GOAL PROGRAMMING APPROACH TO SOLVE THE TIMETABLING PROBLEM AT TURKISH MILITARY ACADEMY

The Institute of Economics and Social Sciences
of
Bilkent University

by

TUNA ŞAHİN

In Partial Fulfillment of the Requirements for the Degree
of
MASTER OF BUSINESS ADMINISTRATION

in

THE DEPARTMENT OF MANAGEMENT
BİLKENT UNIVERSITY
ANKARA

September 2004

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.

Assist. Prof. Yavuz GÜNALAY
Supervisor

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.

Assist. Prof. Nail AKAR
Examining Committee Member

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.

Prof. Erdal EREL
Examining Committee Member

Approval of the Institute of Economics and Social Sciences

Prof. Kürşat AYDOĞAN
Director

ABSTRACT

GOAL PROGRAMMING APPROACH TO SOLVE THE TIMETABLING PROBLEM AT TURKISH MILITARY ACADEMY

Şahin, Tuna
Master of Business Administration
Supervisor: Assist. Prof. Yavuz Günalay
September 2004

The aim of this study is to propose a goal programming model to solve the timetabling problem at Turkish Military Academy. Since the problem is NP-complete, it's not easy to find an optimal solution all the time. It takes a lot of time of the people who are responsible to prepare the timetables of TMA. The model consists in all of the requirements, and is tested with the real data provided by Planning and Programming Department. Since the problem is so big to solve at once as a whole, a five-step iterative solution procedure is proposed. There are four priorities, three for teacher preferences and one for teaching loads. The model aims to minimize the deviations from the preferences and teaching loads of the teachers. Solution process produced a feasible and near-optimal timetable after four steps, in a reasonably short time compared to hand-made timetabling procedure. The result was improved by making some modifications in step five.

In the conclusion, we mentioned the problems we faced, and presented our suggestions for future research.

Keywords: Timetabling, Scheduling, Goal Programming, School Timetabling

ÖZET

KARA HARP OKULU DERS ÇİZELGELEME PROBLEMİNİN HEDEF PROGRAMLAMA YÖNTEMİYLE ÇÖZÜLMESİ

Şahin, Tuna
M.B.A
Tez Yöneticisi: Yrd. Doç. Dr. Yavuz Günalay
Eylül 2004

Bu çalışmanın amacı, Kara Harp Okulu'nun ders çizelgeleme problemini çözecek bir hedef programlama modeli ortaya koymaktır. Ders çizelgelerinin hazırlanması planlayıcıların çok fazla zamanını almakta, ve her zaman için en iyi çözüme ulaşmak mümkün olmamaktadır. Önerilen model tüm problem gereksinimlerini içermektedir ve Plan ve Program Şubesince sağlanan gerçek bilgilerle test edilmiştir. KHO ders çizelgeleme problemi bir seferde çözülemeyecek kadar büyük olduğundan, problemi daha küçük parçalara ayıran ve öğrenci ve öğretmen gruplarını bir sıra dahilinde eşleştirerek sonuca ulaşan beş aşamalı bir çözüm yolu önerilmiştir. Hedef fonksiyonu içinde üç tanesi öğretmen tercihleri, bir tanesi de öğretmen ders yükleri için olmak üzere toplam dört öncelik grubu oluşturulmuştur. Modelin amacı öğretmen tercihleri ve ders yüklerinden oluşacak sapmaları minimize etmektir. İlk dört aşama neticesinde elde edilen sonuç tüm sistem gereksinimlerini karşılamaktadır ve en iyi sonuca yakın bir ders çizelgesi oluşturmaktadır. Beşinci aşamada yapılan elle müdahaleler sonucunda çözüm geliştirilmiştir. Önerilen modelle sonuca ulaşabilmek için elle yapılan çizelgelemeye göre daha az zaman harcanmıştır. Sonuç bölümünde karşılaşılan problemlere değinilmiş ve gelecekteki araştırmalar için öneriler sunulmuştur.

Anahtar Kelimeler: Çizelgeleme, Program, Hedef Programlama, Okul Ders Çizelgeleme

ACKNOWLEDGEMENTS

I wish to express my gratitude to Assist. Prof. Yavuz Günalay for his valuable supervision and patience throughout the study.

I thank to examining committee members for their great contributions.

I also thank to my wife, Deniz Zehra ŞAHİN, for her endless patience and support.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The purpose of this study is to provide a goal programming model that will be helpful in preparing timetables for Turkish Military Academy (TMA). In the next section, a general definition and the characteristics of the timetabling problem will be given.

1.1 Defining the Timetabling Problem

Constructing timetables is a real world scheduling problem on which the schools and the universities must put emphasis and solve each year, or even each semester. Depending on the size of the problem, institutional goals, preferences of the instructors and the students, this activity can require a huge human effort and be very time consuming.

Timetabling problems draw the attention of researchers for more than forty years. Since the problem differs from one institution to another, the researchers proposed a great variety of models and solution approaches. For a review of approaches, the interested readers are referred to (de Werra, 1985; Burke et al., 1995; Schaerf, 1999; Burke and Petrovic, 2002).

Timetabling problem in educational institutions is generally classified into three groups: examination timetabling, course timetabling, and class/teacher timetabling. (e.g., Schaerf, 1999; Carrasco and Pato, 2004) In this thesis, we will also use university timetabling for course timetabling, and school timetabling for class/teacher timetabling.

In examination timetabling problem, it's required to schedule a number of exams within a given amount of time. The examination timetabling is similar to the course timetabling, and it is very difficult to make a distinction between them. Some specific problems can be formulated in both ways. Schaerf (1999) lists some characteristics of examination timetabling that make it different from course timetabling as follows:

- There is only one exam for each course.
- The conflict constraints are generally stricter. The examinations with the same student group cannot overlap.
- There is usually one examination for a student group per day, no too many consecutive exams.
- The number of periods can vary, where it's fixed in course timetabling.
- More than one exam can be planned per room.

Since the examination timetabling is out of scope of this thesis, those who are related in the issue may look at the surveys Carter (1986) and Schaerf (1999) made.

The researchers made various definitions of timetabling. Main principal in all of the definitions is the same. A teacher or a student group cannot be assigned to more than one lecture or a lesson for a time block. Schaerf (1999) defines the issue as follows:

" The timetabling problem consists in scheduling a sequence of lectures between teachers and students in a prefixed period of time (typically a week), satisfying a set of constraints of various types."

It could be somehow difficult to make a clear distinction between course timetabling and class/teacher timetabling. Burke and Elliman (1994) states that although they look the same basically, "There are many minor differences." In most universities, there's always a flexible curricula, rather than a fixed one. Students can choose from a variety of courses, and the class sizes can vary. On the other hand, there's usually a fixed curricula, and the class sizes are the same in a school, and even in some universities. It is assumed that a school has adequate number of rooms and labs for any class. (Burke and Elliman, 1994)

1.2 Characteristics of Timetabling Problems

The constraints that are to be satisfied in every scheduling problem are called *hard constraints.* Some of the examples are:

- A teacher teaches only one course and cannot be at more than one place at the same time block.
- A student group cannot attend to more than one class at a time block.
- All courses are to be assigned to a time block.

Every institute has its own requirements and unique constraints. These are called *soft constraints.* Some of the examples can be listed as:

- Instructors may not want to give lectures on some weekdays.
- Courses that have a teaching period more than two hours should be splitted into weekdays (e.g. a four-hour course, two on Monday and two on Wednesday.)

Under no circumstances, the hard constraints can be violated to have feasible timetables. Once a model verifies the hard constraints, then its quality is measured depending on how much it satisfies the soft ones. The more soft constraints are satisfied, the better the model is.

Those researchers, who are interested in school timetabling, usually take the most commonly known *class/teacher model* as the basic model. Then they improve the basic model according to the requirements of the problem they are interested in. A generalized class/teacher model can be found in Asratian and de Werra (2002).

In this thesis, we present a goal programming model to solve the timetabling problem at Turkish Military Academy. Since the problem as a whole was too big, we tested the model only with the first-class of TMA. The model can be applied to other classes with necessary modifications, and timetables for the whole school can be generated. Since the problem for the first-class was still so big, we used the heuristic solution procedure given in chapter 4 to solve the problem. We reached to a near-optimal timetable for the first-class of TMA.

In this chapter, the timetabling problem is defined and the general characteristics of the problem are given. The next chapter will be devoted to the review of the related literature. The proposed model will be introduced in chapter 3, and an implementation of the model at Turkish Military Academy will be provided in chapter 4. The discussion about the results of the solution process will be given in the same chapter. The final conclusions will be in chapter 5.

As we stated above that each situation is unique, the researchers proposed a variety of formulations and approaches since there's not a global model that fits into every problem. Now we are going to have a look at the related literature.

CHAPTER 2

LITERATURE REVIEW

It’s very difficult to decide on the solution approach at the beginning. As we stated earlier that the requirements are different for every institution, it wouldn’t be surprising that a solution procedure produces poor results, which is proved to be satisfactory for another problem earlier. The combination of the approaches can provide satisfactory results for some cases. Now let’s look at the literature and find out the approaches which researchers have utilized to model and solve timetabling problems for over forty years.

