A MATHEMATICAL MODELLING APPROACH FOR EXAM TIMETABLING

by

Mehmet Fatih ACAR

February 2010

A MATHEMATICAL MODELLING APPROACH FOR EXAM TIMETABLING

by

Mehmet Fatih Acar

A thesis submitted to

the Graduate Institute of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Industrial Engineering

February 2010
Istanbul, Turkey

APPROVAL PAGE

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Mehmet ŞEVKLİ
Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Mehmet ŞEVKLİ
Supervisor

Examining Committee Members

Prof. Dr. Selim ZAİM..

Assist. Prof. Dr. Mehmet ŞEVKLİ..

Assist. Prof. Dr. Özgür UYSAL ....

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate Institute of Sciences and Engineering of Fatih University.

Assoc. Prof. Dr. Nurullah ARSLAN
Director

February 2010

A MATHEMATICAL MODELLING APPROACH FOR EXAM TIMETABLING

Mehmet Fatih ACAR

M.S. Thesis – Industrial Engineering
February 2010

Supervisor: Assist. Prof. Dr. Mehmet ŞEVKLİ

ABSTRACT

Exam timetabling problems are very popular problems in academic enviroments. Many lecturers and students are not happy for their schuduled plans. These schedules are sometimes done with manual, so people face lots of problems such as; having more than one exam in the same time slot or in the same day. In addition to these, there are some restrictions such as, capacity of rooms or number of inviligators. In this research, exam timetabling problem is solved with considering desires of both lecturers and students. For exam timetabling problems, a new mathematical model is generated. However, this mathematical model can not solve large size problems in short time, so two heuristic methods based on the mathematical model is constructed. Main idea of these heuristics is clustering method. In this thesis, Xpress-MP software which is one of the most popular programmes in optimization area is used . Moreover, second heuristic method is applied for Fatih University dataset.

Key Words: Exam Timetabling, Mathematical Modelling, Heuristic Methods

SINAV ÇİZELGELEMESİ İÇİN MATEMATİKSEL MODEL YAKLAŞIMI

Mehmet Fatih ACAR

Yüksek Lisans Tezi – Endüstri Mühendisliği
Şubat 2010

Tez Yöneticisi: Yrd. Doç. Dr. Mehmet ŞEVKLİ

ÖZ

Sınav çizelgeleme problemi akademik ortamlarda karşılaşılan en popüler problemlerden biridir. Bir çok hoca ve öğrenci kendi çizelgelerinden memnun olmamaktadır. Bu çizelgelemeler elle yapılabilmekte, dolayısyla öğrencinin aynı zamanda veya aynı günde iki veya daha fazla sınavı olması gibi çeşitli problemler ortaya çıkabilmektedir. Bununla birlikte sınıfların kapasitesi, gözetmen sayısı gibi kısıtlardan da bahsedilebilir. Bu çalışmada öğretim üyelerinin ve öğrencilerin istekleri göz önünde bulundurularak sınav çizelgeleme problemi çözülmeye çalışılmıştır. Bu probleme çözüm üretmek için yeni bir matematiksel model oluşturulmuştur. Bu matematiksel model büyük verilere sahip problemleri kısa zamanda çözemediği için matematiksel modellemeye dayalı iki yeni sezgisel yöntem geliştirilmiştir. Bu iki yeni sezgisel yöntemin ana fikri sınıflandırma yapmaktır. Bu araştırmada eniyileme alanında ki en popüler yazılımlardan biri olan Xpress-MP adlı yazılım kullanılmış ve bununla birlikte geliştirilen ikinci sezgisel yöntem Fatih Üniversitesi verilerine uygulanmıştır.

Anahtar Kelimeler: Sınav Çizelgeleme Problemi, Matematiksel Modelleme, Sezgisel Yöntemler

DEDICATION

Dedicated to my parents for their endless support and patience during the forming phase of this thesis.

ACKNOWLEDGEMENT

I would like to express my gratitude to my supervisor Assist. Prof. Dr. Mehmet ŞEVKLİ whose help, sitimulating suggestions and encouragement helped me during the course of research and writing of this thesis.

I also want to thank to Prof. Dr. Selim ZAİM, Assist. Prof. Dr. Fahrettin ELDEMİR and Assist. Prof. Özgür UYSAL for their significant contributions to my research endevaors.

I would also like to thank to The Graduate Institute of Sciences and Engineering of Fatih University.

I express my thanks and appreciation to my family for their understanding, motivation and patience. Lastly, but in no sense the least, I am thankful to all colleagues and friends who made me stay at the university a memorable and valuable experience.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

CHAPTER 1

INTRODUCTION

Timetabling is a very popular subject area in today's academic environment because many real life problems can be expressed as time scheduling. Exam and course scheduling, determination of train and bus departure hours, rostering for nurses in hospitals are examples of timetabling problems. Lots of academic people struggle to solve these types of timetabling problems, because inefficiencies can be observed frequently in the schools, hospitals and public transportations. However timetabling problems are difficult (NP- complete). (Even et. al.1976)

1.1. BACKGROUND

Burke et al. (2004a) defined a timetabling problem as; "A timetabling problem is a problem with four parameters: T, a finite set of times; R, a finite set of resources ; M, a finite set of meetings; and C, a finite set of constraints. The problem is to assign times and resources to the meetings so as to satisfy the constraints as far as possible."

Among timetabling problems, exam and course timetabling problems are some of the most widely studied. Academic institutions may face this problem in each semester. The quality of scheduling has an impact on the students, lecturers and administrators. Especially, in the beginning of semesters, lots of students come to their advisors and tell their problems during course selection. Furthermore, exam timetabling is one of the most popular problem in the schools during the final exam days, because some students may have more than one exam in the same time and capacity of rooms, number of timeslots, number of assistants who are duties in the exams are some important constraints. There is no general algorithm to solve all exam timetabling problems, so each problem in the different schools can be new case studies for literature and can be solved by different methods.

Mathematical Programming (MP) provides us obtaining optimum solution for these problems. It is one of the most popular operational research techniques to reach the best solution. Especially after 1950s this method has been used in many areas and using this methodology, lots of problems can be solved in the different industry. However it can be said that mathematical modelling can not solve big problems in the short time, so in the literature heuristic methods are used to solve problems. There are lots of heuristics and meta-heuristic methods which can be considered in the timetabling problems such as; Genetic Algorithms(GA), Tabu Search, Local Search (LS) etc.

In the literature, there are two types of constraints; hard and soft constraints. Hard constraints can not be violated in any situation. For example, any exam can not be organized more than one time slots. If any solution satisfies all hard constraints, then this is the feasible solution.

In addition to these, soft constraints are desirable but not obligatory. In practice, it is not very easy to find a feasible solution which satisfies all soft constraints. Soft constraints also vary from one institution to other (Burke 1996a). One of the most popular soft constraints is spreading exams efficiently that students do not enter exams in consecutively periods. Moreover, another soft constraint is organizing large exams as early as possible. The quality of exams is generally measured by checking of which soft constraints are violated (Qu et. al. 2009).

In the literature; there are two hard constraints about exam timetabling problems. One of them is nobody has two exams in any given time. Furthermore, resources like room capacities must not be violated.

Furthermore, many soft constraints can be seen in different papers. These are

-large exams should be planned as early as possible,

-groups of some exams should be scheduled in the same timeslot,

-some exams should be organized before than some exams,

-there is a capacity constraint in any timeslot,

-exams should be in a similar location like in the same building.

Abdullah (2006) summarised the basic terminolgy is used in the exam timetabling. Basic terminology is given below;

Table 1.1: Basic terminology

Event:	An activity which is scheduled, for example; exams or courses.
Timeslot:	An interval time for scheduled events.
Resource:	Resources for exams; for instance, room and inviligator.
Constraint:	A restriction for timetabling, like room capacity.
Individual:	A person who are in events,such as; students and inviligators.
Conflict:	Clash of exams which have common individual(s) in the same time.

In timetabling problems, a number of events like exams or courses are struggled to assign into limited number of timeslots and rooms with considering of hard and soft constraints. In the literature, the exam timetabling problems have a set of events, E, a set of timeslots, T, and a set of hard constraints, C.

de Werra (1985) presented a mathematical model for exam timetabling problems. Its notation is

- E is the set of exams,
- T is the number of timeslots,
- C(i,t) is the cost of scheduling if exam i is in timeslot t,
- Y(i,t) = 1 if exam i is scheduled in timeslot t, 0 otherwise,
- X(i,j) = 1 if exam i clashes with exam j, 0 otherwise,.
- X(t) = maxsimum number of scheduled exams in the period t.

Objective function of this model is minimizing the cost of scheduled exam i in period t. This model can be shown (Terashima-Marin, 1998).

$$\text{Min} \quad \sum_{n=1}^{N} \sum_{t=1}^{T} C(n,t)Y(n,t)$$

s.t.

$$\sum_{t=1}^{T} Y(n,t) = 1 \qquad \text{for all } n \text{ in } N$$

$$\sum_{n=1}^{N} \sum_{m=1}^{N} \sum_{t=1}^{Te} Y(n,t)*Y(m,t)*X(n,m) = 0$$

$$\sum_{n=1}^{N} Y(n,t) <= X(t) \qquad \text{forall } t \text{ in } T$$

According to first constraint, each exam must be scheduled in one period. Moreover, second constraint guarantees that nobody has two exam in the same time. Lastly, third equation satisfies the capacity constraint.

1.2. PURPOSE OF THE STUDY AND STUDY AREA

In this study, a new mathematical modelling is generated to reach optimum solution for exam timetabling. However, size of the problem is very important for a mathematical model, so two new heuristic methods are constructed. In these methods, benefits of both lecturers and students, room capacities and some special constraints are considered.

In the literature, many constraints can be seen. In this research, different constraints and criterias are used. Some students have two exams in the same time slot, in the consecutive periods, or in the same day, so these situations are problems for students. In the ideal case, nobody has two exams at the same time. Moreover, having exam in the consecutive periods or days are big problem for students, so exams should be scheduled in a wide interval.

Furthermore, exams of courses which are taken by many students should be scheduled in the first timeslots, because grades are announced up to any determined date. It means that exam date of crowded courses - taken by many students - should be planned in the early timeslots and small courses – taken by few students - should be planned in the late timeslots. In addition to these, room capacity is one of the most important restriction, because at the same time, a specific number of students can enter the exam. Last but not least, assigning of assistants is an other problem, because there must be an inviligator in the each room during exam. Our model struggles to solve these problems.

Universities face exam timetabling problems in every year, so mathematical model and heuristic methods which are generated in this study can be used for exam timetabling problems. Heuristic method were also applied for Fatih University dataset. Fatih University has approximately 10000 students, and they can have problems related with their exam schedule. Three big faculties, Faculty of Engineering, Faculty of Science and Arts, Faculty of Management, are considered.

CHAPTER 2

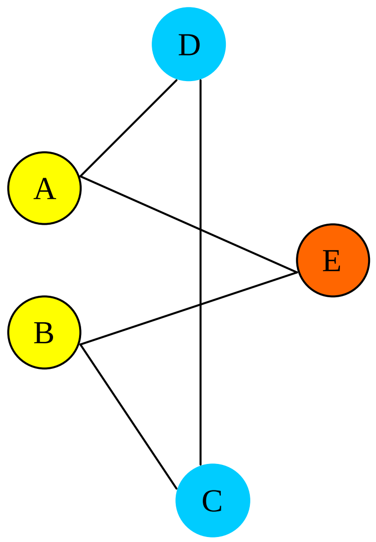
LITERATURE REVIEW

Timetabling problems can be seen in many different forms, such as; educational timetabling (Burke et al. 2004a) , nurse scheduling (Burke et. al. 2004b), sports timetabling (Easton et. al. 2004) and transportation timetabling (Kwan 2004). These problems are studied since 1960s in the area of Operational Research and Artificial Intelligence. Many researchs can be seen in the literature.

Among these problems, educational timetabling is one of the most popular subjects. Educational timetabling includes university course timetabling, exam timetabling and high school timetabling. Exam and course timetabling are closely related problems (Schaerf 1999). However there can be seen some differences (McCollum 2007). Exam timetabling can be expressed as assigning a set of exams into a limited number of timeslots and rooms subject to some constraints. There are lots of different constraints from institutions to institutions (Burke et al.1996a, Carter et al.1994)

Many techniques are used to solve exam timetabling problems, these are graph based sequential techniques, clustering methods, constraint based techniques, heuristic methods , multi-criteria techniques and decomposition techniques.

Welsh and Powell (1967) contributed graph-based technique to the literature. For exam timetabling problems, exams are shown by circles in a graph, and hard constraint is shown by edge between circles. In graph colouring problems, no adjacent circles have same colour then this procedure is applied for exam timetabling problems and exams are assigned to timeslots. For example, Abdullah (2006) gave an example of this procedure. In the below figure, there are 5 exams and some exams have common people and this is shown with edges. Number of colours implies the number of required minumum number of timeslots.



