - 1) & \forall s \in ST, t_1 \in T & (7.2)\\
\sum\limits_{t\in T} x_{st} & \geq & mn_s & \forall s \in ST & (7.3)\\
\sum\limits_{t\in T} x_{st} & \leq & mx_s & \forall s \in ST & (7.4)\\
y_{c_1t} + y_{c_2t} & \leq & 1 & \forall (c_1c_2) \in S, t \in T & (7.5)\\
x_{st} & \geq & 0 & \forall s \in S, t \in T & (7.6)\\
y_{ct} & \in & \{0,1\} & \forall c \in C, t \in T & (7.7)
\end{array}
$$

where M is a large integer that is greater than the largest cost of one student and mn_s and mx_s are the minimum and maximum costs that might be induced by student $s \in ST$. Therefore, it is wiser to set M as mx_s for each $s \in ST$ in constraint (7.2). Constraint (7.1) is the assignment constraint. Constraint (7.2) builds the relation between x_{st} and y_{ct} variables. Constraints (7.3) and (7.4) gives bounds for x_{st} variables. Constraint (7.5) is the conflict constraint. Other constraints are nonnegativity and binary constraints.

In the matheuristic, some exam assignments are fixed to some given timeslots in a feasible assignment (some y_{ct} variables are fixed) and an optimization problem is solved and this procedure is applied iteratively. In addition, since constraint (7.2) increases the size of the problem extremely, this constraint can be deleted for the fixed courses in each iteration. In addition, the same methodology can be used by fixing the cost incurred for some students (some x_{st} variables are fixed), however, this approach was not successful when it was tested. Table 7.1 shows the results for Arbaoui et al. [16] (STP) formulation tested on some Toronto instances. Since the values are stated as the average cost per student in the paper, the values are converted to the total cost, as we use in the thesis.

	Solution found	Best solution found until now	Gap
EAR83	41287	32962	25.2%
HEC92	30008	25971	15.5%
LSE91	28077	26169	7.2%
STA83	96073	95945	0.1%
TRE92	49660	33790	46.9%
YOR83	38054	32784	16.0%

Table 7.1: Matheuristic results for Arbaoui et al. formulation

Deciding which assignments to fix is a dynamic process, exams can be fixed randomly or in a specific rule such as fixing the first 100 courses taken by the maximum number of students and second 100 exams in the second iteration. The number of exams fixed is another critical decision. Although it might be logical to fix a number of exams so that the unfixed exams can be assigned optimally, a time limit can be set and there is no necessity to wait for reaching the optimal solution.

It is proven by our work and the work of Arbaoui et al. [16] that this matheuristic gives good results when it is applied to these two formulations above. Although the gap between the objective value of the suboptimal solution and the objective value of the optimal solution is not known, it is believed that this method creates good solutions since the solutions found are very close to the best solutions found ever for the instances tested.

In this thesis, this matheuristic is applied to the formulation in Chapter 3.2. For *Bilkent*, instead of four-indexed variables, three types of variables are used to calculate the cost of consecutive timeslots, consecutive days, and same day assignments since creating four-indexed variables exceeds the available memory. A time limit of 120 seconds is set for each iteration and a given number of courses is fixed randomly at each iteration. This number must be determined by considering the size of the problem. Starting solutions of the matheuristic are determined by the formulation stated in Section 4.1. Table 7.2 shows the best-found integer objective values when problems are tried to solve up to 700 iterations with a time limit of 120 seconds using CPLEX 12.7 on Java on a UNIX platform provided by Bilkent University.

	Solution found	Best solution found until now	Gap
Bilkent	70426	45964	53.2%
EAR83	42167	32962	27.9%
HEC92	31259	25971	20.3%
LSE91	34472	26169	31.7%
STA83	96263	95945	0.3%
TRE92	38959	33790	15.2%
UTE92	69028	68090	1.3%
YOR83	37920	32784	15.6%
yue20011	62	56	10.7%
yue20023	56	56	0.0%

Table 7.2: Matheuristic results

These results show that the matheuristic provides solutions very close to the best-found values in the literature with a reasonable number of iterations. For *Bilkent*, the best solution found was also achieved via the matheuristic using 500 seconds at each iteration, with a large number of iterations. This demonstrates that even with a large instance it is possible to find good solutions using the matheuristic. Although the matheuristic does not guarantee a specific optimality gap it is possible to find good solutions in a relatively short time period. However, for some instances, a feasible solution could not be found in a compact formulation. This problem was solved with solving the problem with an objective function of 0 at first to find a feasible solution.

Chapter 8

Conclusion and Future Work

In this thesis, successful exact solution methodologies applied to similar problems in the literature are implemented to the examination timetabling problem. We obey the basic examination timetabling hard constraints such as no conflict, and add additional exam and instructor constraints considering the characteristic of the problem. In addition, we aggregate the students taking exactly the same courses to decrease the problem dimensions whenever viable.