Simulation Methods, and Graph Coloring are the first approaches used to solve timetabling problems in 1960s. Welsh and Powell (1967) utilized the graph coloring technique to solve a timetabling problem, and Schmidt and Strohlein (1979) reported on simulation techniques. (Daskalaki et al., 2004)

Mathematical Programming is another approach used in solving timetabling problems. In recent years, the technological advancements in computers made these techniques popular again. Akkoyunlu (1973) and Lawrie

(1969) proposed linear and integer programming models for a simplified version of the problem. These can be listed among the first approaches in mathematical programming. Akkoyunlu was aiming to prevent any conflicts during the assignment process basically. (Birbas et al., 1997)

Birbas et al. (1997) proposed an integer programming model which they call “flexible” and “modular”. They tested the model at a Greek high school with six sections, and found an optimal solution that satisfies both the hard and the soft constraints. Daskalaki et al. (2004) presented an integer programming formulation for a university case. They claim that their model is a complete one, and produces timetables that overcomes any conflicts may occur. Üstünel (2001) proposed an integer programming model and a heuristic solution procedure to solve the timetabling problem at Turkish Military Academy. They tested the model with the real data provided by TMA. They stated that the model satisfied the teacher preferences to a level using the appropriate objective function coefficients. And they described the results as a “good timetable” both for the teachers and the planners.

Multi Objective Programming can be used in timetabling successfully. Burke and Petrovic (2002) stated that “multicriteria decision making methods can lead to a significant insight into timetabling problems and a flexibility in handling the constraints that is not provided by existing methods.”

Schniederjans and Kim (1987) presented a zero-one goal programming model to assign teaching staff to specific courses based on departmental requirements and personal preferences of instructors. Besides the course offerings and teaching load constraints, the model was introducing faculty-course teaching preferences. The problem they formulated included only 87 decision variables and 36 goal constraints, and it took less than 20 minutes to solve the problem on a 512 K Macintosh. The first and second priority, offering all the courses and teaching loads, were fully achieved. The third priority, which is faculty-teaching preferences, is partially achieved.

Dowsland (1990) described the development of a model for a multi-objective university timetabling problem. The solution was to satisfy a set of objectives. They translated some of the objectives into constraints, and defined the problem in terms of the optimization of a single objective, which is student disappointment. They proposed three models –graph coloring, set partitioning, and simulated annealing-, and gave a discussion on the advantages and the disadvantages of using each model to find a solution.

Badri (1996) also proposed a multiobjective zero-one course scheduling model. The model was composed of two stages. In stage one, it was assigning the instructors to the courses, and in stage two it was assigning these faculty-course combinations to suitable time slots. The first stage was based on the model Schniederjans and Kim (1987) offered. The formulated problem had 162 decision variables and 82 goal constraints in stage one, and 126 decision variables and 76 goal constraints in stage two. Both stages were solved in less than one minute. The priority for teaching loads and the priority for faculty course-time preferences are partially fulfilled.

Badri et al. (1998) again presented a multiobjective zero-one course scheduling model. This model provided a one-stage solution to the problem introduced in Badri (1996), rather than two stages. It was assigning the faculty to the courses and also to the time slots simultaneously, depending on the departmental requirements and faculty preferences both for courses and time slots. The faculty-time and faculty –course preferences are not fully achieved in the solution.

Meta-heuristic Methods are relatively newer approaches. These methods are used to reach optimal solutions without being trapped at a local optima. They start with one or more initial solutions, and then try to improve objective function value by iterative improvements. Tabu search, Simulated Annealing, and Genetic Algorithms can be listed among these approaches.

Costa (1994) presented a general technique based on Tabu Search. He tested the algorithm with different real world cases, and reached to solutions that he calls "satisfactory". de Werra (1997) discussed a tabu search approach which may deal with a variety of requirements. Alvarez-Valdes et al. (2002) developed a computer package that depends on a tabu search procedure basically. They stated that the package produced satisfactory results for the test period.

Abramson et al. (1999) used Simulated Annealing for school timetabling problem and compared the performance of six different SA cooling schedules. They stated that the schedules that “incorporated a method to compute the temperature that is used as the reheating point” produced better results in less time than any other scheme. Abramson (1991) examined the use of SA with sequential and parallel algorithms. He showed that the parallel algorithm could provide a faster solution than the equivalent sequential algorithm.

Yu and Sung (2002) proposed a sector based Genetic Algorithm to solve a university timetabling problem. They stated that the first experimental results showed the algorithm is promising. Colorni et al. (1998) present the results of an implementation of the most known three metaheuristics to a timetabling problem with “real-world applications.” Comparing the results obtained by simulated annealing, tabu search and two versions of genetic algorithm, they stated that GA with local search and tabu search based on temporary problem relaxations both provided better solutions than simulated annealing and handmade timetables.

Constraint Programming is also a newer approach. A variety of combinatorial problems, including timetabling, can be solved using constraint satisfaction approach. Brailsford et al. (1999) reported on algorithms and applications of constraint satisfaction. They stated that the researchers in artificial intelligence usually utilize a constraint satisfaction approach when solving this sort of problems. Deris et al. (2000) presented a solution procedure based on a constraint-based reasoning technique implemented in an object-oriented approach.

The algorithm was tested with real data provided by a college offering professional courses, and a timetable was generated in less than 33 minutes, which might have required several weeks if solved manually. They stated that the system could be modified and adapted to support the changes easily. Valouxis and Housos (2003) modeled and solved the timetabling problem for a high school environment using a constraint programming approach. They stated that the problem was solved in “acceptable” time and and the results were close to their optimal values. Baker et al. (2002) proposed integer programming and constraint programming models. The integer programming model provided “good feasible” solutions for the problems in their test set, if not optimal. But constraint logic programming algorithms yielded optimal solutions for the problems in the set, even for the ones the integer programming model was unsuccessful.

Neural Network Approaches were also applied to timetabling problem. Smith et al. (2003) reported on the use of discrete Hopfield neural networks for solving school timetabling problems. Two alternative formulations, a standard Hopfield-Tank approach and a more compact formulation which makes Hopfield network be competitive with swapping heuristics, were provided for the problem. They showed how these formulations could lead to different solutions. They compared the results with the ones obtained by using greedy simulated annealing and tabu search. They stated that the models were capable of generating good quality solutions to extremely difficult timetabling problems.

Carrasco and Pato (2004) examined the use of neural network based heuristics to the class/teacher timetabling problem. They reported on two approaches, the first one using a Potts mean-field annealing simulation, and the second one using a discrete neural network simulation. They stated that the discrete approach performed better than the other in terms of solution quality and execution time.

Some researchers studied on Computer Based Information Systems to solve timetabling problem. They created such systems that usually use a database, which includes all the necessary data to model and solve the problem. One of the advantages of these systems is that the data are kept always up-to-date. Departments have the authority to access to the related part of the database, and make necessary changes. The systems generate timetables for each department separately. For example, Dimopolou and Miliotis (2004) presented a computer network based system that uses a centralized database and utilizes an integer programming model. They tested the system with data provided by the Athens University of Economics and Business. The problems were solved on a pc type computer, with XPRESS-MP, and they got satisfactory results most of the time. They also presented a computer system to solve an examination timetabling problem in (Dimopolou and Miliotis, 2001). Abbas and Tsang (2004) presented a timetabling system that utilizes constraint satisfaction techniques. They used a formal specification language, called DEPICT 0.1, to specify the timetabling problem. They stated that, having been used by a university and a school, the program is “well tested” and “operational”. Karataş (1996) claimed that the solution procedure to the course scheduling problem can be based on a system analysis approach, and the amount of the data transferred by the user to the system is increased with the help of an interactive software. They didn't use any mathematical models in the solution procedure, and they define the whole procedure as a system analysis approach. In addition to their methodology, they presented a computer software to test and to implement their algorithm. They defined the methodology "flexible" as a solution system. Botsali (2000) designed a system that utilizes both constraint programming and mathematical programming techniques to solve timetabling problem at Bilkent University. The problem is solved in three stages. In the first two stages, a course schedule is formed by utilizing the constraint programming techniques. In the last stage, the courses are assigned to the classrooms by using mathematical programming. The system is tested by the real data from the past semesters, and the required courses for all students were successfully scheduled.

CHAPTER 3

THE PROPOSED MODEL

We are going to propose a goal programming model for class/teacher type of timetabling problem. As we know, students at each class of a school are divided into groups. We will call these student groups as *sections* and use the term *meeting* for teacher-course-section combinations. We will define the problem in TMA briefly in the next paragraph, and then give general characteristics of our model.

Since the curriculum is fixed, all of the planned courses should be offered to all sections. There are no elective courses, and the students have to attend all those compulsory courses. The teachers are pre-assigned to the courses, and they teach only one type of course. Only one teacher offers a course to a section. The teachers can make preferences for time periods.