Exams:A,B,C,D,E

Edges (constraints):
(A,D), (A,E),
(B,E), (B,C), (D,E)

Colours (timeslots):
Yellow
Blue
Red

Figure 2.1: Graph colouring method (Abdullah, 2006)

Graph colouring based huristics are constructive methods. Exams are assigned to timeslots one by one. There are lots of ordering strategies for exam timetabling problem.

- Largest degree first: exams which have a large number of conflicts with other exams are scheduled early (Broder 1964). These type of exams cause lots of problems so these should be planed firstly (Burke et. al., 2002)

- Largest weighted degree: this method is like largest degree first method and based on calculation of number of student in each conflict (Burke et al., 2002; Carter et. al., 1996)

- Saturation degree: in each step of timetabling procedure, exam which has the smallest number of available time periods is scheduled firstly.

- Colour degree: exam which has the largest number of conflict with the scheduled exams is planned firstly (Burke et. al., 2002).

- Random ordering: In this method, exams are scheduled randomly (Qu et. al 2009).

Mathematical programming helps us to find an optimal solution in any optimisation problem. Therefore, mathematical programming is used to solve exam timetabling problems. In the literature, some models can be observed about this problem. Models are constructed and exams are considered as variables with finite domains (Qu et. al., 2009). Exams are assigned to domains which represent rooms and timeslots according to objective function. First researchs are related with finding the optimal solution (Qu et. al 2009). Brailsford et. al.(1999) also showed that this problem can be seen like an optimisation problem.

Mathematical programming models find an optimal solution, however there is a big problem: time. When the size of problem is large, models can not reach optimal solution in the short time. Sometimes, run time of problems is several months or years. So different heuristic methods are applied to these problems (Carter and Laporte 1996; White 2000).

David (1998) constructed a model for exam timetabling problem in a French school, its name is the Ecole des Mines de Nantes. Time is very big problem to solve this so some heuristic methods were applied to partial solution and then complete solution was obtained.

Reis and Oliveira (1999) constructed an examination timetabling system using with ECLiPSe –an open source software- (Ajili and Wallace, 2003). They show the efficiency of their model with random data and real data from University of Fernando Pessoa in Porto. Merlot et. al. (2003) using OPL, an optimisation programming language, found an initial solution and then applied simulated annealing and hill climbing methods. Duong and Lam (2004) used a new constraint programming and they found an initial solution. After this, they used simulated annealing strategy to improve the quality of timetabling.

Le Huede et al. (2006) integrated constraint programming and multi-criteria optimisation. It can be easily seen that after constraint programmimg, heuristic methods are frequently used. Moreover, there is no paper except Merlot et. al. (2003) that contains the comparasions the constraint programming and heuristic methods for the same problem (Qu et.al., 2009).

In addition to these, McCollum et. al. (2008) developed a new mathematical model for exam timetabling. They consider the welfare of students, lecturers and administrators. They defined different cost functions according to them. Also, Sevkli et. al. (2008) generated a mixed-integer mathematical model for exam timetabling. They defined a cost function over the common matrix and this research is developed on this mathematical model. Lastly, MirHassani (2006) showed a new mathematical model which a large number of data can be solved with the help of this model.

Tabu Search is one of the most popular meta-heuristic methods which is widely used in the literature. Di Gaspero and Schaerf (2001) used Tabu Search, then Di Gaspero (2002) again applied with some changing and improved their first solution. White and Xie (2001) constructed a four-stage Tabu Search which is called OTTABU for university of Ottowa. Moreover White et. al. (2004), Paquete and Stützle (2002) developed some models with using Tabu Search. In these research, tabu lists change from paper to paper.

Simulated annealing has been generated from natural annealing process. (Aarts and Korst, 1989). Thompson and Downsland (1998) used this meta-heuristic within two stage, in the first stage they get e feasible solution, and in the second stage they improve the satisfaction of soft constraints. Bullnheimer (1998) discussed that how Quadratic Assignment Problem is converted to small scale timetabling problem and Simulated Annealing was used and not only exams but also timeslots were changed. Furthermore, Merlot (2003) got an inital solutial with Costraint Programming and then improved the solution with using simulated annealing and hill climbing. Duong and Lam (2004) had an inital solution again with Constraint Programming and they improved the quality of timetable with Simulated Annealing for exam timetabling problem for HMCH University of Technology. Moreover Burke (2004c) studied with heuristic which is variant of Simulated Aneealing called The Great Deluge algorithm (Dueck 1993).

In the problems, some researhers have used different techniques to escape from local optima. Therefore different techniques are used such as; Kempe chain neighbourhood, Variable Neighbourhood Seearch. First method, Kempe chain neighbourhood was studied by Casey and Thompson (2003); Cote et. al. (2005) and Merlot et. al.(2003). The main idea of this method is that chains of conflicting exams are swapped between timeslots (Qu et. al., 2009). Abdullah et. al.(2007) constructed a model with using large neighbourhood search. Furhermore, Variable Neighbourhooh Search has been studied by Hansen and Mladenovic (2001), Mladenovic and Hansen (1997). Burke et. al. (2006) also used Variable Neighbourhood Search and reached a good solution.

Genetic Algorithm is one of the most widely used algorithm in the literature in exam timetabling problems. Corne et. al. (1994) discussed the Genetic Algorithm which is used for exam timetabling in the past. Ross et. al. (1996), Ross et. al. (2003) studied about using Genetic Algorithms in academic timetabling problems. Terashima-Marin et. al. (1999) and Erben (2001) also studied with Genetic Algorithms. Shebani (2002) firstly constructed a mathematical model and then applied Genetic Algorithm to reach a solution. Wong et al. (2002) discussed the implemantation of Genetic Algorithm for the data of Ecole de Technologie Superieure. Cote et. al. (2005) considered two objective functions while using evalutionary algorithm; one of them is the minimising the length of timetable, an other is minimizing the conflinting of exams. In addition to these, Ulker et. al. (2007) developed a Genetic Algorithm for academic timetabling problems.

Memetic Algorithms (Moscato and Norman, 1992) are a further version of Genetic Algorithm. Its main idea is individuals in a population are improved during their life-time. Generally local search is used after Genetic Algorithms in Memetic Algorithms. Burke and Landa Silva (2004d) discussed the using of Memetic Algorithms for scheduling and timetabling problems. Krasnogor and Smith (2005), Osman and Laporte (1996) mentioned about the Memetic Algorithms to solve combinatorial optimisation problems. Burke (1996b) used Memetic Algorithm for exam timetabling problem. Also he implemented hill climbing and the quality of solution is high but it requires a large amount of time. Moreover, in his paper, he had given the data of Nottingham University and then many researchers have studied this problem. Burke and Newall (1999) presented a new heuristic model with using decomposition method and Memetic Algorithm.

Ant Algorithms (Dorigo and Blum, 2005; Merkle and Middendorf, 2005) is one the most popular population based meta-heuristic methods. Costa and Hertz (1997) did one of the first reseachs about this topic. Naji Azimi (2004) used Ant Colony method in the first stage ant then he used local search algorithm to improve his solution. He also compared the Ant Colony algorithm with Simulated Annealing, Tabu Search and

Genetic Algorithm. Furthermore Naji Azimi (2005) studied the extension of Tabu Search and Ant Colony algorithms. Dowsland and Thompson (2005) generated an Ant Algorithm based on graph colouring. Eley (2007) compared two Ant Algorithms; Max-Min ant system (Socha et. al. 2003) and ANTCOL algorithm (Costa and Hertz, 1997). When both algorithm is used with hill climbing, ANTCOL algorithm is better than other.

Weight of constraints can be changed from person to person. In this approach, some constraints have different weights. Landa Silva et. al. (2004) wrote about the multi-criteria approach for scheduling and timetabling applications. Colijin and Layfield (1995) used a multi-stage approach for exam timetabling in the University of Calgary. Burke et. al. (2001) developed a multi-criteria approach which consists two-stage and nine criterias in exam timetabling problems. In addition to these, Le Huede et. al. (2006) used a multi-criteria optimisation and constraint programming for exam timetabling problems.

There are many meta-heruristic algorithms to solve exam timetabling problems, however some heuristic methods work well for some problems, but not for other problems. So researchers find different and new algorithms for different problems. Hyper-heuristics are inspired from this view and it means heuristics of choosing heuristics (Qu et. al. 2009). Aim of hyper-heuristics is to give general ways to solve timetabling problems. Ross et.al. (1998) suggested Genetic Algorithm, Ahmadi et. al. (2003) generated a Variable Neighbourhood Search, Ross et. al. (2004) developed a Genetic Algorithm again, Kendall and Hussin (2005a, 2005b) searched the usability of Tabu Searc as hyper–heuristic, Burke et. al. (2007) investigated Tabu Search Algorithm again, Bilgin et. al (2007) worked with 7 heuristic methods and Ersoy et. al. (2007) studied three hill climbers within Memetic Algorithm.

The main idea of the decomposition is that large problems are broken into small subproblems which can be solved by relatively simple methods (Carter, 1983). Altough it has some advantages (Burke and Newall, 1999). However this technique have not been used frequently because of some problems (Qu et. al., 2009). Firstly, early assignments can cause a later feasibility, secondly, globally high quality solutions can be missed (Qu et. al., 2009). Carter and Laporte (1996) did early research about decomposition approaches and exams are divided by two groups; conflict-free and lowconflict. Carter et. al (1996), Carter and Johnson (2001) also used docomposition techniques in their researchs.

With using this method, exams are grouped, then these groups which satisfy hard constraints are scheduled in the timeslots. White and Chan (1979) did early research using this technique. Because of exams are grouped at the beginning of algorithm may result poor quality for timetabling (Burke et. al., 2002). Arani and Lofti (1989), Lofti and Cerveny (1991) presented a three phases method. In the first phase, exams are grouped according to minimising the students who have more than one exam in the same time slot. These clusters are assigned to timeslots while minimizing the number of students who have two or more exams in a day. In the third stage, exam days and clusters are planned to minimize the number of students who have consecutive exams.

Burke and Newall (1999) also studied with decomposition techniques for exam timetabling problems. They used backtracking and look-ahead techniques to avoid early assignments which cause later infeasibilities. In addition to these, they applied Memetic Algorithm and solution of this researh is high quality. Furthermore Lin (2002) generate a new model and problem is decomposed into sub-problems. Qu and Burke (2007) applied decomposition techniques into exam timetabling problem of University of Toronto data. Solution of this research is again high quality when comparing the solutions of other methods.

CHAPTER 3

A PROPOSED MATHEMATICAL MODEL

In the literature, there are some mathematical models about exam timetabling problems. In this research, different mathematical models are considered, and especially model of Sevkli et. al.(2008) is studied and it is developed. Sevkli et al.(2008) presented the model for Fatih University Vocational School as follows;

Let n represents the exam day (n = 1,2,3,…,13).

Let T be the number of timeslots; t =1,2,3,…,T.

Let E be the number of exams to be scheduled; e=1,2,3,…,E.

Let S be the number of students; s = 1,2,3,…,S.

Let C be the number of classrooms; c=1,2,3,…,C.

Let Pc be the capacity of classroom c ; c=1,2,3,…,C.

Let W(e) be the number of students taking an exam e ; e=1,2,3,…,E.

Let COMMON (e,d) be a matrix which gives the number of students who take both courses.

There are four timeslots in each day.

TAKEN (s,e) = { 1 if student s takes course e

0 Else }

X (e,t) = { 1 if exam e is organized at time t

0 Else }

$$\text{Minimize Total Conflict} = \sum_{d=1}^{E} \sum_{e=1}^{E} \sum_{t=1}^{T} X(e,t)*X(d,t)*COMMON(e,d) \qquad (1)$$

s. t.

$$\sum_{t=1}^{T} X(e,t) = 1 \quad \text{for all e in E} \qquad (2)$$

$$\sum_{e=1}^{E} W(e)*X(e,t) <= \sum_{c=1}^{C} Pc \quad \text{for all t} \qquad (3)$$

$$\sum_{t=1}^{2}\sum_{e=1}^{E} TAKEN(s,e) * X(e,t) <= 1 \text{ for all s and n} \qquad (4)$$

$$\sum_{t=2}^{3}\sum_{e=1}^{E} TAKEN(s,e) * X(e,t) <= 1 \text{ for all s and n} \qquad (5)$$

$$\sum_{t=3}^{4}\sum_{e=1}^{E} TAKEN(s,e) * X(e,t) <= 1 \text{ for all s and n} \qquad (6)$$

The objective function gives the total conflict. 2nd equation indicates that each exam must be scheduled only at a specific time.3rd constraint implies that number of students who enter the exam at time t can not exceed the total capacity. The last three equation satisfies that a student at most two exams in a day and these are not in consecutive timeslots.