In the first part of the thesis, formulations applied to the p-hub median hub location problem and the quadratic assignment problem are tried to solve well-known examination timetabling problems from the literature. It is shown that these formulations give exact optimal solutions for small dimensions, and when some exam assignments are fixed to given timeslots, it is possible to solve the remaining problem optimally. However, real-world problems are generally much larger than the ones that could be solved, and these formulations were not effective to solve these problems. Bender's decomposition algorithms are applied to these formulations and some valid cuts are added, however, for the unsolved large problem instances, these attempts mostly fail.

Combination and permutation formulations are stated for problems with less number of timeslots and high average course taken by a student/number of timeslots ratio. We applied these formulations to a well-known instance $UTE92$ in $Toronto$ dataset. In this way, using the bound we found, we proved that the best solution found so far is in the 1.82% of optimality. This is proven for the first time in the literature. In addition, the information of students taking the set of exams which is a subset of another student's exam set is used through adding constraints. For the permutation formulation, students whose courses taken are a subset of another student are eliminated, and the problem dimension can be made smaller. Branch and cut method is also applied to the permutation problems.

It was observed that adopted formulations did not provide good lower bounds. XYL formulation is slightly better than other formulations in terms of lower bounds. However, flow formulations $FP1$ and $FP2$ give slightly better suboptimal solutions in three hours. Combination and permutation formulations give much better lower bounds for $UTE92$ instance. These formulations can be used to prove the optimality of a solution. We suggest using these formulations for small instanes. Combination and permutation formulations are useful for large instances that have a small number of timeslots. For large instances that have a large number of timeslots, it was observed that these formulations were not helpful. Hence, for these instances, we suggest using the matheuristic that we provide in this thesis. Table 8.1 gives the combined table for computations used in this thesis.

Formulations	Bilkent100		UTE92		yue20011		yue20023	
Instances	Bound	Solution	Bound	Solution	Bound	Solution	Bound	Solution
P	42212*	42212*	1131	85827	-	-	56*	56*
BDFC	42212*	42212*	-	-	-	-	56*	56*
BDWFC	42212*	42212*	-	-	-	-	56*	56*
FP1	42212*	42212*	900	81686	NA	NA	56*	56*
FP2	42212*	42212*	844	83206	NA	NA	56*	56*
BDFP	42212*	42212*	-	-	NA	NA	56*	56*
LYG	42212*	42212*	-	-	NA	NA	56*	56*
XYL	42212*	42212*	2911	86057	0	362	56*	56*
BDXYL	42212*	42212*	-	-	-	-	56*	56*
CP	NA	NA	64022	68090	NA	NA	NA	NA
BDCP	NA	NA	-	-	NA	NA	NA	NA
PP	NA	NA	65648	68090	NA	NA	NA	NA

Table 8.1: Computational Results

Lower bound calculation methods are applied to the instances in the literature. Efficient and simple lower bounds can be calculated by solving the problem for each individual student and benefiting from different methodologies used by Fischetti et al. [30], Gilmore [25] and Lawler [26] for the quadratic assignment problems. These lower bounds are calculated in a short time and can be added to any model for the examination timetabling problem to strengthen the formulation. Finally, a matheuristic is stated and it is proven that it gives good results for many large instances in the literature in a significantly short time.

As future work, in this study, we set our objective function as to minimize the sum of the soft constraints which are created by the closeness of exams for each student. In the literature, there are problems to minimize various objectives such as the total number of timeslots used in an assignment. In addition, some problems are solved to just find a feasible assignment rather than using an objective. We believe formulations stated in this thesis might be beneficial for such considerations as well. Moreover, branch and price applications might be applied to the formulations stated in the thesis to solve the examination timetabling problem. These applications might be beneficial for the formulations in Chapter 6 since these formulations have too many variables to handle.

Bibliography

[1] M. W. Carter, G. Laporte, and S. Y. Lee, “Examination timetabling: Algorithmic strategies and applications,” *The Journal of the Operational Research Society*, vol. 47, no. 3, p. 373, 1996.

[2] R. K. Qu, E. T. G. Burke, B. Y. Mccollum, L. u. Merlot, and S. u. Lee, “A survey of search methodologies and automated system development for examination timetabling,” *Journal of Scheduling*, vol. 12, no. 1, pp. 55 – 89, 2008.

[3] E. K. Burke, J. P. Newall, and R. F. Weare, “A memetic algorithm for university exam timetabling,” *Practice and Theory of Automated Timetabling Lecture Notes in Computer Science*, pp. 241 – 250, 1996.

[4] A. J. Parkes and E. Ozcan, “Properties of yeditepe examination timetabling benchmark instances,” *PATAT 2010*, May 2010.

[5] S. Even, A. Itai, and A. Shamir, “On the complexity of timetable and multicommodity flow problems,” *SIAM Journal on Computing*, vol. 5, no. 4, pp. 691 – 703, 1976.

[6] E. Burke and J. Newall, “A multistage evolutionary algorithm for the timetable problem,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 1, pp. 63 – 74, 1999.