We are going to introduce some other requirements in addition to hard constraints. Our hard constraints are the same as usual; all courses for all sections should be offered, and only one meeting should be assigned for a certain time slot, both for teachers and sections. The other constraints we introduce are; teaching loads, daily limits, teacher preferences, consecutiveness and a course-section combination offered only to a teacher.

3.1 Variables, Constants and Notations

I = {1,2,…,n} : Set of teachers.

CI = {ai,bi,…}: These are the subsets of teachers. Each one refers to the group of teachers who are assigned to a course. For example, ai = {1,2,3} refers to the teachers who are assigned to course a.

J = {a1,a2,…,af,b1,b2,…,bf…} : Set of course-section combinations. Letters resemble the initials of the courses, and the numbers are for sections. The letter f stands for the last section's number.

K = {1,2,…,o} : Set of time periods. A period represents one-hour length of time.

pk, sk, ck, pek, cuk : Disjoint subsets of K, where they represent available periods of days of a week in order (i.e., pk for Monday, sk for Tuesday, ck for Wednesday, pek for Thursday, cuk for Friday); and pk □ sk □ ck □ pek □ cuk = K.

DY = {1, 2, 3, 4, 5} : Days of the week.

SC = {1,2…,s_t} : Set of sections in a class.

CG = {c1,c2,…,c_{st}} : These are the subsets of course-section combinations. Each one refers to the group of courses that has to be assigned to section *sc*. For example, c1 = {a1,b1,…,m1}.

n Total number of teachers to assign.

m Total number of course-section combinations to assign.

nj Total number of teachers can be assigned to a course-section combination j.

mi Total number of course-section combinations that the teacher i can be assigned to.

m_{cg} Total number of course-section combinations can be assigned to a section.

o Total number of time periods to assign.

g Total number of daily time periods to assign.

tl(i) Teaching load of a teacher.

V ={1,2,…,vt} Teacher priority group 1. vt is the total number of preferences made by teachers in priority group one.

Y ={1,2,…,yt} Teacher priority group 2. yt is the total number of preferences made by teachers in priority group two.

Z ={1,2,…,zt} Teacher priority group 3. zt is the total number of preferences made by teachers in priority group three.

d_i^+, d_i^- Positive and negative deviations from teaching loads, i = 1,2,…,n.

d_v^+, d_v^- Positive and negative deviations from teacher preferences in priority group one, v = 1,2,…,vt.

d_y^+, d_y^- Positive and negative deviations from teacher preferences in priority group two, y = 1,2,…,yt.

d_z^+, d_z^- Positive and negative deviations from teacher preferences in priority group three, z = 1,2,…,zt.

s_j Total number of hours a course-section combination has to be given.

s_{jdy} Total number of hours a course-section combination has to be given on day *dy*.

$$x_{ijk} \text{ Decision variable } = \left\{ \begin{array}{ll} 1 & , \text{ if teacher } i \text{ is assigned to a course-section combination } j \text{ at time slot } k \\ 0 & , \text{ otherwise} \end{array} \right\}$$

$$d_{ijk} \text{ Availability of classes } = \left\{ \begin{array}{ll} 1 & , \text{ when a teacher } i \text{ can be assigned to a course-section combination } j \text{ at time slot } k, \\ & \text{ if the related } j \text{ is available at time slot } k \\ 0 & , \text{ otherwise} \end{array} \right\}$$

$$q_{jdy} = \left\{ \begin{array}{ll} 1 & , \text{ if course-section combination } j \text{ is assigned on day } dy \\ 0 & , \text{ otherwise} \end{array} \right\}$$

$$h_{ij} = \left\{ \begin{array}{ll} 1 & , \text{ if teacher } i \text{ is giving course } j \\ 0 & , \text{ otherwise} \end{array} \right\}$$

3.2 Constraints

The first group of constraints ensures that all course-section combinations are assigned. This is a hard constraint.

$$\sum_{i=1}^{n} \sum_{k=1}^{o} d_{ijk} * x_{ijk} = s_j \text{,} \quad \text{for all j.} \qquad (1)$$

The second group of constraints represents the teaching loads for teachers. One constraint should be written for each teacher.

$$\sum_{j=1}^{m} \sum_{k=1}^{o} d_{ijk} * x_{ijk} + d_i^- - d_i^+ = tl(i)_{,} \quad \text{for all i.} \qquad (2)$$

The third group of constraints ensures that only one course-section combination is assigned to a teacher for a certain time period. This is the second hard constraint.

$$\sum_{j=1}^{m} x_{ijk} \leq 1 \text{,} \quad \text{for all i and k.} \qquad (3)$$

The fourth group of constraints ensures that only one teacher (course) is assigned to a section for a certain time period. This is also introduced as a hard constraint. The constraints should be written for all the members of subset cg.

$$\sum_{i=1}^{n} \sum_{j=1}^{mcg} x_{icg(j)k} \leq 1 \ , \quad \text{for all k and cg} = \{c1, c2, \ldots, c_{st}\}. \qquad (4)$$

The fifth constraint is introduced for the daily limits. The administrators may require that the courses, whose total teaching hours exceed some certain time periods, should be split into the weekdays. For example, let's say that a course's duration is four hours. The first two-hour period can be planned on Monday, and the other two-hour period should be planned on some other weekday except Monday. The constraints should be written for all multi-period *j* and days of the week.

$$\sum_{i=1}^{n} \sum_{k=1}^{g} x_{ijk} = s_{jdy*}q_{jdy} \ , \quad \text{for all multi-period j and day of the week.} \qquad (5)$$

But this constraint is not enough for those courses that cannot be divided into equal periods. These are the courses that last 3 or 5 periods. In such a case, we should create dummy courses. For example, let's take a three-hour course. It should be divided this way; two as the main course, and one as the dummy course. The dummy course should be treated as a separate course, but requiring the same conditions as the main one does. It means that the same teachers should offer it with the main course. At this point, another constraint is required to ensure that the main course and the dummy one are not assigned on the same day.

$$\sum_{i=1}^{n} \sum_{k=1}^{g} x_{ijk} + x_{ij(d)k} \leq 2 , \quad \text{for j that has a dummy course j(d) and each day of the week.} \qquad (6)$$

The sixth constraint group is introduced to the model to reflect the preferences of the teachers for time periods. The teachers are supposed to be pre-assigned to the courses. So they don’t make preferences for the courses. We created three priority groups. The administrator and the planning staff are supposed to form these groups according to the importance of preferences of the teachers. The constraints should be written for all teachers if they all declare their preferences. If not, the constraints should be written only for those who declare preferences. In any case, each k represents the time period at which the teachers don’t want to give lectures. Constraint set 6 stands for priority group one, 7 for group two, and 8 for group three.

$$\sum_{j=1}^{mi} x_{ijk} + d_v^- - d_v^+ = 0 , \quad \text{for all i in priority group I, and k that the related i is not wiling to teach, v □ V.} \qquad (7)$$

$$\sum_{j=1}^{mi} x_{ijk} + d_y^- - d_y^+ = 0 , \quad \text{for all i in priority group II, and k that the related i is not wiling to teach, y □ Y.} \qquad (8)$$

$$\sum_{j=1}^{mi} x_{ijk} + d_z^- - d_z^+ = 0 , \quad \text{for all i in priority group III, and k that the related i is not wiling to teach, z □Z.} \qquad (9)$$

The seventh group of constraints is the consecutiveness constraint. A course that lasts more than one period should not be split and all it's periods are to be consecutive. For example, if a two-hour course is assigned on Monday and the first hour is assigned to time slot one, then the second hour should be assigned to nowhere else but to the second time slot. The constraints should be written for each day separately, for all j and the time periods that cannot be consecutive for that j on a day.

$$\sum_{i=1}^{nj} (x_{ijk} + x_{ij(k+w)}) \leq 1 \text{ , for all j and the time periods that cannot be consecutive for that j on a day, } (k+w) = k+s_{jdy},\ldots,g \text{ .} \tag{10}$$

The last constraint group is introduced to provide that a course-section combination j is assigned to only one teacher. In other words, only one teacher can give a course to a section, and it's not allowed that a course-section combination is assigned to two or more teachers. On the right-hand-side, the h_{ij} reflects that whether the j is assigned to the teacher i or not. The total of h_{ij} must be equal to 1.

$$\sum_{k=1}^{o} x_{ijk} = s_j{*}h_{ij,} \text{ for all i who are giving course j .} \tag{11}$$

$$\sum_{i=1}^{n} h_{ij} = 1 \text{ , for all j .} \tag{12}$$

3.3 Objective Function

The objective function has four priorities; one for teaching loads and three for teacher preferences. The administration is authorized to determine and assign the priorities.

$$\text{Min } z = P_1 \sum_{i=1}^{n} (d_i^- + d_i^+) + P_2 \sum_{v=1}^{vt} (d_v^- + d_v^+) + P_3 \sum_{y=1}^{yt} (d_y^- + d_y^+) + P_4 \sum_{z=1}^{zt} (d_z^- + d_z^+)$$

CHAPTER 4

IMPLEMENTATION

In this chapter, the model proposed in the previous section will be adapted to the timetabling problem of the first-class students of Turkish Military Academy (TMA). We will pursuit a non-preemptive approach and assign weights to the priorities that were defined by the administration. We will present the application of the model only to the first-class for the reason that the timetabling problem for TMA as a whole is too large to solve at once. The model can be applied to each class after modifying the goal constraints according to the data of the related class.