3.1 PROPOSED MATHEMATICAL MODEL FOR EXAM TIMETABLING

Mathematical Programming (MP) is widely used method to optimise any system. Especially after World War 2, Operational Research tecniques have been very popular in the academic and business enviroments. In this study, firstly a cost function is defined and then this cost function is struggled to minimize according to some constraints. The model is given below.

E: set of exams

T: set of times

R: set of rooms

K(r): capacity of room r

Y(e): number of people who take the course e

NA: number of available assistant

common(e,d): how many people take both e and d courses

W: set of days which are in weekend

$$plan(e,t) = \begin{cases} 1 & \text{if exam e is organized at time t} \\ 0 & \text{Else} \end{cases}$$

$$plan(e,r,t) = \begin{cases} 1 & \text{if exam e is organized on room r at time t} \\ 0 & \text{Else} \end{cases}$$

$$y(e,d,t) = \begin{cases} 1 & \text{if exam e and exam d are organized at time t} \\ 0 & \text{Else} \end{cases}$$

Minimise: C0 + C1 + C2 + C3

s.t.

$$\sum_{t=1}^{T} plan(e,t) = 1 \text{ for all e in E} \quad (1)$$

$$\sum_{e=1}^{N} plan(e,r,t) <= 1 \quad \text{for all r in R, t in T} \quad (2)$$

$$\sum_{r=1}^{R} \sum_{t=1}^{T} K(r)*plan(e,r,t) >= Y(e) \quad \text{for all e in E} \quad (3)$$

$$\sum_{r=1}^{R} plan(e,r,t) <= M*plan(e,t) \quad \text{for all e in E, t in T} \qquad (4)$$

$$\sum_{e=1}^{E} \sum_{r=1}^{R} plan(e,r,t) <= NA \quad \text{for all t in T} \qquad (5)$$

$$y(e,d,t) = plan(e.t)*plan(d,t) \quad \text{for all e,d in E, t in T} \qquad (6)$$

$$C0=\sum_{e=1}^{N} \sum_{d=1}^{N} \sum_{t=1}^{T} y(e,d,t)*common(e,d) \qquad (7)$$

$$C1=\sum_{e=1}^{N} \sum_{d=1}^{N} \sum_{t1 € T}^{T} \sum_{t2 € T}^{T} y1(e,t1,d,t2)*common(e,d) \qquad (8)$$

and e,d in E, t1- t2 = 1 or t2 – t1 = 1

$$C2=\sum_{t=1}^{T} \sum_{e=1}^{E} t*plan(e,t)*C(e) \qquad (9)$$

$$C3=\sum_{e=1}^{E} \sum_{t=1}^{W} plan(e,t) \quad \text{t in W} \qquad (10)$$

Equation 6 is nonlinear. So it should be converted to a linear situation. Following equations satisfies the linearity below (Sevkli et. al., 2008);

Linearization of equation 6 ;

plan(e,t) + plan(d,t) - y(e,d,t) <= 1 for all e, d in E

plan(e.t) + plan(d,t) - 2*y(e,d,,t) >= 0 for all e, d in E

Equation (1) explain that, each exam must be organized only one time. According to equation (2); at most one exam is planned on a room in the same time. Futhermore equation (3) says that summation of capacities of rooms on which an exam is organized must be bigger than the number of student who take the course of this exam. In addition to these, equation (4) constructs the relationship between plan(e,r,t) and plan(e,t), meanwhile M is a big number. Equation (5) explains the constraint of assistant as inviligator. In any time, number of rooms are used for exams must be smaller than available assistants. In the equation (6), y(e,d,t) is defined.

Moreover, different four cost functions are expressed, first of all is C0. It is the number of people who have more than one exam in the given time, t. C1 is the number of people who have exam in consecutively periods. C2 is the cost function related with size of exams. Lastly, C3 defines the number of exam is organized in the weekend days.

Objective function includes C0, C1, C2 and C3. However, model does not work efficiently for large size problems. For example;

- for 10 exams, 5 times, 5 assistants and 10 rooms, run time takes 88.6 seconds,

- for 30 exams, 10 times, 5 assistants and 20 rooms, run time takes more than 400 seconds,

- for 82 exams, 24 times, 40 assistants and 18 room, run time takes more than 18 minutes.

This model is not practical in the application, so in this research; C1, C2 and C3 are ignored.

For the new proposing approach, run time is smaller than old one. For example;

- for 10 exams, 5 times, 5 assistants and 10 rooms, run time takes 5 seconds,

- for 82 exams, 24 times, 40 assistants and 18 room, run time takes 513 seconds.

Furthermore, for 214 exams, 24 times, 40 assistants and 35 rooms, sofware can not find any solution, becuse memory of computer is not enough to solve this problem.

3.2. HEURISTIC MODEL

Heuristic and meta-heuristic methods are very popular to solve combinatorial optimisation problems. For big size problems, classical optimisation methods such as; simplex method, branch and bound allgorithm etc. can not reach the optimum solution in the short time, so heuristic methods are widely used in the literature. The proposed mathematical model can not solve the big size problems, so a heuristic methods based on this mathematical model are generated with using decomposition and classification methods. In the following parts, the heuristic is expressed in detail.

There are many exams to assign to timeslots and rooms. Firstly, exams are grouped and different sets which consist of exams are generated. These are exam sets. Moreover, there are also different sets which consist of rooms and these are named with place sets. In addition to these, there are a defined number of sessions in a day. Beginning from the any exam set and the any session of any place set, exams are scheduled with using mathematical model. Then, for an exam set except old one is scheduled on any session of place set which is different from previous one. When this exam set is planned, exams which were scheduled in the past are considered. Likewise, each set is scheduled with considering all of the scheduled exams previously. Common matrix is developed between set which will be scheduled and all sets which were scheduled. In addition to these, all scheduled exams in the past are new constraints for future mathematical models. At the end of processes, pairwise changing may be applied to reduce cost function of all model, because each solution of subproblems may be bigger than 0. It means that some people have more than one exam in the same time. Doing pairwise changing between problems may reduce overall cost function.

For instance; suppose that there are two place sets, six different sets, three sessions in a day and thirteen days.

In the Figure 3.1, this process is shown, there are two place sets; set A and set B, three sessions in each day and thirteen days.

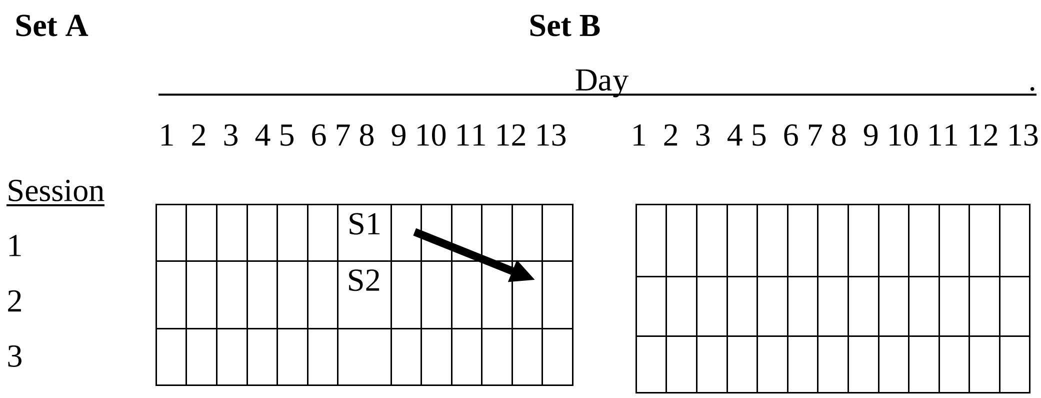


Figure 3.1: Planning of set S2

S1 set is scheduled with using mathematical model. After that, S2 set is planned, and common matrix is generated between S1 and S1 + S2. The matrix is shown in Figure 3.2.

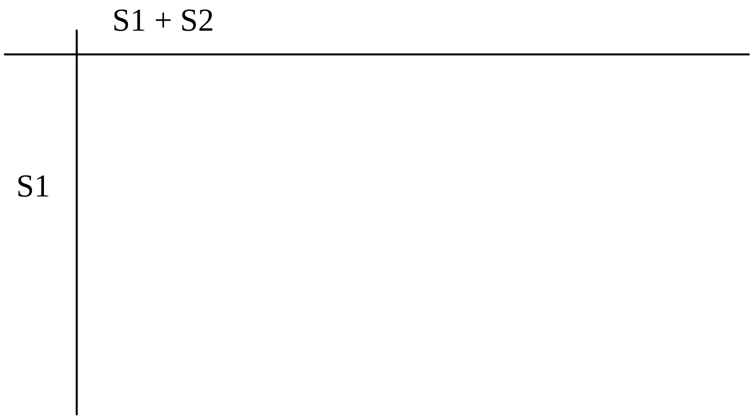


Figure 3.2: First matrix

Then set S3 is scheduled with using the outcomes of S1 and S2. This process is shown in the Figure 3.3. In the following figure, two place sets, three session in a day and twelve days are presented.

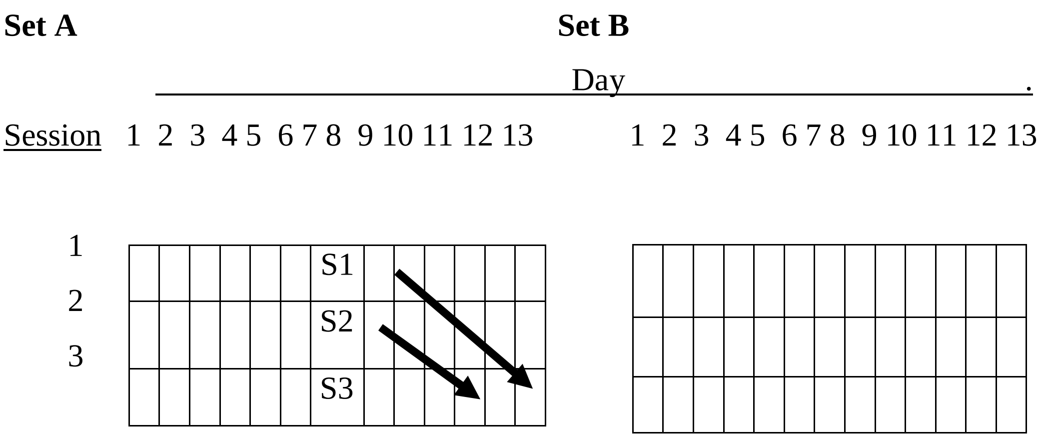


Figure 3.3: Planning of set S3

The common matrix is constructed between S3 and S1 + S2 + S3 (Figure 3.4)

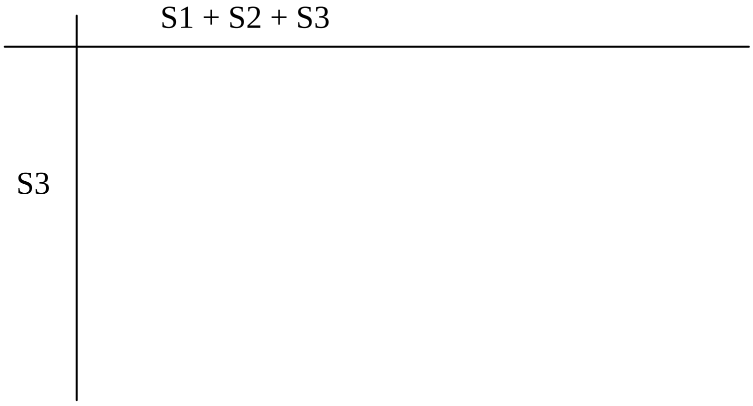


Figure 3.4: Second matrix

In addition, a set S4 is also planned according to scheduled sets, S1, S2 and S3 (Figure 3.5). Likewise, a common matrix is constructed between S4 and the other three sets, S1, S2 and S3 (Figure 3.6).