[7] P. Ross, D. Corne, and H. Terashima-Marin, “The phase-transition niche for evolutionary algorithms in timetabling,” *Practice and Theory of Automated Timetabling Lecture Notes in Computer Science*, pp. 309 – 324, 1996.

[8] Ö. Ülker, E. Özcan, and E. E. Korkmaz, "Linear linkage encoding in grouping problems: Applications on graph coloring and timetabling," *Practice and Theory of Automated Timetabling VI Lecture Notes in Computer Science*, pp. 347 – 363, 2007.

[9] L. D. Gaspero and A. Schaerf, "Tabu search techniques for examination timetabling," *Lecture Notes in Computer Science Practice and Theory of Automated Timetabling III*, pp. 104 – 117, 2001.

[10] G. M. White, B. S. Xie, and S. Zonjic, "Using tabu search with longer-term memory and relaxation to create examination timetables," *European Journal of Operational Research*, vol. 153, no. 1, pp. 80 – 91, 2004.

[11] J. M. Thompson and K. A. Dowsland, "A robust simulated annealing based examination timetabling system," *Computers & Operations Research*, vol. 25, no. 7-8, pp. 637 – 648, 1998.

[12] L. T. G. Merlot, N. Boland, B. D. Hughes, and P. J. Stuckey, "A hybrid algorithm for the examination timetabling problem," *Practice and Theory of Automated Timetabling IV Lecture Notes in Computer Science*, pp. 207 – 231, 2003.

[13] K. A. Dowsland and J. M. Thompson, "Ant colony optimization for the examination scheduling problem," *Journal of the Operational Research Society*, vol. 56, no. 4, pp. 426 – 438, 2005.

[14] G. Kendall and N. M. Hussin, "An investigation of a tabu-search-based hyper-heuristic for examination timetabling," *Multidisciplinary Scheduling: Theory and Applications*, pp. 309 – 328, 2005.

[15] R. Qu and E. K. Burke, "Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems," *Journal of the Operational Research Society*, vol. 60, no. 9, pp. 1273 – 1285, 2009.

[16] T. Arbaoui, J.-P. Boufflet, and A. Moukrim, "A matheuristic for exam timetabling," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1289 – 1294, 2016.

[17] B. Mccollum, P. Mcmullan, A. J. Parkes, E. K. Burke, and R. Qu, "A new model for automated examination timetabling," *Annals of Operations Research*, vol. 194, pp. 291 – 315, Jun 2011.

[18] A. Parkes and E. Ozcan, "Properties of yeditepe examination timetabling benchmark instances," pp. 531–534, 01 2010.

[19] A. T. Ernst and M. Krishnamoorthy, "Efficient algorithms for the uncapacitated single allocation p-hub median problem," *Location Science*, vol. 4, no. 3, pp. 139 – 154, 1996.

[20] C.-C. Lin, J.-Y. Lin, and Y.-C. Chen, "The capacitated p-hub median problem with integral constraints: An application to a chinese air cargo network," *Applied Mathematical Modelling*, vol. 36, p. 2777–2787, Jun 2012.

[21] D. Skorin-Kapov, J. Skorin-Kapov, and M. Okelly, "Tight linear programming relaxations of uncapacitated p-hub median problems," *European Journal of Operational Research*, vol. 94, no. 3, pp. 582 – 593, 1996.

[22] M. Labbe, H. Yaman, and E. Gourdin, "A branch and cut algorithm for hub location problems with single assignment," *Mathematical Programming*, vol. 102, pp. 371 – 405, Jul 2004.

[23] H. Mokhtar, M. Krishnamoorthy, and A. T. Ernst, "A modified benders method for the single- and multiple allocation p-hub median problems," *Operations Research Proceedings Operations Research Proceedings 2017*, pp. 135 – 141, 2018.

[24] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica*, vol. 25, no. 1, p. 53, 1957.

[25] P. C. Gilmore, "Optimal and suboptimal algorithms for the quadratic assignment problem," *Journal of the Society for Industrial and Applied Mathematics*, vol. 10, no. 2, pp. 305 – 313, 1962.

[26] E. L. Lawler, "The quadratic assignment problem," *Management Science*, vol. 9, no. 4, pp. 586 – 599, 1963.

[27] L. Kaufman and F. Broeckx, “An algorithm for the quadratic assignment problem using benders decomposition,” *European Journal of Operational Research*, vol. 2, no. 3, pp. 207 – 211, 1978.

[28] Y. Xia and Y.-X. Yuan, “A new linearization method for quadratic assignment problems,” *Optimization Methods and Software*, vol. 21, no. 5, pp. 805 – 818, 2006.

[29] H. Zhang, C. Beltran-Royo, and L. Ma, “Solving the quadratic assignment problem by means of general purpose mixed integer linear programming solvers,” *Annals of Operations Research*, vol. 207, pp. 261 – 278, Jul 2012.

[30] M. Fischetti, M. Monaci, and D. Salvagnin, “Three ideas for the quadratic assignment problem,” *Operations Research*, vol. 60, no. 4, pp. 954 – 964, 2012.