In the next section, a brief description of the timetabling problem for the first class of TMA and the data provided by the Planning Department will be presented. Then the model will be adapted to the timetabling problem of the first class. Finally, after presenting the methodology for the solution process, the results of the solution process will be discussed.

4.1 Defining The Problem

The first class consists of 29 sections, all of which have almost the same number of students. Each section has to undertake 8 compulsory courses, and there are no electives. Military training, physical training and the foreign language courses are not included in this number, which are scheduled by other departments apriory. The periods that are occupied by these courses and activities are taken as the unavailable periods of the sections by the Planning Department. The parameters d_{ijk} are formed depending on this data. Table 4.1 on the next page displays this information.

The scheduling cycle in TMA is one week. Total number of available periods is 30 for a scheduling cycle, and 6 for each day. After omitting the unavailable periods for each section, which sum ups to eleven hours, the Planning Department has 19 periods to schedule all 8 courses.

There are 49 teachers who give the courses to the first class of TMA. The teachers are pre-assigned to the courses, and their own departments determine their teaching loads. Under no circumstances, the teaching load of a teacher can exceed 20 periods. Weekly lecture hours (in periods) of courses and the number of teachers pre-assigned to them are shown in table 4.2.

Table 4.1 Unavailable Periods of Sections

Days	Periods	Sections							
		1-5	6-8	9-12	13-15	16-18	19-21	22-26	27-29
1	1	M.T.	M.T.	F.L.	F.L.		P.T.	F.L.	F.L.
	2								
	3					P.T.			
	4								
	5								
	6								
2	1			M.T.	M.T.				P.T.
	2								
	3							P.T	
	4	F.L.	F.L.			F.L.	F.L.		
	5								
	6								
3	1		P.T.			M.T.	M.T.		
	2								
	3	P.T.							
	4								
	5								
	6								
4	1				P.T.			M.T.	M.T.
	2								
	3			P.T.					
	4								
	5								
	6								
5	1								
	2								
	3								
	4								
	5								
	6								

Note: M.T.: Military Training, P.T.: Physical Training, F.T.: Foreign Language.

Table 4.2 Pre-assigned number of teachers and weekly lecture hours of courses.

Courses	1	2	3	4	5	6	7	8	9	10
# of teachers assigned	8	9	6	6	3	3	3	4	4	3
Weekly lecture hours	4	3	2	2	1	1	3	3	3	3

Sections are divided into two main groups. Courses 7,8,9, and 10 are offered to these groups interchangeably. The first group, including 14 sections, takes courses 1-6, 7 and 8, and the second group, the rest 15 sections, is offered courses 1-6, 9 and 10. Daily limit is 2 periods for multi-period courses. So, the courses 1,2,7 and 10 should be split into two days of the week. There are two exceptions. Although courses 8 and 9 are three-hour long, they should not be split, i.e., all hours of these courses must be assigned to the same day.

Since the three-hour long courses 2,7, and 10 cannot be divided into two equal pieces, three dummy courses should be created for these three, in order to fulfill the constraint that they should be split into two days. After creating the dummy courses, all sections have to undertake 10 courses.

Now we are going to formulate the timetabling problem for class I. The assignment procedure is subject to constraints presented in section 3.2. We will indicate the courses not with numbers, but letters as we stated in section 3.1. The notation will be as follows: m for1, f for 2, k for 3, t for 4, i for 5, a for 6, d for 7, tr for 8, b for 9, to for 10, ff for dummy f, df for dummy d, and tof for dummy to.

4.2 Formulation

4.2.1 Variables, Constants and Notations

I = {1,2,…,49} : Set of teachers.

bi = {1,2,3,4} Teachers assigned to course b.

mi = {5,6,…,12} Teachers assigned to course m.

fi = {13,14,…,21} Teachers assigned to course f.

ki = {22,23,…,27} Teachers assigned to course k.

di = {28,29,30} Teachers assigned to course d.

ti = {31,32,…,36} Teachers assigned to course t.

ii = {37,38,39} Teachers assigned to course i.

ai = {40,41,42} Teachers assigned to course a.

toi = {43,44,45} Teachers assigned to course to.

tri = {46,47,48,49} Teachers assigned to course tr.

J = {a1,a2,…,a29,b1,b2,…,b29…} : Set of course-section combinations.

K = {1,2,…,30} : Set of time periods.

pk, sk, ck, pek, cuk : Disjoint subsets of K, where they represent available periods of days of a week in order (i.e., pk for Monday, sk for Tuesday, ck for Wednesday, pek for Thursday, cuk for Friday); and pk □ sk □ ck □ pek □ cuk = K.

DY = {1, 2, 3, 4, 5} : Days of the week.

SC = {1,2…,29} : Set of sections in a class.

CG = {c1,c2,…,c29} : These are the subsets of course-section combinations.

c1 = {m1,f1,k1,t1,i1,a1,ff1,tr1,d1,df1} Example for the first section group (1-14).

c15 = {m15,f15,k15,t15,i15,a15,ff15,b1,to1,tof1} Example for the second section group (15-29).

49 Total number of teachers to assign.

232 Total number of course-section combinations to assign.

nj Total number of teachers can be assigned to a course-section combination j.

mi Total number of course-section combinations that the teacher i can be assigned to.

m_{cg} Total number of course-section combinations can be assigned to a section.

30 Total number of time periods to assign.

6 Total number of daily time periods to assign.

tl(i) Teaching load of a teacher.

V ={1,2,…,vt} Teacher priority group 1. vt is the total number of preferences made by teachers in priority group one.

Y ={1,2,…,yt} Teacher priority group 2. yt is the total number of preferences made by teachers in priority group two.

Z ={1,2,…,zt} Teacher priority group 3. zt is the total number of preferences made by teachers in priority group three.

w1 = 100 The weight for teacher priority group1.

w2 = 10 The weight for teaching loads.

w3 = 1 The weight for teacher priority group 2.

w4 = 1 The weight for teacher priority group 3.

d_i^+, d_i^- Positive and negative deviations from teaching loads, i = 1,2,…,49.

d_v^+, d_v^- Positive and negative deviations from teacher preferences in priority group one, v = 50,51,…,83.

d_y^+, d_y^- Positive and negative deviations from teacher preferences in priority group two, y = 84,85,…,104.

d_z^+,d_z^- Positive and negative deviations from teacher preferences in priority group three, z = 105,106,…,111.

s_j Total number of hours a course-section combination has to be given.

s_{jdy} Total number of hours a course-section combination has to be given on day *dy.*

$$x_{ijk} \text{ Decision variable } = \left\{ \begin{array}{ll} 1 & \text{, if teacher } i \text{ is assigned to a course-} \\ & \text{section combination } j \text{ at time slot } k \\ 0 & \text{, otherwise} \end{array} \right\}$$

$$d_{ijk} \text{ Availability of classes } = \left\{ \begin{array}{ll} 1 & \text{, when a teacher } i \text{ can be assigned to a course-} \\ & \text{section combination } j \text{ at time slot } k, \text{ if the} \\ & \text{related } j \text{ is available at time slot } k \\ 0 & \text{, otherwise} \end{array} \right\}$$

$$q_{jdy} = \left\{ \begin{array}{ll} 1 & \text{, if course-section combination } j \text{ is assigned on day } dy \\ 0 & \text{, otherwise} \end{array} \right\}$$

$$h_{ij} = \left\{ \begin{array}{ll} 1 & \text{, if teacher } i \text{ is giving course } j \\ 0 & \text{, otherwise} \end{array} \right\}$$

4.2.2 Constraints

$$\sum_{i=1}^{nj} \sum_{k=1}^{30} d_{ijk} * x_{ijk} = s_j , \quad \text{for all j.} \qquad (1)$$

$$\sum_{j=1}^{mi} \sum_{k=1}^{30} d_{ijk} * x_{ijk} + d_i^- - d_i^+ = tl(i)_{,} \quad \text{for all i.} \qquad (2)$$

$$\sum_{j=1}^{mi} x_{ijk} \leq 1 , \quad \text{for all i and k.} \qquad (3)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{10} x_{icg(j)k} \leq 1 , \quad \text{for all k and cg} = \{c1,c2,\ldots,c29\}. \qquad (4)$$

$$\sum_{i=1}^{nj} \sum_{k=1}^{6} x_{ijk} = s_{jdy*}q_{jdy} , \quad \text{for all multi-period j and day of the week.} \qquad (5)$$

$$\sum_{i=1}^{nj} \sum_{k=1}^{6} x_{ijk} + x_{ij(d)k} \leq 2 , \quad \text{for j} = \{f,d,to\}, \text{ and each day of the week.} \qquad (6)$$

$$\sum_{j=1}^{mi} x_{ijk} + d_v^- - d_v^+ = 0 , \quad \text{for all i in priority group I, and k that the related} \qquad (7)$$

i is not wiling to teach, v □ V.

$$\sum_{j=1}^{mi} x_{ijk} + d_y^- - d_y^+ = 0 , \quad \text{for all i in priority group II, and k that the related} \qquad (8)$$

i is not wiling to teach, y □ Y.