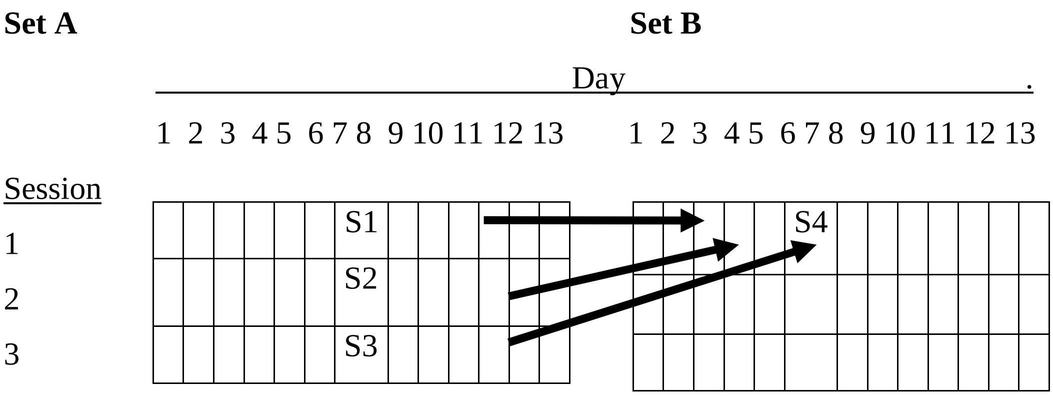


Figure 3.5 Planning of S4 set

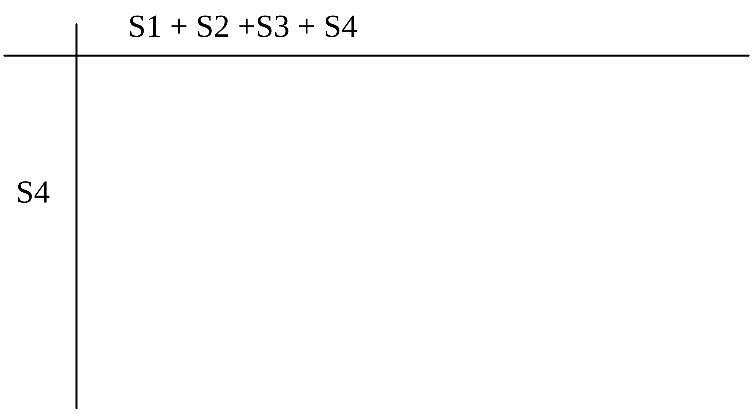


Figure 3.6: Third matrix

Like that all sets are assigned according to this heuristic. Lastly, pairwise changing may be done with subproblems.

CHAPTER 4

FATIH UNIVERSITY APPLICATION

In this chapter, the heuristic method is applied to Fatih University for spring term of 2008-2009 education year. Final exam days are organized by school after the end of each semester. Many students have same type of problems that they have more than one exam in the same time slot. This situation crises for both lecturers and students. We are planning to organize exams with considering this problem which is faced frequently by students in each year. Data are generated from Fatih University database. There are more than 1000 courses in three main faculties. These are Faculty of Engineering, Faculty of Science and Arts and Faculty of Management. Furthermore, there are approximately 100 that rooms can be used for exams.Using second heuristic methods, exams have been planned and nobody has two exams in the same time. In addition, number of students who have more than one exam in the same should be minimized. Details of application process are as follows:

As mentioned above, Faculty of Engineering, Faculty of Science and Arts and Faculty of Management are considered for case study. There are approximately 1000 different courses, and approximately 100 rooms for exams. Problems which are observed in the Fatih University are listed below.

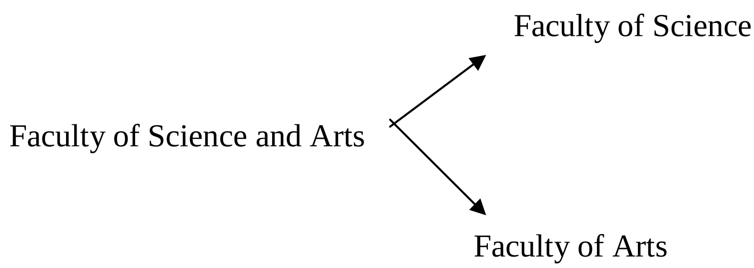
- Turkish and English sections of same course may be in different timeslots. Fatih University has both Turkish and English programmes for many departments. It means that teaching language in one department is Turkish and the other is English. Generally, same professor gives English and Turkish version of same course. For instance, IE_104 is an English section and ENM_104 is a Turkish section of Engineering Design course and these are taught by same lecturer. Generally, professors want that both of courses should be planned in the same timeslot, because they do not want to prepare different questions for both Turkish and English sections, and do not want to spend extra times to organize exams.

- Secondly, sections of courses sometimes may be in different days. Usually, different sections of course are taught by a lecturer. For instance, three sections of IE_104; IE_104A, IE_104B and IE_104C are given by same person. Likewise above problem, professors want that exams of different sections of courses should be planned on the same timeslot, because if they are scheduled on different timeslots, professors prepare different questions for each section and they spend many times to organize exams.

- Thirdly, exams of any faculty are not planned on its building. Students want to enter exams in building of their faculties, because they may lose time to find their exam rooms on other buildings. For instance, IE_104 is a course of Faculty of Engineering. If it is planned on the building of Vocational School, students who enter the IE_104 exam may look for their rooms and spend times before exam.

- Sometimes, exams of different sections of any course are organized in different buildings. This is a big problem for lecturers to organize an exam properly, because they want to visit rooms during exam. Therefore, lecturers want that exams of different sections should be planned on the same building.

To solve some problems as mentioned above, Turkish, English and different sections of any course are considered like one course. For instance, IE_400 consists of IE400_A, IE400_B, ENM400_A and ENM400_B. When a course is assigned to any timeslot, exams of the Turkish, English and different sections are also planned on the same timeslot. Therefore, problem which is scheduling of different sections on different timeslots is prevented. In addition, problem size is very important criteria to solve mathematical model. Therefore, to decrease the problem size, Turkish, English and different sections of courses are combined. Furthermore, because Faculty of Science and Arts has many courses and its size is very big, it is also divided by two parts as Faculty of Science and Faculty of Arts



After this process, six sets are constructed .First set includes courses which are taken by only Engineering Faculty students.

Second set includes courses which are taken by only Science Faculty students.

Third set includes courses which are taken by only Arts Faculty students.

Fourth set includes courses which are taken by only Management Faculty students.

Fifth set includes courses which are taken by students of any two faculties.

Sixth courses includes courses which are taken by students of any three faculties or students of all faculties.

In addition to these, 13 days are considered to plan all schedule. These days start from to Monday and ends to Saturday of the second week.

Courses belong to first, second, third, fourth, fifth and sixth set are shown on the following tables.

Courses of first set are;

CENG_102, CENG_103, CENG_252, CENG_304, CENG_305, CENG_362, CENG_421, CENG_482, CENG_490, CENG_491, CENG_492, EEE_237, EEE_286, ENVE_212, EEE_292, EEE_322, EEE_338, EEE_362, EEE_373, EEE_412, EEE_422, EEE_435, EEE_445, EEE_463, EEE_484, ENVE_407, ENVE_204, IE_470, ENVE_432, ENVE_498, ENVE_540, ENVE_545, GBE_102, GBE_104, GBE_208, IE_216, IE_226, IE_320, IE_345, IE_473, IE_502 , IE_455, CENG_576, CENG_581, CENG_583, CENG_593, ÇEM_206, ÇEM_402 , ÇEM_404, EEE_122, EEE_555,

EEE_585, EEM_452, ENGR_100, ENVE_200, IE_435, CENG_497, CENG_498, CENG_565, ENVE_304, ENVE_312, EEE_498, EEE_505, EEE_533, IE_541, IE_559.

Courses of second set are;

BIOL_102, BIOL_106, BIOL_202, BIOL_204, BIOL_252, BIOL_302, BIOL_304, BIOL_306, BIOL_308, BIOL_354, BIOL_356, BIOL_358, BIOL_401, BIOL_418, BIOL_504, BIOL_516, BIOL_517, BIOL_518, BİYO_352, BİYO_419, BİYO_520, BİYO_525, BİYO_527, CHEM_404, CHEM_414, CHEM_531, CHEM_533, CHEM_554, CHEM_599, FİZ_206, FİZ_208, FİZ_210, FİZ_252, FİZ_304, FİZ_306, FİZ_324, FİZ_352, FİZ_404, FİZ_406, FİZ_422, FİZ_426, KİM_105, KİM_106, KİM_202, KİM_206, KİM_208, KİM_252, KİM_256, KİM_258, KİM_306, KİM_310, KİM_312, KİM_352, KİM_356, KİM_402, KİM_407, KİM_410, KİM_555, KİM_558, KİM_574, MATE_112, MATE_115, MATE_206, MATE_214, MATE_232, MATE_316, MATE_320, MATE_350, MATE_360, MATE_408, MATE_414, MATE_428, MATE_492, MATH_430, MATH_508, MATH_530, MATH_532, MATH_540, MATH_543, MATH_560, MATH_568, MATH_641, PHYS_502, PHYS_510, PHYS_514, PHYS_520.

Courses of third set are;

ACL_112, ACL_114, ACL_122, ACL_132, ACL_134, ACL_198, ACL_212, ACL_224, ACL_232, ACL_236, ACL_240, ACL_290, ACL_332, ACL_362, ACL_362, ACL_366, ACL_372, ACL_418, ACL_432, ACL_434, ACL_436, ACL_438, ACL_492, CLL_102, CLL_104, CLL_106, CLL_108, CLL_202, CLL_204, CLL_206, CLL_208, CLL_214, CLL_302, TLE_426, CLL_304, CLL_306, CLL_308, CLL_326, CLL_402, CLL_404, CLL_406, CLL_408, CLL_432, FEL_101, GEO_104, GEO_122, GEO_152, GEO_182, GEO_202, GEO_222, GEO_252, GEO_282, GEO_302, GEO_312, GEO_322, GEO_352, GEO_402, GEO_496, GEO_524, GEO_567, GEO_585, HIST_102, HIST_104, HIST_124, HIST_204, HIST_210, HIST_218, TLE_427, HIST_236, HIST_308, HIST_404, HIST_410, HIST_504, HIST_520, PHIL_106, PHIL_108, PHIL_110, PHIL_118, PHIL_122, PHIL_208, PHIL_216, PHIL_302, PHIL_305, PHIL_312, PHIL_412, PHIL_492, PHIL_504, PHIL_510, PHIL_515, PSY_102, PSY_104, PSY_210, PSY_236, PSY_302, PSY_308, PSY_314, PSY_322, PSY_330, PSY_336, PSY_360, PSY_362, TLE_422, PSY_436,

PSY_506, PSY_508, PSY_539, RLL_110, RLL_114, RLL_116, RLL_216, RLL_222, RLL_224, RLL_312, RLL_314, RLL_410, RLL_412 RLL_436, RLL_502, RLL_504, RLL_508, SLL_102, SLL_104, SLL_106, SLL_108, SLL_112, SLL_202, SLL_204, SLL_206, SLL_214, SLL_304, SLL_308, SLL_312, SOC_512, TDE_204, TDE_206, TLE_419, TDE_208, TDE_304, TDE_306, TDE_308, TDE_326, TDE_336, TDE_338, TDE_406, TDE_408, TDE_422, TDE_430, TDE_436, TDE_442, TDE_504, TDE_511, TDE_516, TDE_526, TDE_532, TDE_533, TDE_536, TLE_102, TLE_104, TLE_106, TLE_108, TLE_202, TLE_206, TLE_302, TLE_304, TLE_306, TLE_402, TLE_410.

Courses of fourth set are;

ECON_232, ECON_304, ECON_312, ECON_388, ECON_402, PUB_510, ECON_406, ECON_414, ECON_502, ECON_532, ECON_534, PUB_510, EKON_322, INT_402, INT_504, INT_538, INT_544, MAN_598, INT_550, KAM_362, KAM_466, MAN_417, MAN_436, MAN_550, MAN_446, MAN_483, MAN_524.

Courses of fifth set are;

BİLM_110, BİLM_204, BİLM_310, BİYO_104, BİYO_206, CENG_104, CHEM 207, EEE_202, ENVE_210, ENVE_344, ENVE_408, ENVE_415, ENVE_417, FİZ 423, FRE302, GBE_106, GBE_204, GBE_206, PUB_10, ULUS_330, GBE_306, IE_104, IE_476, JAP_202, KİM_108, KİM_442, MATH_110, MATH_234, MATH_329, MATH_330, MATH_346, MATH_348, ACL_212, HIST_204, PHIL_106, PHIL_108, PHIL_110, PUB_126, PUB_466, PHIL_118, PHIL_122, PHIL 208, PSY_362, RLL_436, HIST_108, PHIL_242, PHIL_404, PSY_201, PSY_237, PSY_330, SOS_202, SOS_210, SOS_302, TDE_106, TDE_32, INT 202, INT 318, PUB_444, MATH_222, KAM_314, KİM_108, KİM_442, MAN_436, MAN_446, ECON_204, ECON_238, ECON_302, EKON_436, HIST_226, HIST_228, HIST_320, HIST_336, HIST_428, HIST_442, INT_302, INT_426, KAM_106, PUB_454, İŞLE_452, PSY_233, RLL_112, RLL_118, RLL_302, RLL_306, RLL_414, SOC_210, SOC_524, TDE_432, BİYO_206, FRE_202, İŞLE_202, İŞLE_204, İŞLE_212, İŞLE_250, İŞLE_306.