$\sum_{j=1}^{mi} x_{ijk} + d_z^- - d_z^+ = 0$, for all i in priority group III, and k that the related i is not wiling to teach, z □ Z. (9)

$\sum_{i=1}^{nj} (x_{ijk} + x_{ij(k+w)}) \leq 1$,for all j and the time periods that cannot be consecutive for that j on a day, $(k+w) = k+s_{jdy},\ldots,6$. (10)

$\sum_{k=1}^{30} x_{ijk} = s_j * h_{ij},$ for all i who are giving course j . (11)

$\sum_{i=1}^{nj} h_{ij} = 1$, for all j . (12)

4.2.3 Objective Function

$$\text{Min } z = w_1 \sum_{v=1}^{vt} (d_v^- + d_v^+) + w_2 \sum_{i=1}^{n} (d_i^- + d_i^+) + w_3 \sum_{y=1}^{yt} (d_y^- + d_y^+) + w_4 \sum_{z=1}^{zt} (d_z^- + d_z^+)$$

4.3 Solution Process

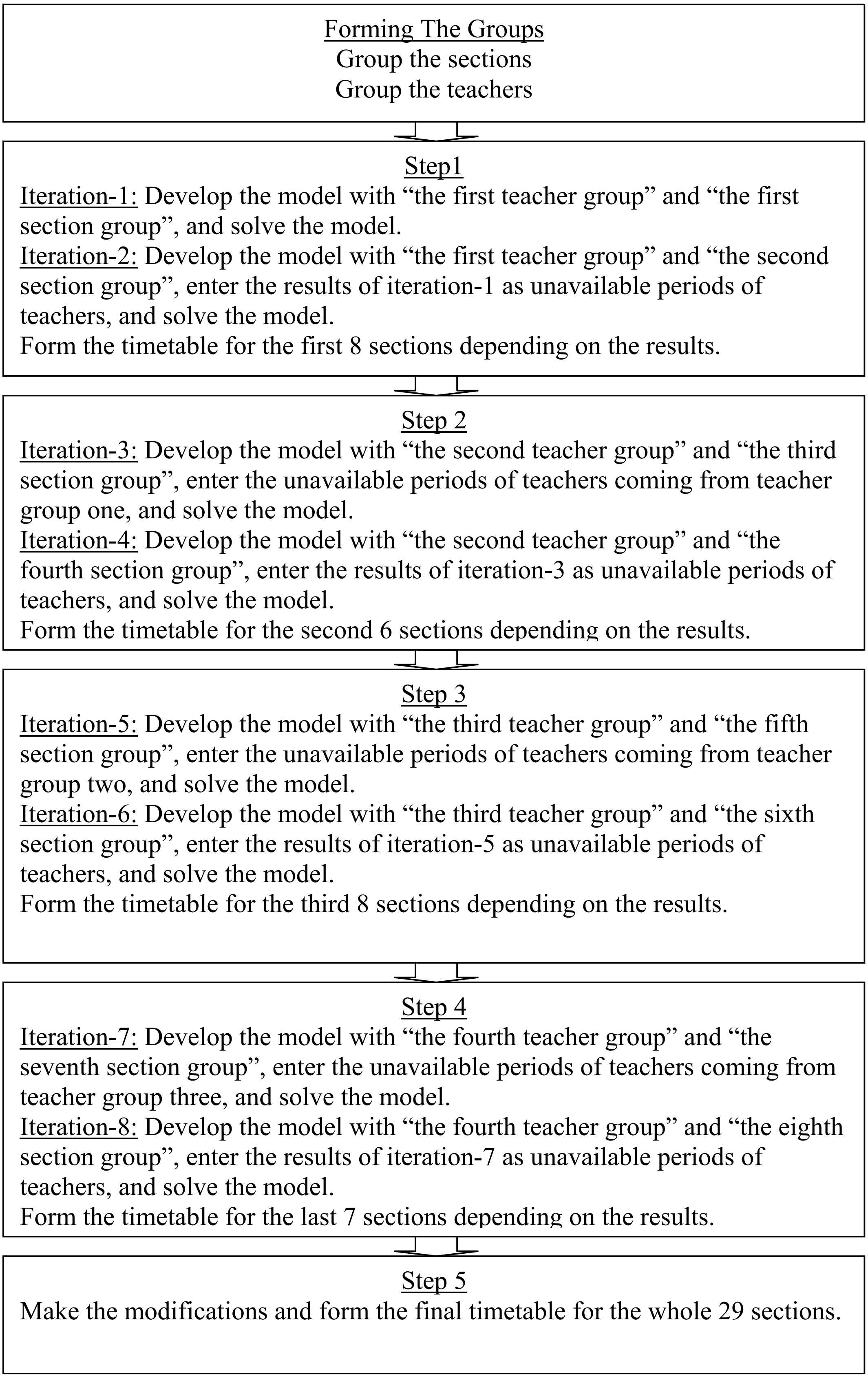
Having modeled the timetabling problem for the first-class of TMA as a goal programming model, we applied an iterative solution heuristic to find a near-optimal solution. Because the problem is still too large to be solved at once on a PC, with 36540 decision variables and 9207 constraints. Even when the constraints of teaching loads and teacher preferences are omitted, the number of constraints is decreased only by 112. We also ran the model with the whole 29 sections, 14 sections, and 8 sections. But in each run, we couldn't reach to a solution within the iteration limits in less than 48 hours. Flowchart of the solution process is shown in figure 4.1 on the next page.

4.4 Test Runs

We tested the model with five problem sets. We tried to find out that how our model would react to special cases of our problem that are defined and discussed below.

Problem Set 1: The first problem had 12 teachers, 40 courses, and 4 sections. The model had no preference constraints, and the objective function was reduced to only one priority, that is to minimize teaching loads. Our aim was to see whether we could get a feasible solution when we were free of preference goal constraints. We reached to an optimal solution in almost five minutes.

Problem Set 2: This problem set had the same number of teachers, courses, and sections as in problem set 1. There were no preference constraints again, and we assigned average course teaching loads to the teachers. This activity mostly reduced the teaching loads, which led to stricter conditions. It took more than thirty minutes to reach to an optimal solution this time.



Forming The Groups

Group the sections

Group the teachers

Step1

Iteration-1: Develop the model with “the first teacher group” and “the first section group”, and solve the model.

Iteration-2: Develop the model with “the first teacher group” and “the second section group”, enter the results of iteration-1 as unavailable periods of teachers, and solve the model.

Form the timetable for the first 8 sections depending on the results.

Step 2

Iteration-3: Develop the model with “the second teacher group” and “the third section group”, enter the unavailable periods of teachers coming from teacher group one, and solve the model.

Iteration-4: Develop the model with “the second teacher group” and “the fourth section group”, enter the results of iteration-3 as unavailable periods of teachers, and solve the model.

Form the timetable for the second 6 sections depending on the results.

Step 3

Iteration-5: Develop the model with “the third teacher group” and “the fifth section group”, enter the unavailable periods of teachers coming from teacher group two, and solve the model.

Iteration-6: Develop the model with “the third teacher group” and “the sixth section group”, enter the results of iteration-5 as unavailable periods of teachers, and solve the model.

Form the timetable for the third 8 sections depending on the results.

Step 4

Iteration-7: Develop the model with “the fourth teacher group” and “the seventh section group”, enter the unavailable periods of teachers coming from teacher group three, and solve the model.

Iteration-8: Develop the model with “the fourth teacher group” and “the eighth section group”, enter the results of iteration-7 as unavailable periods of teachers, and solve the model.

Form the timetable for the last 7 sections depending on the results.

Step 5

Make the modifications and form the final timetable for the whole 29 sections.

Figure 4.1 Flowchart of The Solution Process

Problem Set 3: We imposed preference constraints to the problem set 2 for two teachers. Surprisingly we had to wait only a minute to reach to an optimal solution.

Problem Set 4: This time we had 8 teachers with increased teaching loads, 40 courses and 4 sections. We included preference constraints for two teachers. Only one teacher for each course is assigned, and it took three minutes to reach to the best possible solution. Three preferences out of six were violated in the final solution.

Problem Set 5: Finally in the fifth, we again had 12 teachers, 40 courses and 4 sections. We imposed on the model very strict conditions. Preference constraints for almost every teacher were included, and conflicting conditions were created. Thirty-six preferences out of fifty-eight were in priority group 1. We reached to a near-optimal solution within the iteration limits, in which seven preferences in priority group1 and three in priority group 2 were violated.

Table 4.3 Results of The Test Runs

Problem Sets	**Time**	**Solution Quality**
Problem Set 1	5 min.	Optimal
Problem Set 2	30 min.	Optimal
Problem Set 3	1 min.	Optimal
Problem Set 4	3 min.	Optimal
Problem Set 5	1 hr.	Near-optimal

It is easier to reach optimal solutions when the problem is small and the conditions are softer, and it takes more time to find solutions when the goal constraints get tighter. It took sometimes more time to get a solution even when there was only one preference constraint. And it took really more time to reach to a solution, especially in latter steps, when the teaching loads and the availability of the teachers were declined. So you can value that how big the problem is, with 49 teaching loads and 63 preference goal constraints.

4.5 Results of The Solution Process

We solved all eight models, two in each step, on a Unix workstation by using Mixed Integer Programming (MIP) solver in GAMS. We made one-hour runs for each model except for the one in step three iteration five, which required a three-hour run. In the solution outputs, the periods where decision variable $x_{ijk} = 1$ are the periods that a teacher is assigned to a section-course combination.

The output files of eight models were transformed to timetables for the first-class of TMA by using MATLAB 6.5. This procedure provided us with great advantages. The timetables are generated in shorter times compared to the ones prepared by hand. We can generate timetables according to the criteria, whatever we choose, in seconds. We can generate timetables for any section or section groups, teacher or teacher groups, teachers exceeding teaching loads etc. We can use them as check lists and easily examine the timetables and see whether there is an assignment made to an unavailable time block. Our capabilities are restricted if we do all these activities by hand, and the possibility that we make mistakes is higher.

We used the heuristic procedure to decrease the problem to a solvable size. In each iteration, we found feasible solutions, and optimal ones for some models. After all four steps, we came up with a timetable that satisfies all of the strict requirements of TMA, and minimizes the objective function. We didn't intervene the process between the steps, and made no manipulations. Having completed the all four steps, we got a better timetable by making some modifications, decreasing the objective function values to lower levels in step 5.

For the first model, an optimal objective function value of 0 was found in less than ten minutes and in 88051 iterations. For the second model, a near-optimal objective function value of 103 was found. For both of the models three and four, optimal solutions were found in less than ten minutes. For model five, again an optimal solution was found in a three-hour run. The objective function value for model six was 40, and found in 230139 iterations. And finally the objective function values for models seven and eight were 7 and 854 respectively.

After completing four steps, we can summarize the solutions as follows to get the picture more clearly:

Eleven preference made in priority group 1, nine preference made in priority group 2, and three preference made in priority group 3 were violated, and fourteen teachers were assigned courses below or above their teaching loads. Finally after modifications in step 5, the numbers declined to only one in group 1, and seven in group 2. The number increased in group 3 by two, but only two instructors whose teaching loads are violated were left. Table 4.3 shows this information.

Table 4.4 **Results of The Solution Process**

	Preference Violations				
	Priority group 1	Priority group 2	Priority group 3	Total	Teachers overload
After step 4	11	9	3	23	14
After step 5	1	7	5	13	1

Since we worked on the same problem in Üstünel (2001), we want to make a comparison here. Although, we both tried to produce solutions for the same problem, we couldn't make an exact comparison of the models using the same input data, due to time limitations. Their model was applied to a different class, and had 39,745 binary variables and 110,316 constraints, whereas ours had 36540 decision variables and 9207 constraints. Our model has a lot less constraints than their model. One major advantage we seem to have is that we always have a chance to find a feasible solution since we used goal programming, whereas they might face infeasible solutions because of the mathematical model. The modeling approach in both studies are different. They used penalties in the objective function. We formed four priorities in the objective function, and assigned weights to them according to their importance determined by the administration, since we used a non-preemptive approach. So the models might look as if they were the same from this aspect. We wrote the constraints in a compact manner, which reduced the total number. But we had to create some dummy variables to fulfill some requirements. Because of the size of the problem, we had to use a heuristic solution process similar to theirs. We both reached to near-optimal solutions. In the final solution, their assignment has only one instant of a non-preferred time period, and also ten assignments to the moderately preferred time periods, by four teachers. Since the ratio of non-preferred assignments was low and the time consumed by this model was shorter compared to the hand made timetabling, they defined the final timetable as “a good timetable” both for the teachers and the timetablers. In our model, we had twenty-three non-preferred assignments after step four. Having made the modifications in step five, we reduced the number to thirteen. We reached to solutions definitely in short times compared to the hand made timetabling. We can also define our model as “a good timetable”, for the teachers and especially for the timetablers. By solving a problem with both models, which has the same requirements, it would be possible to make an exact comparison between them.

CHAPTER 5

CONCLUSION AND FUTURE DIRECTIONS

We offered a goal programming model to solve the timetabling problem in Turkish Military Academy, and came up with a timetable that is feasible and near optimal. We used a non-preemptive approach. We ran the model with five different problem sets to test it and see how it would react to conditions of different problems. Since we decomposed the problem, we had to group the teachers and the sections. The sections were grouped according to their unavailable periods, and the teachers were grouped according to their teaching loads and their unavailable time periods to match the section groups formed. But some groups had more goal constraints and it took more time to reach to an optimal or near-optimal solution, and more preferences were violated. We tried to overcome this problem in the modification phase, and were successful to decrease these violations to lower levels. But it does not guarantee that we can get better results all the time. Some other rules in grouping operations can be improved. If a solution procedure that solves the model at once can be found and applied, then there will be no need for all these grouping activities.

We did not interrupt the solution process and waited for the end of the whole procedure to make modifications, to see if we can make improvements. The opportunities should be investigated, whether making interactive modifications can lead to better results.

For future research, a user interface can be written to make the program more user friendly. The efforts to find new solution processes that will lead to reach to better results in relatively shorter times should be continued. Since the size of the problem is too big, more emphasis should be put on the use of heuristic approaches, and these methods should be investigated rather than mathematical ones.

SELECT BIBLIOGRAPHY

Abbas, A., and E.P.K. Tsang. 2004. "Software Engineering Aspect of Constraint-Based Timetabling—A Case Study," *Information and Software Technology* 46(6): 359-372.

Abramson, D. 1991. "Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms," *Management Science* 37(1): 98-113.

Abramson, David, Mohan Krishnamoorthy, and Henry Dang. 1999. "Simulated Annealing Cooling Schedules for the School Timetabling Problem," *Asia-Pacific Journal of Operational Research* 16:1-22.

Akkoyunlu, E.A. 1973. "A Linear Algorithm for Computing the Optimum University Timetable," *The Computer Journal* 16(4): 347-350. In Daskalaki, S., T. Birbas, and E. Housos. 2004. "An Integer Programming Formulation for a Case Study in University Timetabling," *European Journal of Operational Research* 153: 117-135.

Alvarez-Valdes, Ramon, Enric Crespo, and Jose M. Tamarit. 2002. "Design and Implementation of A Course Scheduling System Using Tabu Search," *European Journal of Operational Research* 137: 512-523.

Asratian, A.S., and D. de Werra. 2002. "A Generalized Class-Teacher Model for Some Timetabling Problems," *European Journal of Operational Research* 143: 531-542.

Badri, Masood A. 1996. "A Two Stage Multi-Objective Scheduling Model for [faculty-course-time] Assignment," *European Journal of Operational Research* 94: 16-28.

Badri, Masood A., Donald L. Lewis, Donna F. Davis, and John Hollingsworth. 1998. "A Multi-Objective Course Scheduling Model: Combining Faculty Preferences for Courses and Times," *Computers and Operations Research* 25(4): 303-316.

Baker, Kenneth R., Michael J. Magazine, and George G. Polak. 2002. "Optimal Block Design Models for Course Timetabling," *Operations Research Letters* 30: 1-8.

Birbas, T., S. Daskalaki, and E. Housos. 1997. "Timetabling for Greek High Schools," *The Journal of the Operational Research Society* 48(12): 1191-1200.

Botsali, Ahmet Reha. 2000. "A Timetabling Problem: Constraint and Mathematical Programming Approaches." Unpublished master's thesis. Bilkent University, Ankara.

Brailsford, Sally C., Chris N. Potts, and Barbara M. Smith. 1999. "Constraint Satisfaction Problems: Algorithms and Applications," *European Journal of Operational Research* 119: 557-581.

Burke, E.K., and Elliman D.G. 1994. "A University Timetabling System Based on Graph Colouring and Constraint Manipulation," *Journal of Research on Computing in Education* 27(1): 1-18.

Burke, E.K., D.G. Elliman, and R.F. Weare. 1995. "The Automation of the Timetabling Process in Higher Education," *Journal of Educational Technology Systems* 23(4): 257-266.

Burke, E.K., and Sanja Petrovic. 2002. "Recent Research Directions in Automated Timetabling," *European Journal of Operational Research* 140: 266-280.