Courses of sixth set are;

GEO_324, ARB_202, PHIL_230, PRS_202, TDE_210, İŞLE_304, İŞLE_310, İŞLE_416, İŞLE_425, İŞLE_432, İŞLE_102, İŞLE_201, KAM_274, MATE_106, MATH_114, MATH 230, PUB_362, PUB_488, ECON_102, ECON_254, ECON_388, EKON_332, FEL_101, GEO_472, HIST_106, HIST_388, INT_212, INT_244, INT_252, INT _452, KAM_314, RLL_218, GER_202, PUB_212, PUB_284, PUB_368, RUS_202, SOC_314, SOC_422, SOS_281, SOS_286, TLE_110, CHN_202, ECON_386, FRE_202, INT_244, INT_356, INT_354, INT_462, İŞLE_224, İŞLE_232, İŞLE_314, İŞLE_422, İŞLE_434, İŞLE_108, KAM_250, PSİ_122, PSİ_122, PUB_216, PUB_308, SOC_108, SOC_320, SOC_102, SOS_340, SOS_383, SOS_282, SOS_385.

Firstly, there are two obligatory courses for all students in the university. These courses are APHR and TURK courses, and because these courses are taken by many students, mathematical model can not efficiently solve problem which includes these courses. So they are planned on Sunday of the first week. Generally, exam of TURK course takes 30 minutes, and this course is very easy. Therefore, there is no big problem for students that being organized of TURK courses on the same day with APHR. When these exams are scheduled on Sunday, then 12 days are considered in the following processes. Moreover, number of available inviligators for each problem set are assumed as 40 in this research.

4.1. **PLANNING OF EXAMS**

First set is planned in the building of Engineering Faculty in the first session of the 12 days. There are 66 courses in this set and 18 available rooms in the building of Engineering Faculty (Figure 4.1**).** The mathematical model is coded in Xpress-MP. The code is given in Appendix A.

Session		Day												
		1	2	3	4	5	6	7	8	9	10	11	12	13
1								**X**						
2	S1							**X**						
3								**X**						
4								**X**						

Figure 4.1: Engineering Faculty Building

The result of this code generates no confliction. In other words, cost founction is zero.

For the second set, exams are scheduled in the first session of Science Faculty building with same code (Figure 4.2)

Session		Day												
		1	2	3	4	5	6	7	8	9	10	11	12	13
1								**X**						
2	S2							**X**						
3								**X**						
4								**X**						

Figure 4.2: Science Faculty Building

The result of this code generates no confliction, it meand that cost founction is zero.

Thirdly, since set S3 is very big, exams are organized on the building of the Arts Faculty Building in the first and second sessions (Figure 4.3)

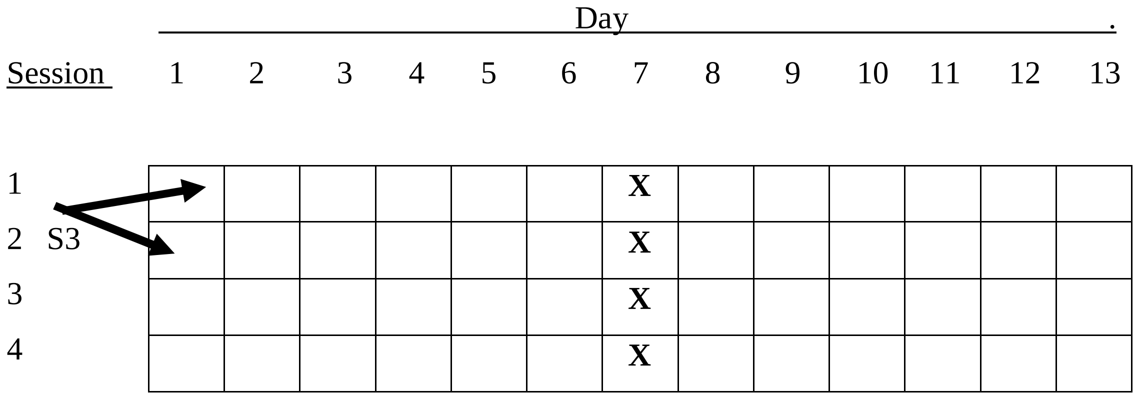


Figure 4.3: Arts Faculty Building

When the first part of S3 is planned, model reachs the point of solution that objective function equals to 3. Although software works minutes, better solution than 3 can not be found. Therefore programme is being stopped after reaching the cost function = 3 and solution at this point is accepted. Moreover, second part of S3 is organized in the second session of the Arts Faculty Building and cost function is equal to 0.

Set S4 is also planned on the first session of Management Faculty Building (Figure 4.4)

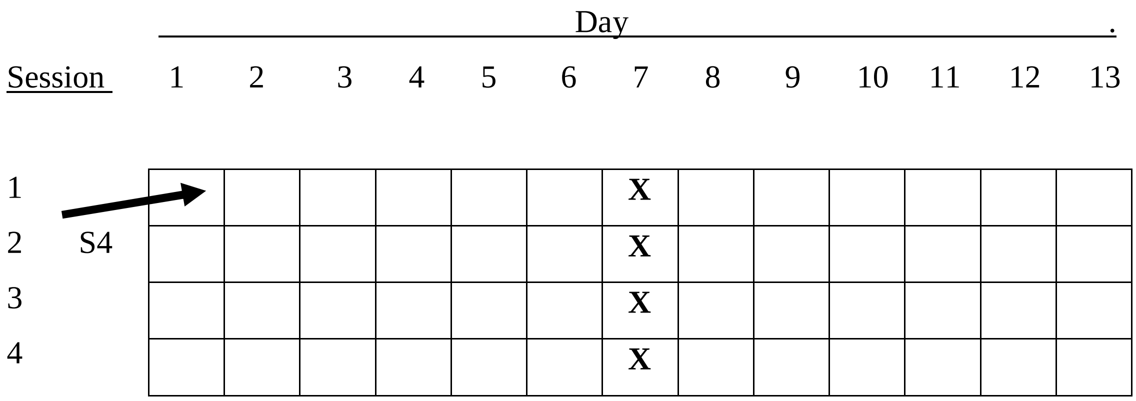


Figure 4.4: Management Faculty Building

Optimum value of this problem is zero, it means that there is no confliction. Benefit of this type of classification is that there is no relationship between S1, S2, S3 and S4 sets. It means that nobody takes courses from both of S1 and S2 sets at the same time. Likewise, this situation is valid under other conditions.

So now, there are two sets S5 and S6 which will be planned. S5 is organized in the third sesssion of the all building (Figure 4.5). Furthermore, common matrix is not only generated between S5 and S5 + S1 + S2 + S3 + S4. Therefore, new exams are scheduled considering with all scheduled exams. However, this common matrix is very large to solve. Decomposition is a good way to reach a solution, so S5 is divided by three parts which are S5a, S5b, S5c. First of all, S5a is planned on some rooms, then S5b is planned on defined rooms cosidering with all scheduled exams in the past, lastly S5c is organized on the some rooms of the third session considering with all scheduled courses previously. Since size of set S5 is very large and to decrease the number of student who have more than one exam in the same day, extra 12 rooms in the building of Vacotional School are used to schedule exams. 34, 30 and 32 rooms are taken for set S5a, S5b and S5c, respectively (Figure 4.5). Common matrixes are also constructed as in Figure 4.6

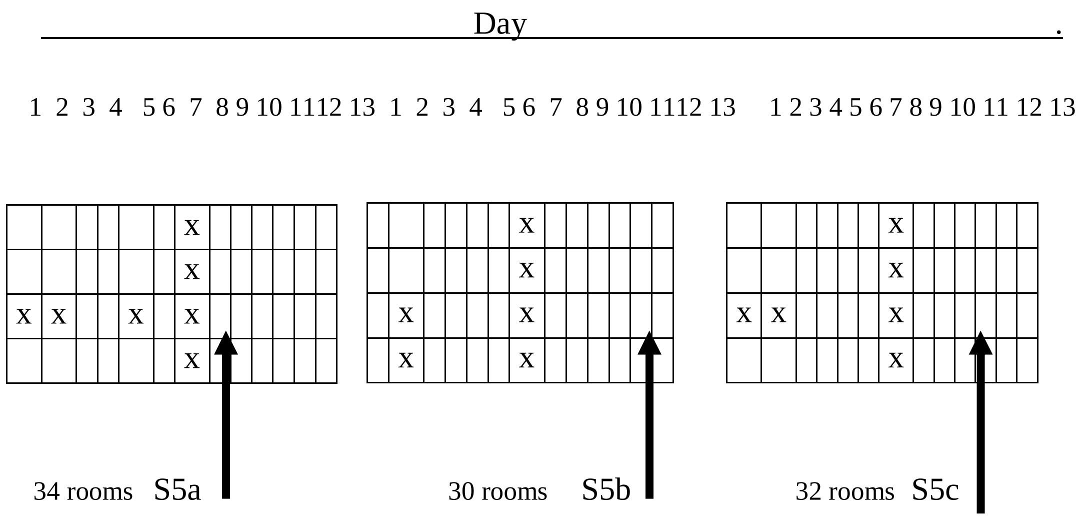


Figure 4.5: Planning of set S5

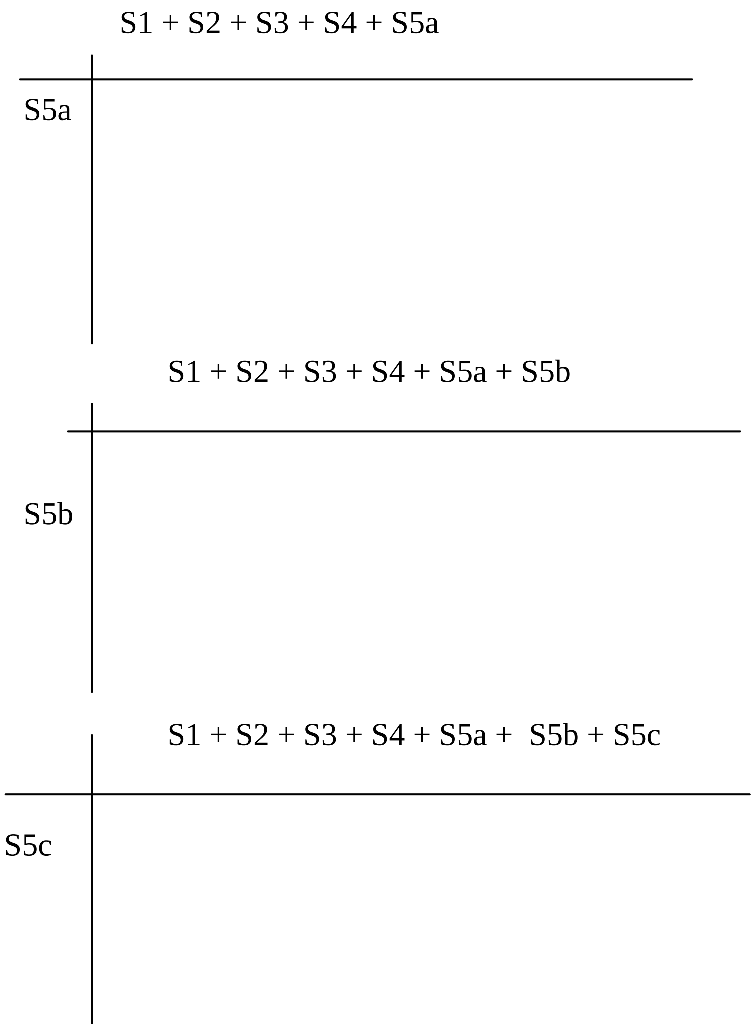


Figure 4.6: Matrices of S5 set

Three faculties have 84 rooms, however set S5 is very large, so additional 12 rooms in the building of Vocational School are used in the third session. When set S5a is scheduled on 32 rooms, 1st, 2nd and 5th times are not considered, because PHYS_104, MATH_114 and ING courses are scheduled on these times. Since PHYS_104 and MATH_114 courses are obligatory for all Engineering students and ING is obligatory for all students of Turkish departments, many students take these courses and sizes of these courses are very large. Therefore, they affect the efficiency of mathematical model and they are scheduled independently. Times are determined as follow; firstly PHYS_104 course is selected and interaction is checked for each day between PHYS_104 and courses which were scheduled previously in the first and second session. Lastly, one of the suitable days which give the few interaction is selected. In this study, 1st day is selected for PHYS_104. Likewise, 2nd and 5th days in the third session are selected for MATH_114 and ING courses, respectively. Furthermore, set S5b and S5c are scheduled on 64 rooms and 1st and 2nd days are not used, because MATH_104 and ING courses are planned in these times. Since MATH_104 course is obligatory for students of Management Faculty and sizes of both courses are very large, they are scheduled independently. Xpress-MP code of set S5a is shown in the Appendix B. Moreover, objective function equals to 17 for S5a, 1 for S5b and 0 for S5c. When analysing the cost values, it is observed that costs are sourced from having more than one exam in a day, not in the same time slot.