Burke, E.K., G. Kendall, and E. Soubeiga. 2003. "A Tabu-Search Hyperheuristic for Timetabling and Rostering," *Journal of Heuristics* 9: 451-470.

Carrasco, M.P., and M.V. Pato. 2004. "A Comparison of Discrete and Continuous Neural Network Approaches to Solve the Class/Teacher Timetabling Problem," *European Journal of Operational Research* 153: 65-79.

Carter, M.W. 1986. "A Survey of practical applications of Examination Timetabling Problems," *Operations Research* 24(2): 193-201.

Colorni, Alberto, Marco Dorigo, and Vittorio Maniezzo. 1998. "Metaheuristics for High School Timetabling," *Computational Optimization and Applications* 9: 275-298.

Costa, Daniel. 1994. "A Tabu Search Algorithm for Computing an Operational Timetable," *European Journal of Operational Research* 76: 98-110.

Daskalaki, S., T. Birbas, and E. Housos. 2004. "An Integer Programming Formulation for a Case Study in University Timetabling," *European Journal of Operational Research* 153: 117-135.

de Werra, D. 1985. "An Introduction to Timetabling," *European Journal of Operational Research* 19: 151-162.

-----. 1997. "The Combinatorics of Timetabling," *European Journal of Operational Research* 96: 504-513.

Deris, Safaai, Sigeru Omatu, and Hiroshi Ohta. 2000. "Timetable Planning Using the Constraint-Based Reasoning," *Computers and Operations Research* 27: 819-840.

Dimopoulou, M., and P. Miliotis. 2001. "Implementation of a University Course and Examination Timetabling System," *European Journal of Operational Research* 130: 202-213.

-----. 2004. "An Automated University Course Timetabling System Developed in a Distributed Environment: A Case Study," *European Journal of Operational Research* 153: 136-147.

Dowsland, Kathryn A. 1990. "A Timetabling Problem in Which Clashes are Inevitable," *The Journal of the Operational Research Society* 41(10): 907-918.

Karataş, Koray. 1996. " An Information System Based on System Analysis Approach For Course Scheduling in Universities." Unpublished master's thesis. Bilkent University, Ankara.

Lawrie, N.L. 1969. "An Integer Linear Programming Model of A School Timetabling Problem," *The Computer Journal* 12: 307-316. In Daskalaki, S., T. Birbas, and E. Housos. 2004. "An Integer Programming Formulation for a Case Study in University Timetabling," *European Journal of Operational Research* 153: 117-135.

Schaerf, A. 1999. "A Survey of Automated Timetabling," *Artificial Intelligence Review* 13: 87-127.

Schmidt, G., and T. Strohlein. 1979." Timetable Construction-An Annotated bibliography," *The Computer Journal* 23(4): 307-316. In Daskalaki, S., T. Birbas, and E. Housos. 2004. "An Integer Programming Formulation for a Case Study in University Timetabling," *European Journal of Operational Research* 153: 117-135.

Schniederjans, Marc J., and Gyu Chan Kim. 1987. "A Goal Programming Model to Optimize Departmental Preference in Course Assignments," *Computers and Operations Research* 14(2): 87-96.

Smith, Kate A., David Abramson, and David Duke. 2003. "Hopfield Neural Networks for Timetabling: Formulations, Methods, and Comparative Results," *Computers and Industrial Engineering* 44:283-305.

Üstünel, A. Cem. 2001. "A Mathematical Model For Planning of Class Timetables: An Application in Turkish Military Academy." Unpublished master's thesis. Yeditepe University, İstanbul.

Valouxis, Christos, and Efthymios Housos. 2003. “Constraint Programming Approach for School Timetabling,” *Computers and Operations Research* 30: 1555-1572.

Welsh, D.J.A., and M.B. Powell. 1967. “An Upper Bound to the Chromatic Number of A Graph and Its Application to Timetabling Problem,” *The Computer Journal* 10: 85-86. In Daskalaki, S., T. Birbas, and E. Housos. 2004. “An Integer Programming Formulation for a Case Study in University Timetabling,” *European Journal of Operational Research* 153: 117-135.

Yu, Enzhe, and Ki-Seok Sung. 2002. “A Genetic Algorithm for a University Weekly Courses Timetabling Problem,” *International Transactions in Operational Research* 9: 703-717.

APPENDIX A TIMETABLE FOR INSTRUCTORS AFTER STEP 4

		Monday						Tuesday						Wednesday						Thursday						Friday					
course	inst.	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
course 1	1				X	X	X							X	X	X	X	X	X									X	X	X	
	2				X	X	X									X	X	X								X	X	X	X	X	X
	3				X	X	X							X	X	X	X	X	X				X	X	X				X	X	X
	4	▨	▨	▨	X	X	X	▨	▨	▨	▨	▨	▨													X	X	X			
course 2	5							X	X					X	X			X	X		X	X		X	X	X	X	X	X	X	X
	6							X	X					X	X			X	X	X	X	X	X			X	X	X	X	X	X
	7					X	X							X	X									X	X					X	X
	8					X	X	▨	▨	▨	▨	X	X	X	X	X	X	X	X					▨	▨	X	X	X	X	X	X
	9					X	X	▨	▨	▨	▨	X	X	X	X	X	X	X	X					▨	▨	X	X	X	X	X	X
	10					X	X					▨		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	11	X	X																	X	X	X	X							X	X
	12			X	X	X	X		X	X										X	X			X	X	X	X	X	X	X	X
course 3	13						X	X																X	X			X	X		
	14			X		X	X	X		X										X	X	X				X	X	X	X		
	15				X									X	X			X	X	X	X			X	X	X			X		X
	16				X			X					X	X	X	X		X	X								X	X		X	X
	17													X	X	X		X	X	X	X	X	X		X	X	X				
	18					▨	▨	X	X	X										X	X	X	X	X	X	X	X		X		
	19				X			X	X	X	X			X					X	▨	▨	▨	▨	▨	▨	X	X		X	X	X
	20				X																					X	X				
	21				X											X	X		X							X	X				
course 4	22	X	X	X	X			X	X											X	X			▨		X	X	X	X		
	23	▨	▨	▨	▨				X	X				X	X	X	X	X	X					X	X	X	X	X	X	X	X
	24					X	X	X	X			X	X	X	X			X	X							X	X	X	X	X	X
	25					X	X																	X	X	X	X				
	26	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨					X	X
	27	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨			X	X	X	X	▨	▨	▨	▨	▨	▨	▨	▨	X	X	▨	▨

APPENDIX A TIMETABLE FOR INSTRUCTORS AFTER STEP 4 (cont'd)

		Monday						Tuesday						Wednesday						Thursday						Friday					
course	inst.	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
course 5	28							X	X	X								X	X	X		X	X					X	X	X	X
	29							X	X	X						X				X	X	X	X	X	X	X	X	X		X	X
	30					X		X	X	X										X	X	X	X	X	X	X	X	X	X	X	
course 6	31									X	X	X	X	X	X	X	X	X	X							X	X	X	X	X	X
	32								X	X				X	X	X	X				X	X		X	X	X	X	X	X	X	X
	33	X	X			X	X													X	X	X	X	X	X	X	X			X	X
	34					X	X															X	X					X	X		
	35																			X	X			X	X						
	36															X	X														
course 7	37				X	X			X			X	X	X	X		X					X							X	X	
	38				X	X	X	X	X	X							X	X										X			X
	39									X										X	X			X			X	X		X	X
course 8	40							X		X										X		X	X			X			X	X	X
	41				X		X	X	X						X	X	X									X		X			X
	42				X			X	X	X		X				X							X	X						X	X
course 9	43				X	X	X							X	X			X	X						X	X	X			X	X
	44				X	X	X	X	X			X	X	X	X	X	X	X	X							X	X	X	X		X
	45					X	X			X	X	X	X	X	X	X	X	X	X								X	X	X		
course 10	46							X	X	X							X	X	X	X	X	X	X	X	X						
	47				X	X	X	X	X	X							X	X	X				X	X	X			X	X	X	
	48																						X	X	X	X	X	X	X	X	X
	49																			X	X	X	X	X	X						