Lastly, there is a set S6 which was not scheduled before. Now, this set is considered and is scheduled in the fourth session of the all building (Figure 4.7)

		Day												
Session		1	2	3	4	5	6	7	8	9	10	11	12	13
1								**X**						
2								**X**						
3	S6							**X**						
4	→							**X**						

Figure 4.7: Planning of set S6

Moreover, there are 439 exams which are scheduled previously and our common matrix size is 65 x 504(65+ 439), however this matrix is so big for software and computer. Therefore, 147 exams which do not have any common person with set S6 are eliminated and size of matrix is transformed to 65 x 357. After running of software, objective function equals to 66 and only two of them is sourced from having two exams in the same timeslot.

Now, there is a problem that cost function of having in the same timeslot is 5 in the all model. These are;for S31 and t = 6, 1 person have two exams,

for S31 and t = 7, 1 person have two exams,

for S31 and t = 11, 1 person have two exams,

for set S6 and in t = 11, 2 people have two exams.

Lastly, pairwise changing is recommended in the heuristic. In the 6th, 8th and 12th day, totally five students have more than one exam in the same timeslot and these cost belong to S3a and S6 sets. Common matrixes are analysed and five problem exams are determined. At the end of analysis, while satisfying constraints, timeslots of these five exams are changed manually. Three exams are shifted to second session and two exams are shifted to first session, lastly nobody has more than one exam in the same timeslot (Figure 4.8).

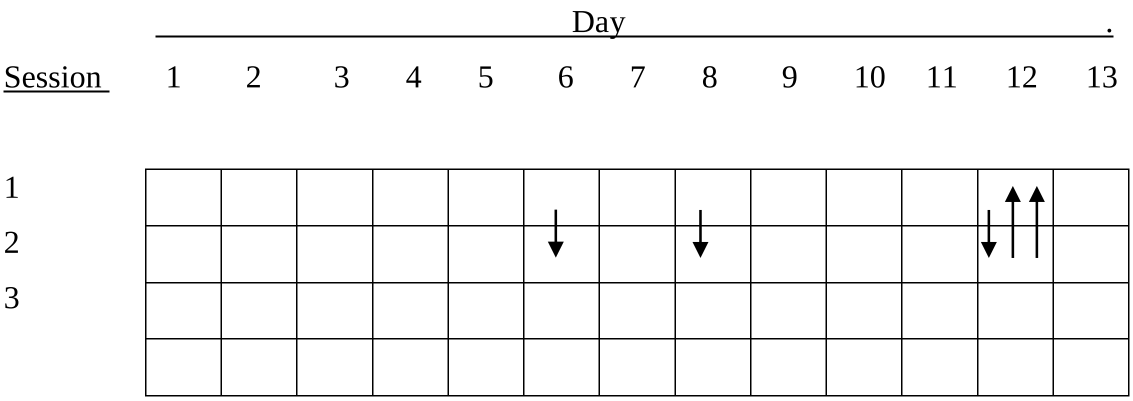


Figure 4.8: Chainging of time

All processes are summarised in Table 3.1.

Table 4.1: Report table

Problem Set	**Number of exams**	**Number of rooms**	**Cost function (same timeslot)**	**Cost function (same day)**	**Run Time**
S1	66	18	0	0	286 s.
S2	86	16	0	0	780 s.
S3a	84	17	3	0	357 s.
S3b	83	17	0	0	184.4 s
S4	24	33	0	0	1.6 s
S5a	32	34	0	17	18 s
S5b	32	30	0	1	3.2 s
S5c	32	32	0	0	3 s
S6	65	84	2	64	54.5 s
Total	504	-	5	81	1687,7 s

Problem set S1 contains 66 exams and 18 rooms are available. Solution of mathematical model is 0. It means that there is anybody who has more than one exam in the same timeslot and also in the same day. Furthermore, run time time takes 286 seconds. In addition, set S2 contains 86 different exams and 16 rooms are available. Solution of this problem is anybody has more than one exam in the same timeslot and in the same day. Run time takes 780 seconds and information about other sets are given above.

To sum up, 504 courses are scheduled with five costs in the same timeslot and after changing the timeslots of some exams, cost for the same timeslot decreases to 0.

CHAPTER 5

CONCLUSION

Exam timetabling is one the most popular scheduling problems in the literature. Many collouges and universities struggle to schedule exams with considering the desires of students and lecturers. Different techniques are used to solve timetabling problems. In the beginnings, mathematical model and clustering methods are used, and in the last years meta-heuristic methods are widely used to schedule exams.

In this research, a new mathematical model is developed. Objective function is number of students who have more than one exam in the same timeslot. Planning of an exam in a timeslot, number of inviligators and capacities of rooms are some constraints of this mathematical model. However, this model can not solve large size problems in the short time, so a new heuristic method embedded in the mathematical model are developed. Using this heuristic, number of students who have more than one exam in the same timeslot and in the same day is minimized. To decrease size of problem, the heuristic method is based on decomposition and classification. According to this heuristic, firstly, any part of problem is solved with mathematical model and then the other parts are done with using the former results.

In addition, the heuristic method is applied to dataset of Fatih University. In this research, there are 504 courses, 13 days and 4 sessions in each day. Courses are classfied and six sets are generated. Method of classification as follow; courses which are taken by only Engineering Faculty students, by only Science Faculty students, by only Arts Faculty students, by only Management Faculty students, by students of any two faculties, by students of any three faculties or students of all faculties.

At the end of analysis, anybody has more than one exam in the same timeslot. In this study, APHR and TURK are planned on Sunday of the first week, because these are obligatory for all students and exams of these courses need many rooms. Morover, they are very easy courses, so planning of two exams in the same day are not big problem for students. Except Sunday, approximately 80 students have two or more exams during the remaining 12 days.

In the future research, a model may be developed which gives the minimum number of timeslots while minimizing the number of students who have more than one exam in the same timeslot and day.

REFERENCES

Aarts, E. H. L., & Korst, J., *Simulated annealing and Boltzmann machines*, New York: Wiley, 1989.

Abdullah S., *Heuristic Approaches for Exam Timetabling Problems*, Phd Thesis, University of Nottingham, 2006.

Abdullah, S., Ahmadi, S., Burke, E. K., & Dror, M., *Investigating Ahuja–Orlins large neighbourhood search for examination timetabling*, OR Spectrum, 29(2), 351–372, 2007a.

Ahmadi, S., Barone, R., Cheng, P., Cowling, P., & McCollum, B. (2003), "Perturbation based variable neighbourhood search in heuristic space for examination timetabling problem", *In Proceedings of multidisciplinary international scheduling: theory and applications (MISTA 2003)*, pp. 155–171, Nottingham, 13–16 August, 2003.

Arani T. And Lothii V., *A Lagrangian relaxation approach to solve the second phase of the exam scheduling problem*, European Journal of Operational Research, 34, 372-383, 1989.

Ajili, F., & Wallace, M. W., "Hybrid problem solving in ECLiPSe", In M. Milano (Ed.), *Constraint and integer programming: toward a unified methodology*, pp. 169–201, Dordrecht: Kluwer Academic, 2003.

Broder, S., *Final examination scheduling*, Communications of the ACM, 7, 494–498, 1964.

Brailsford, S. C., Potts, C. N., & Smith, B. M., *Constraint satisfaction problems: Algorithms and applications*, European Journalof Operational Research, 119, 557–581, 1999.

Bullnheimer, B., “An examination scheduling model to maximise students study time”, In E. K. Burke & M. W. Carter (Eds.), *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II: selected papers from the 2nd international conference,* pp. 78–91, Berlin: Springer, 1998.

Burke, E. K., Elliman, D. G., Ford, P. H., & Weare, R. F., “Examination timetabling in British universities: a survey”, In E. K.Burke & P. Ross (Eds.), *Lecture notes in computer science:Vol. 1153, Practice and theory of automated timetabling I: selected papers from the 1st international conference*, pp. 76–90, Berlin: Springer, 1996a.

Burke, E. K., Newall, J. P., & Weare, R. F., “A memetic algo-rithm for university exam timetabling”, In E. K. Burke & P. Ross (Eds*.), Lecture notes in computer science: Vol. 1153. Practice and theory of automated timetabling I: selected papers from the 1st international conference*, pp. 241–250, Berlin: Springer, 1996b.

Burke, E. K., & Newall, J. P., *A multi-stage evolutionary algorithm for the timetable problem*, IEEE Transactions on Evolutionary Computation, 3(1), 63–74, 1999.

Burke, E. K., Bykov, Y., & Petrovic, S., “A multi-criteria approach to examination timetabling”, In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference*, pp. 118–131, Berlin: Springer, 2001.

Burke, E. K., & Petrovic, S., *Recent research directions in automated timetabling*. European Journal of Operational Research, 140(2), 266–280, 2002.

Burke, E. K., Kingston, J. H., & de Werra, D., “Applications to timetabling”, In J. Gross & J. Yellen (Eds.), *The handbook of graph theory*, pp. 445–474, London: Chapman Hall/CRC, 2004a.

Burke, E. K., De Causmaecker, P., Vanden Berghe, G., & Van Lan-deghem, H., *The state of the art of nurse rostering*, Journal of Scheduling, 7(6), 441–499., 2004b.

Burke, E. K., Bykov, Y., Newall, J. P., & Petrovic, S., *A time-predefined local search approach to exam timetabling problems*, IIE Transactions, 36(6), 509–528, 2004c.

Burke, E. K., & Landa Silva, J. D. “The design of memetic algorithms for scheduling and timetabling problems”, In W. E. Hart, N. Krasnogor, & J. E. Smith (Eds.), *Studies in fuzziness and soft computing: Vol. 166, Recent advances in memetic algorithms and related search technologies*, pp. 289–312, Berlin: Springer, 2004d.

Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., & Qu, R., *Hybrid variable neighbourhood approaches to university exam timetabling*, Technical Report NOTTCS-TR-2006-2, School of Computer Science, University of Nottingham, 2006.

Burke, E. K.,McCollum, B.,Meisels, A., Petrovic, S.,& Qu., *A graph based hyper-heuristic for examtimetabling problems*. European Journal of Operational Research, 176, 177–192, 2007.

Bilgin, B., Özcan, E., & Korkmaz, E. E., "An experimental study on hyper-heuristics and exam timetabling", In E. K. Burke & H. Rudova (Eds.), *Lecture notes in computer science: Vol. 3867, Practice and theory of automated timetabling VI: selected papers from the 6th international conference*, pp. 394–412, Berlin: Springer, 2007.

Carter, M. W., Laporte, G., & Chinneck, J. W., A *general examination scheduling system,* Interfaces, 24, 109–120, 1994.

Carter, M.W., & Laporte, G., "*Recent developments in practical examination timetabling".* In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science: Vol. 1153, Practice and theory of automated timetabling I: selected papers from the 1st international Conference*, pp. 3–21, Berlin: Springer, 1996.

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

Carter, M. W., & Johnson, D. G., Extended *clique initialisation in examination timetabling*, Journal of Operational Research Society, 52, 538–544, 2001.

Carter, M. W., *A decomposition algorithm for practical timetabling problems,* Technical Paper 83-06, Department of Industrial Engineering, University of Toronto, 1983.

Casey, S., & Thompson, J., "GRASPing the examination scheduling problem", In E. K. Burke & P. De Causmaecker Eds., *Lecture notes in computer science: Vol. 2740, Practice and theory of automated timetabling IV: selected papers from the 4th international conference,* pp. 232–244, Berlin: Springer, 2003.

Colijn, A. W., & Layfield, C., “Conflict reduction in examination schedules”, In E. K. Burke & P. Ross (Eds.), *Proceedings of the 1st international conference on the practice and theory of automated timetabling*, pp. 297–307, 30 August–1 September 1995, Edinburgh: Napier University, 1995a.

Corne, D., Ross, P., & Fang, H., “Evolutionary timetabling: Practice, prospects and work in progress”, In P. Prosser (Eds.), *Proceedings of UK planning and scheduling SIG workshop*, 1994.

Costa, D., & Hertz, A., *Ant can colour graphs*, Journal of Operational Research Society, 48, 295–305, 1997.

Côté, P.,Wong, T., & Sabouri, R., *“Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem”*, In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science: Vol. 3616. Practice and theory of automated timetabling V: selected papers from the 5th international conference*, pp. 151–168, Berlin: Springer, 2005.

David, P., “A constraint-based approach for examinationtimetabling using local repair techniques”, In E. K. Burke & M.W.Carter (Eds.), *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II: selected papers from the 2nd international conference*, pp. 169–186. Berlin: Springer, 1998.

De Werra, D. *An introduction to timetablin,*. European Journal of Operational Research, 19, 151–162, 1985.