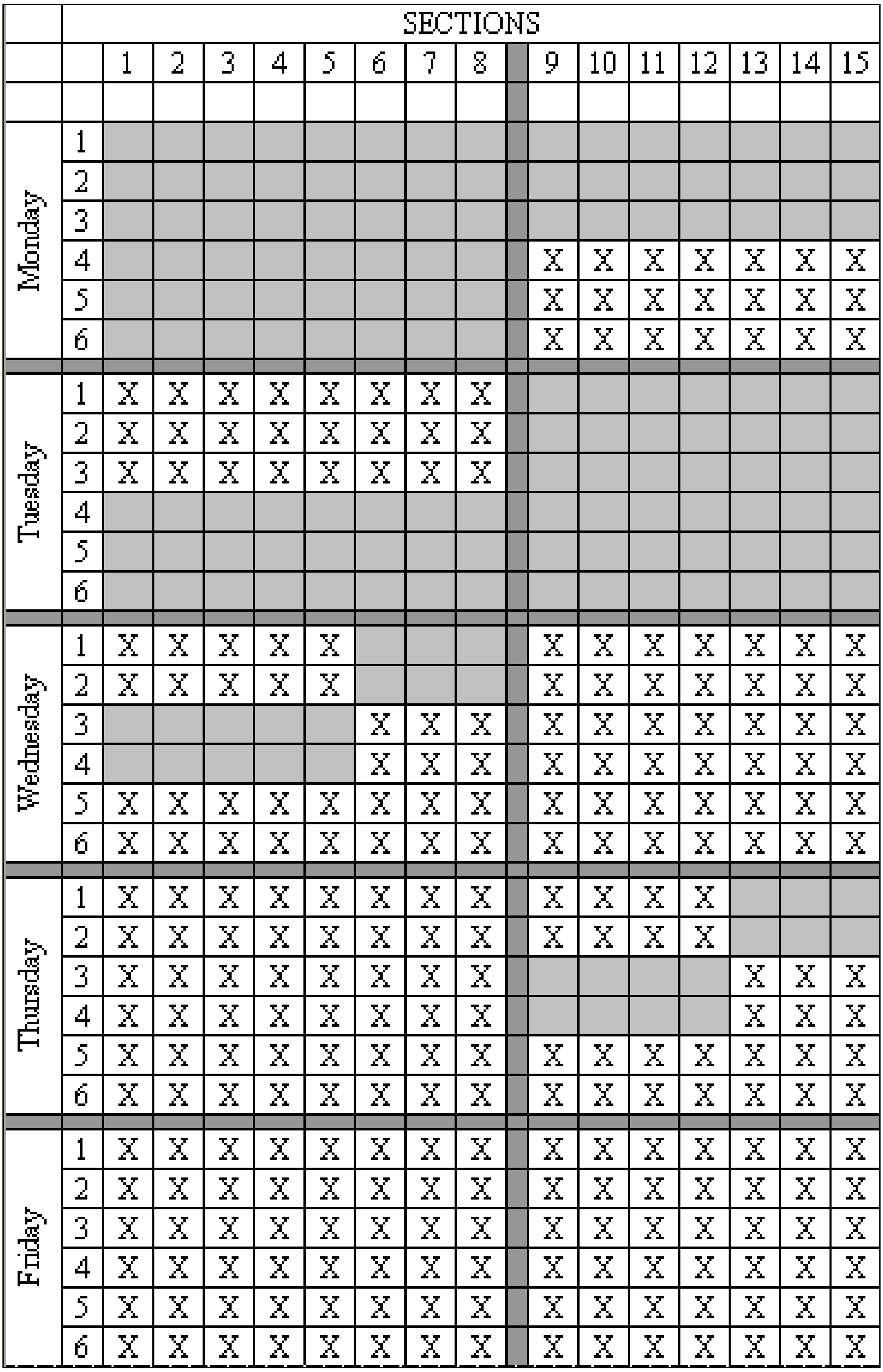
APPENDIX B TIMETABLE FOR INSTRUCTORS AFTER STEP 5

		Monday						Tuesday						Wednesday						Thursday						Friday					
course	inst.	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
course 1	1				X	X	X							X	X	X	X	X	X									X	X	X	
	2				X	X	X									X	X	X								X	X	X	X	X	X
	3				X	X	X							X	X	X	X	X	X				X	X	X				X	X	X
	4	▨	▨	▨	X	X	X	▨	▨	▨	▨	▨	▨													X	X	X			
course 2	5							X	X					X	X			X	X		X	X		X	X	X	X	X	X	X	X
	6							X	X					X	X			X	X	X	X	X	X			X	X	X	X	X	X
	7					X	X							X	X									X	X					X	X
	8					X	X	▨	▨	▨	▨	X	X	X	X	X	X	X	X					▨	▨	X	X	X	X	X	X
	9	X	X			X	X	▨	▨	▨	▨	X	X	X	X	X	X	X	X	X	X			▨	▨	X	X	X	X	X	X
	10					X	X					▨		X	X	▨	▨	▨	▨	X	X	X	X	X	X	X	X	X	X	X	X
	11															X	X	X	X			X	X							X	X
	12			X	X	X	X		X	X										X	X			X	X	X	X	X	X	X	X
course 3	13				X			X																X	X			X	X		
	14			X		X	X	X		X										X	X	X				X	X	X	X		
	15				X		X							X	X			X	X	X	X			X	X	X	X	X	X		X
	16				X			X					X	X	X	X		X	X								X	X		X	X
	17													X	X	X		X	X	X	X	X	X		X	X	X				
	18					▨	▨	X	X	X										X	X	X	X	X	X	X	X		X		
	19				X			X	X	X	X			X					X	▨	▨	▨	▨	▨	▨	X	X		X	X	X
	20				X																					X	X				
	21																		X							X	X				
course 4	22	X	X	X	X			X	X											X	X			▨		X	X	X	X		
	23	▨	▨	▨	▨				X	X				X	X	X	X	X	X					X	X	X	X	X	X	X	X
	24					X	X	X	X			X	X	X	X			X	X							X	X	X	X	X	X
	25					X	X																	X	X	X	X				
	26	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨			X	X	X	X
	27	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨			X	X	X	X	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

APPENDIX B TIMETABLE FOR INSTRUCTORS AFTER STEP 5 (cont'd)

		Monday						Tuesday						Wednesday						Thursday						Friday					
course	inst.	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
course 5	28							X	X	X								X	X	X		X	X					X	X	X	X
	29							X	X	X						X				X	X	X	X	X	X	X	X	X		X	X
	30					X		X	X	X										X	X	X	X	X	X	X	X	X	X	X	
course 6	31									X	X	X	X			X	X	X	X			X	X			X	X	X	X	X	X
	32								X	X				X	X	X	X				X	X		X	X	X	X	X	X	X	X
	33					X	X									X	X			X	X	X	X	X	X	X	X			X	X
	34	X	X			X	X																								
	35																			X	X			X	X						
	36													X	X	X	X														
course 7	37				X	X			X			X	X	X	X		X					X		X						X	
	38				X	X	X	X	X	X							X	X										X			X
	39									X										X	X			X			X	X		X	X
course 8	40							X		X										X		X	X			X			X	X	X
	41				X		X	X	X						X	X	X									X		X			X
	42				X			X	X	X		X				X							X						X	X	X
course 9	43					X	X			X	X			X	X	X		X	X					X	X	X			X	X	X
	44				X	X	X	X	X			X	X	X	X	X	X	X	X							X	X	X	X		X
	45				X	X	X					X	X	X	X		X	X	X	X							X				
course 10	46							X	X	X							X	X	X				X	X	X						
	47				X	X	X	X	X	X							X	X	X				X	X	X			X	X	X	
	48																			X	X	X	X	X	X	X	X	X	X	X	X
	49																			X	X	X	X	X	X						

APPENDIX C TIMETABLE FOR SECTIONS



		SECTIONS														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Monday	1															
	2															
	3															
	4									X	X	X	X	X	X	X
	5									X	X	X	X	X	X	X
	6									X	X	X	X	X	X	X
Tuesday	1	X	X	X	X	X	X	X	X							
	2	X	X	X	X	X	X	X	X							
	3	X	X	X	X	X	X	X	X							
	4															
	5															
	6															
Wednesday	1	X	X	X	X	X				X	X	X	X	X	X	X
	2	X	X	X	X	X				X	X	X	X	X	X	X
	3						X	X	X	X	X	X	X	X	X	X
	4						X	X	X	X	X	X	X	X	X	X
	5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Thursday	1	X	X	X	X	X	X	X	X	X	X	X	X			
	2	X	X	X	X	X	X	X	X	X	X	X	X			
	3	X	X	X	X	X	X	X	X					X	X	X
	4	X	X	X	X	X	X	X	X					X	X	X
	5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Friday	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

APPENDIX C TIMETABLE FOR SECTIONS (cont'd)

	SECTIONS													
	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Monday	X	X	X											
	X	X	X											
				X	X	X								
				X	X	X	X	X	X	X	X	X	X	X
	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Tuesday	X	X	X	X	X	X	X	X	X	X	X			
	X	X	X	X	X	X	X	X	X	X	X			
	X	X	X	X	X	X						X	X	X
												X	X	X
							X	X	X	X	X	X	X	X
							X	X	X	X	X	X	X	X
Wednesday							X	X	X	X	X	X	X	X
							X	X	X	X	X	X	X	X
							X	X	X	X	X	X	X	X
							X	X	X	X	X	X	X	X
							X	X	X	X	X	X	X	X
							X	X	X	X	X	X	X	X
Thursday	X	X	X	X	X	X								
	X	X	X	X	X	X								
	X	X	X	X	X	X								
	X	X	X	X	X	X								
	X	X	X	X	X	X								
	X	X	X	X	X	X								
Friday	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X	X	X	X	X	X	X	X	X	X	X	X	X	X