Di Gaspero, L., & Schaerf, A., “Tabu search techniques for examination timetabling”, In E. K. Burke &W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference*, pp. 104–117, Berlin: Springer, 2001.

Di Gaspero, L., “Recolour, shake and kick: A recipe for the ex-amination timetabling proble”, In: E. K. Burke & P. De Caus-maecker (Eds.), *Proceedings of the 4th international conference on practice and theory of automated timetabling*, pp. 404–407, KaHo St.-Lieven, Gent, Belgium, 21–23 August 2002.

Dueck, G., *New optimization heuristics: the great deluge and the record-to-record travel,* Journal of Computational Physics, 104, 86–92, 1993.

Duong, T. A., & Lam, K. H., *"Combining constraint programming and simulated annealing on university exam timetabling", In Proceedings of the 2nd international conference in computer sciences, research, innovation & vision for the future,RIVF2004*, pp. 205–210, Hanoi, Vietnam, 2–5 February, 2004.

Dorigo, M., & Blum, C., *Ant colony optimisation theory: a survey,* Theoretical Computer Science, 344(2–3), 243–278., 2005.

Dowsland, K. A., & Thompson, J., *Ant colony optimisation for the examination scheduling problem*, Journal of Operational Research Society, 56, 426–438, 2005.

Easton, K., Nemhauser, G., & Trick, M., *"Sports scheduling"*, In J. Leung (Eds.), *Handbook of scheduling: algorithms, models, and performance analysis*, Boca Raton: CRC Press, Chap. 52., 2004.

Eley, M., *"Ant algorithms for the exam timetabling problem"*, In E. K. Burke & H. Rudova (Eds.),*Lecture notes in computer science: Vol. 3867, Practice and theory of automated timetabling VI: selected papers from the 6th international conference*, pp. 364–382, Berlin: Springer, 2007.

Erben, W., "A grouping genetic algorithm for graph colouring and exam timetabling", In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079, Practice and theory of automated timetabling III: selected papers from the 3rd international conference*, pp. 132–156. Berlin: Springer, 2001.

Ersoy, E., Özcan, E., & Etaner, A. S., "Memetic algorithms and hyperhill-climbers", *In Proceedings of the 3rd multidisciplinary international conference on scheduling: theory and applications*, pp. 159–166, Paris, France, August 2007.

Even S., Itai A., Shanir A., ”On the complexity of timetable and multicommodity flow problems”, *SIAM J. Comput. 5(4)*, pp.691-703, 1976

Hansen, P., & Mladenovi´c, N., *Variable neighbourhood search: principles and applications*, European Journal of Operational Research, 130, 449–467, 2001.

Kendall, G., & Hussin, N. M., "A tabu search hyper-heuristic approach to the examination timetabling problem at the MARA university of technology", In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science: Vol. 3616. Practice and theory of automated timetabling V: selected papers from the 5th international conference*, pp. 199–218. Berlin: Springer, 2005a.

Kendall, G.,&Hussin, N.M., "An investigation of a tabu search based hyper-heuristic for examination timetabling", In G. Kendall, E. Burke, & S. Petrovic (Eds.), *Selected papers from multidisciplinary scheduling; theory and applications*, pp. 309–328, 2005b.

Krasnogor, N., & Smith, J. E., *A tutorial for competent memetic algorithms: model, taxonomy and design issues*, IEEE Transactions on Evolutionary Computation, 9(5), 474–488, 2005.

Kwan, R., "Bus and train driver scheduling", In J. Leung (Eds.), *Handbook of scheduling: algorithms, models, and performance*, Boca Ratom: CRC Press, Chap. 51, 2004.

Landa Silva, J. D., Burke, E. K., & Petrovic, S., "An introduction to multi-objective meta-heuristics for scheduling and timetabling", In X. Gandibleux, M. Sevaux, K. Sorensen, & V. Tkindt (Eds.), *Lecture notes in economics and mathematical systems: Vol. 535. Multiple objective meta-heuristics*, pp. 91–129, Berlin: Springer, 2004.

Le Huédé, F., Grabisch, M., Labreuche, C., & Savéant, P., *MCS—a new algorithm for multicriteria optimisation in constraint programming*, Annals of Operational Research, 147, 143–174, 2006.

Lin, S. L. M., "A broker algorithm for timetabling problem", In E. K. Burke & P. De Causmaecker Eds., *Proceedings of the 4th international conference on practice and theory of automated timetabling*, pp. 372–386, KaHo St.-Lieven, Gent, Belgium, 21–23 August 2002.

McCollum, B. G. C., "A perspective on bridging the gap betweentheory and practice in university timetabling", In E. K. Burke &H. Rudova (Eds.), *Lecture Notes in Computer Science: Vol. 3867, Practice and theory of automated timetabling VI: selected pa-pers from the 6th international conference*, pp. 3–23, Berlin:Springer, 2007.

McCollum, B., McMullan, P., Burke, E. K., Parkes, A. J., &Qu, R., *The second international timetabling competition: examination timetabling track*, Technical Report

QUB/IEEE/Tech/ITC2007/Exam/v1.0/1., Queen's Belfast University, N. Ireland, 2008.

Merkle, D., & Middendorf, M., “Swarm intelligence”, In E. K. Burke & G. Kendall (Eds.), *Search methodologies: introductory tutorials in optimisation and decision support techniques*, pp. 401–436, Berlin: Springer, 2007.

Merlot, L. T. G., Boland, N., Hughes, B. D., & Stuckey, P. J., “A hybrid algorithm for the examination timetabling problem”, In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes incomputer science: Vol. 2740*, *Practice and theory of automated timetabling IV: selected papers from the 4th international conference*, pp. 207–231, Berlin: Springer, 2003.

MirHassani S.A., *Improving paper spread in examination timetables using integer programming*, Applied Mathematics and Computation, 179, 702- 706, 2006.

Mladenovi´ c, N., & Hansen, P., *Variable neighbourhood search*, Computers and Operations Research, 24(11), 1097–1100, 1997.

Moscato, P., & Norman, M. G., “A “memetic” approach for the travelling salesman problem—implementation of a computational ecology for combinatorial optimisation on message passing systems”, *In Proceedings of the international conference on parallel computing and transputer applications*, pp. 177–186, Amsterdam: IOS Press, 1992.

Naji Azimi, Z., *Comparison of metaheuristic algorithms for examination timetabling problem*, Applied Mathematics and Computation, 16(1–2), 337–354, 2004.

Naji Azimi, Z., *Hybrid heuristics for examination timetabling problem*, Applied Mathematics and Computation, 163(2), 705–733, 2005.

Osman, I. H., & Laporte, G., *Metaheuristics: a bibliography*, Annals of Operational Research, 63, 513–628, 1996.

Paquete, L., & Stützle, T., “Empirical analysis of tabu search for the lexicographic optimisation of the examination timetabling problem, In E. K. Burke & P. De Causmaecker” (Eds.), *Proceedings of the 4th international conference on practice and theory of automated timetabling*, pp. 413–420, KaHo St.-Lieven, Gent,Belgium 21–23 August 2002.

Qu, R., & Burke, E. K., “Adaptive decomposition and construction for examination timetabling problems”, *In Multidisciplinary international scheduling: theory and applications (MISTA’07)*, pp. 418–425, Paris, France, August 2007.

Qu R., Burke E.K., McCollum B., Merlot L.T.G., Lee S.Y., *A survey of search methodologies and automated system development for examination timetabling*, Journal of Scheduling 12, 55-89, 2009.

Reis, L. P., & Oliveira, E., “Constraint logic programming using set variables for solving timetabling problems”, *In 12th international conference on applications of Prolog*, 1999.

Ross, P., Corne, D., & Terashima-Marin, H., “The phase transition niche for evolutionary algorithms in timetabling”, In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science: Vol. 1153, Practice and theory of automated timetabling I: selected papers from the 1st international conference*, pp. 309–324, Berlin: Springer, 1996.

Ross, P., Hart, E., & Corne, D., “Some observations about GA based exam timetabling”, In E. K. Burke & M. W. Carter (Eds.), *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II: selected papers from the 2nd international conference*, pp. 115–129. Berlin: Springer, 1998.

Ross, P., Hart, E., & Corne, D., *Genetic algorithms and timetabling*, In A. Ghosh & S. Tsutsui Eds., Advances in evolutionary computing: theory and applications, pp. 755–771. New York: Springer, 2003.

Ross, P., Marin-Blazquez, J. G., & Hart, E., “Hyper-heuristics applied to class and exam timetabling problems”, *In Proceedings of the 2004 congress on evolutionary computation (CEC2004)*, pp. 1691–1698, Washington: IEEE Press, 2004.

Sheibani, K., “An evolutionary approach for the examination timetabling problems”, In E. K. Burke & P. De Causmaecker (Eds.), *Proceedings of the 4th international conference on practice and theory of automated timetabling*, pp. 387–396, KaHo St.-Lieven, Gent, Belgium, 21–23 August 2002.

Schaerf, A., *A survey of automated timetabling*, Artificial Intelligence Review, 13(2), 87–127, 1999.

Sevkli M.,Uysal Ö., Sarı M., “A Mixed Integer Mathematical Model for Exam Timetabling: A Case Study at Fatih University Vocational School”, *Proceedings of the seventh international conference on the practice and theory of automated timetabling*, Montreal, Canada, 19- 22 August 2008.

Socha, K., Sampels, M., & Manfrin, M., “Ant algorithms for the university course timetabling problem with regard to state-of-the-art”, *In Proceedings of the 3rd European workshop on evolutionary computation in combinatorial optimisation*, pp. 334–345, Essex, UK, April 2003.

Terashima –Marín H., *Combining of GAs and CSP strategies for solving the examination timetabling problem*, PhD Thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey, 1998.

Terashima-Marin, H., Ross, P., & Valenzuela-Rendon, M., “Evolution of constraint satisfaction strategies in examination timetabling”, *In Proceedings of the genetic and evolutionary conference*, pp. 635–642, Orlando, FL., 1999.

Thompson, J., & Dowsland, K., *A robust simulated annealing based examination timetabling system.*, Computers & Operations Research, 25, 637–648, 1998.

Ulker, O., Özcan, E., & Korkmaz, E. E., “Linear linkage encoding in grouping problems: applications on graph colouring and timetabling”, In E. K. Burke & H. Rudova (Eds.), *Lecture notes in computer science: Vol. 3867, Practice and theory of automated timetabling VI: selected papers from the 6th international conference*, pp. 347–363, Berlin: Springer, 2007.

Lofti V. And Cerveny R., *A final exam exam scheduling package*, Journal of the Operational Research Society, 42, 205-216, 1991.

Welsh, D. J. A., & Powell, M. B., *The upper bound for the chromatic number of a graph and its application to timetabling problems*, The Computer Journal, 11, 41–47, 1967.

White, G.M., Chan, P.W., *Towards the construction of optimal examination timetables*, INFOR 17, 219–229, 1979.

White, G. M., “Constrained satisfaction, not so constrained sat-isfaction and the timetabling problem”, In E. K. Burke &W. Erben(Eds.) , *Proceedings of the 3rd international conference on the practice and theory of automated timetabling*, pp. 32–47, Constance, Germany, 16–18 August 2000.

White, G. M., & Xie, B. S., “Examination timetables and tabu search with longer-term memory”, In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science:*

Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference, pp. 85–103, Berlin: Springer, 2001.

White, G. M., Xie, B. S., & Zonjic, S., *Using tabu search withlonger-term memory and relaxation to create examination timetables,* European Journal of Operational Research, 153(16), 80–91, 2004.

Wong, T., Côté, P., & Gely, P., “Final exam timetabling: a practical approach”, *In IEEE Canadian conference on electrical and computer engineering (CCECE 2002),* Vol. 2, pp. 726–731, 2002.

APPENDIX A

```
model ExamTimetabling
uses "mmxprs";
!gain access to the Xpress-Optimizer solver
declarations
coreexam=1..66
times=1..12
room=1..18
common:array (coreexam,coreexam) of integer
C:array(coreexam)of integer
K:array(room) of integer
plan:array(coreexam,times) of mpvar
ass:array(coreexam,room,times) of mpvar
y:array(coreexam,coreexam,times) of mpvar

end-declarations
initializations from 'muh.dat'
common C K
end-initializations

forall(e in coreexam) sum (t in times) plan(e,t) = 1
!All exam must be organized only in one time
forall(e,d in coreexam, t in times | e<d ) plan(e,t) + plan(d,t) - y(e,d,t) <= 1 !linearity
forall(e,d in coreexam, t in times | e<d ) plan(e,t) + plan(d,t) - 2 * y(e,d,t) >= 0 !linearity
c0:=sum(e,d in coreexam, t in times |common(e,d)>0 ) (y(e,d,t)*common(e,d))
!Number of people who have more than one exam in the same time (t) during the final-exam days
```

```
forall(r in room,t in times) sum(e in coreexam)  ass(e,r,t) <= 1
!Given any time t, more tham one exam can not be planned in a room
forall(e in coreexam) sum(r in room,t in times) K(r)*ass(e,r,t)>=C(e)
!Summation of capacities of rooms on which exam e is planned must be bigger than or equal to number of  students who take the course e
forall(e in coreexam,t in times) sum(r in room) ass(e,r,t)<= 100*plan(e,t)
!It constructs the relationship between ass(e,r,t) and plan(e,t)
forall(t in times) sum(e in coreexam,r in room) ass(e,r,t)<= 40
!Number of rooms are used in any time t must be smaller than or equal to number of assistant 40
forall(e in coreexam,t in times) plan(e,t) is_binary
forall(i,j in coreexam, t in times) y(i,j,t) is_binary
forall(e in coreexam,r in room,t in times) ass(e,r,t) is_binary

minimize(c0)

forall(t in times) do
write("slot ", t, ": ")
forall(e in coreexam)
if (getsol(plan(e,t))=1) then write(e," "); end-if
writeln
end-do
writeln("Total loss: ", getobjval)
end-model.
```

APPENDIX B

```
model ExamTimetabling
uses "mmxprs"; !gain access to the Xpress-Optimizer solver

declarations
coreexam=1..375
newexam=1..32
times=1..12
room=1..34
common:array (coreexam,newexam) of integer
common2:array(coreexam,newexam) of integer
C:array(newexam) of integer
K:array(room) of integer
plan:array(coreexam,times) of mpvar
plan2:array(newexam,times) of mpvar
ass:array(newexam,room,times) of mpvar
y:array(coreexam,newexam,times) of mpvar

end-declarations
initializations from '2-1.dat'
common C K
end-initializations

forall(e in newexam) sum (t in times) plan2(e,t) = 1
forall(e in coreexam) sum (t in times) plan(e,t) = 1
forall(e in 1..32,d in newexam, t in times|e=d) (plan(e,t)- plan2(d,t))=0
forall(e in newexam,t in 1..2) plan2(e,t) = 0
forall(e in newexam,t=5) plan2(e,t) = 0
```

forall(e in coreexam,d in newexam,t in times|e>d) plan(e,t) + plan2(d,t) - y(e,d,t) <= 1

forall(e in coreexam,d in newexam, t in times|e>d) plan(e,t) + plan2(d,t) - 2 * y(e,d,t) >= 0

c0:=sum(e in coreexam,d in newexam, t in times |common(e,d)>0 and e>d) y(e,d,t)*common(e,d)

forall(r in room,t in times) sum(e in newexam) ass(e,r,t) <= 1

forall(e in newexam) sum(r in room,t in times) K(r)*ass(e,r,t)>=C(e)

forall(e in newexam,t in times) sum(r in room) ass(e,r,t)<= 100*plan2(e,t)

plan(33,12)=1 plan(34,7)=1 plan(35,6)=1 plan(36,1)=1 plan(37,2)=1 plan(38,5)=1 plan(39,8)=1 plan(40,3)=1 plan(41,8)=1 plan(42,7)=1 plan(43,10)=1 plan(44,12)=1 plan(45,4)=1 plan(46,8)=1 plan(47,9)=1 plan(48,1)=1 plan(49,9)=1 plan(50,12)=1 plan(51,1)=1 plan(52,3)=1 plan(53,2)=1 plan(54,11)=1 plan(55,8)=1 plan(56,5)=1 plan(57,6)=1 plan(58,12)=1 plan(59,11)=1 plan(60,4)=1 plan(61,7)=1 plan(62,1)=1 plan(63,3)=1 plan(64,12)=1 plan(65,1)=1 plan(66,9)=1 plan(67,8)=1 plan(68,2)=1 plan(69,10)=1 plan(70,9)=1 plan(71,7)=1 plan(72,10)=1 plan(73,5)=1 plan(74,3)=1 plan(75,4)=1 plan(76,11)=1 plan(77,2)=1 plan(78,10)=1 plan(79,5)=1 plan(80,12)=1 plan(81,3)=1 plan(82,11)=1 plan(83,10)=1 plan(84,5)=1 plan(85,6)=1 plan(86,4)=1 plan(87,4)=1 plan(88,11)=1 plan(89,5)=1 plan(90,9)=1 plan(91,10)=1 plan(92,7)=1 plan(93,1)=1 plan(94,11)=1 plan(95,3)=1 plan(96,6)=1 plan(97,3)=1 plan(98,4)=1 plan(99,8)=1 plan(100,5)=1 plan(101,6)=1 plan(102,2)=1 plan(103,5)=1 plan(104,8)=1 plan(105,10)=1 plan(106,4)=1 plan(107,3)=1 plan(108,9)=1 plan(109,8)=1 plan(110,11)=1 plan(111,7)=1 plan(112,12)=1 plan(113,1)=1 plan(114,6)=1 plan(115,7)=1 plan(116,12)=1 plan(117,5)=1 plan(118,6)=1 plan(119,9)=1 plan(120,11)=1 plan(121,7)=1 plan(122,1)=1 plan(123,2)=1 plan(124,6)=1 plan(125,12)=1 plan(126,3)=1 plan(127,3)=1 plan(128,1)= plan(129,4)=1 plan(130,9)=1 plan(131,5)=1 plan(132,7)=1 plan(133,8)=1 plan(134,11)=1 plan(135,10)=1 plan(136,6)=1 plan(137,11)=1 plan(138,5)=1 plan(139,11)=1 plan(140,10)=1 plan(141,12)=1 plan(142,4)=1 plan(143,3)=1 plan(144,6)=1 plan(145,1)=1 plan(146,2)=1 plan(147,7)=1 plan(148,1)=1 plan(149,8)=1 plan(150,1)=1 plan(151,11)=1 plan(152,10)=1 plan(153,9)=1 plan(154,5)=1 plan(155,6)=1 plan(156,2)=1 plan(157,4)=1 plan(158,2)=1 plan(159,6)=1 plan(160,1)=1 plan(161,4)=1 plan(162,6)=1 plan(163,2)=1 plan(164,4)=1 plan(165,8)=1 plan(166,5)=1 plan(167,11)=1 plan(168,9)=1 plan(169,12)=1

plan(170,5)=1	plan(171,1)=1	plan(172,7)=1	plan(173,10)=1	plan(174,3)=1
plan(175,12)=1	plan(176,2)=1	plan(177,3)=1	plan(178,10)=1	plan(179,3)=1
plan(180,12)=1	plan(181,12)=1	plan(182,7)=1	plan(183,9)=1	plan(184,5)=1
plan(185,7)=1	plan(186,11)=1	plan(187,3)=1	plan(188,10)=1	plan(189,8)=1
plan(190,12)=1	plan(191,1)=1	plan(192,10)=1	plan(193,4)=1	plan(194,6)=1
plan(195,9)=1	plan(196,7)=1	plan(197,5)=1	plan(198,1)=1	plan(199,2)=1
plan(200,11)=1	plan(201,12)=1	plan(202,4)=1	plan(203,8)=1	plan(204,3)=1
plan(205,6)=1	plan(206,9)=1	plan(207,9)=1	plan(208,1)=1	plan(209,2)=1
plan(210,11)=1	plan(211,5)=1	plan(212,7)=1	plan(213,4)=1	plan(214,12)=1
plan(215,3)=1	plan(216,8)=1	plan(217,11)=1	plan(218,5)=1	plan(219,7)=1
plan(220,6)=1	plan(221,2)=1	plan(222,1)=1	plan(223,3)=1	plan(224,12)=1
plan(225,10)=1	plan(226,4)=1	plan(227,1)=1	plan(228,5)=1	plan(229,6)=1
plan(230,3)=1	plan(231,2)=1	plan(232,9)=1	plan(233,8)=1	plan(234,10)=1
plan(235,5)=1	plan(236,1)=1	plan(237,7)=1	plan(238,12)=1	plan(239,8)=1
plan(240,9)=1	plan(241,4)=1	plan(242,11)=1	plan(243,7)=1	plan(244,4)=1
plan(245,6)=1	plan(246,1)=1	plan(247,3)=1	plan(248,9)=1	plan(249,3)=1
plan(250,11)=1	plan(251,2)=1	plan(252,12)=1	plan(253,2)=1	plan(254,6)=1
plan(255,8)=1	plan(256,1)=1	plan(257,7)=1	plan(258,3)=1	plan(259,8)=1
plan(260,4)=1	plan(261,11)=1	plan(262,5)=1	plan(263,2)=1	plan(264,10)=1
plan(265,6)=1	plan(266,12)=1	plan(267,3)=1	plan(268,2)=1	plan(269,11)=1
plan(270,4)=1	plan(271,5)=1	plan(272,3)=1	plan(273,6)=1	plan(274,8)=1
plan(275,3)=1	plan(276,7)=1	plan(277,11)=1	plan(278,2)=1	plan(279,10)=1
plan(280,8)=1	plan(281,2)=1	plan(282,9)=1	plan(283,11)=1	plan(284,10)=1
plan(285,4)=1	plan(286,7)=1	plan(287,9)=1	plan(288,3)=1	plan(289,10)=1
plan(290,1)=1	plan(291,8)=1	plan(292,5)=1	plan(293,6)=1	plan(294,5)=1
plan(295,4)=1	plan(296,12)=1	plan(297,7)=1	plan(298,4)=1	plan(299,8)=1
plan(300,1)=1	plan(301,2)=1	plan(302,10)=1	plan(303,12)=1	plan(304,3)=1
plan(305,9)=1	plan(306,8)=1	plan(307,4)=1	plan(308,6)=1	plan(309,10)=1
plan(310,8)=1	plan(311,6)=1	plan(312,7)=1	plan(313,1)=1	plan(314,4)=1
plan(315,6)=1	plan(316,5)=1	plan(317,3)=1	plan(318,9)=1	plan(319,11)=1
plan(320,12)=1	plan(321,10)=1	plan(322,9)=1	plan(323,6)=1	plan(324,1)=1
plan(325,6)=1	plan(326,8)=1	plan(327,9)=1	plan(328,5)=1	plan(329,3)=1
plan(330,8)=1	plan(331,7)=1	plan(332,1)=1	plan(333,2)=1	plan(334,12)=1
plan(335,11)=1	plan(336,5)=1	plan(337,4)=1	plan(338,9)=1	plan(339,6)=1

plan(340,7)=1 plan(341,2)=1 plan(342,3)=1 plan(343,10)=1 plan(344,1)=1 plan(345,1)=1 plan(346,5)=1 plan(347,4)=1 plan(348,3)=1 plan(349,8)=1 plan(350,11)=1 plan(351,6)=1 plan(352,4)=1 plan(353,6)=1 plan(354,12)=1 plan(355,10)=1 plan(356,2)=1 plan(357,1)=1 plan(358,9)=1 plan(359,11)=1 plan(360,6)=1 plan(361,6)=1 plan(362,5)=1 plan(363,7)=1 plan(364,8)=1 plan(365,8)=1 plan(366,7)=1 plan(367,5)=1 plan(368,1)=1 plan(369,9)=1 plan(370,10)=1 plan(371,2)=1 plan(372,9)=1 plan(373,12)=1 plan(374,8)=1 plan(375,3)=1

forall(t in times) sum(e in newexam,r in room) ass(e,r,t)<= 40

c1:=sum(e in coreexam,d in newexam, t=1 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c2:=sum(e in coreexam,d in newexam, t=2 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c3:=sum(e in coreexam,d in newexam, t=3 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c4:=sum(e in coreexam,d in newexam, t=4 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c5:=sum(e in coreexam,d in newexam, t=5 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c6:=sum(e in coreexam,d in newexam, t=6 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c7:=sum(e in coreexam,d in newexam, t=7 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c8:=sum(e in coreexam,d in newexam, t=8 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c9:=sum(e in coreexam,d in newexam, t=9 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c10:=sum(e in coreexam,d in newexam, t=10 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c11:=sum(e in coreexam,d in newexam, t=11 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c12:=sum(e in coreexam,d in newexam, t=12 |common(e,d)>0 and e>d) (y(e,d,t)*common(e,d))

c0=c1+c2+c3+c4+c5+c6+c7+c8+c9+c10+c11+c12

```
forall(e in coreexam,t in times) plan(e,t) is_binary
forall(e in newexam,t in times) plan2(e,t) is_binary
forall(i in coreexam,j in newexam, t in times ) y(i,j,t) is_binary
forall(e in newexam,r in room,t in times) ass(e,r,t) is_binary
minimize(c0)
forall(t in times) do
  write("slot ", t, ": ")
  forall(e in newexam)
   if (getsol(plan2(e,t))=1) then write(e," "); end-if
  writeln
 end-do
 writeln("Total loss: ", getobjval)
end-model
```