T.C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

INVESTIGATING THE USABILITY OF INTEGER PROGRAMMING FOR THE SCHEDULING PROCESS IN AN EDUCATIONAL INSTITUTE

Bayhan Temur

B.S. Teaching Mathematics

(141100919930009)

THESIS
FOR THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING PROGRAMME

SUPERVISOR
Assist. Prof. MERT DEMIR

ISTANBUL 2006

T.C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

INVESTIGATING THE USABILITY OF INTEGER PROGRAMMING FOR THE SCHEDULING PROCESS IN AN EDUCATIONAL INSTITUTE

Bayhan Temur

B.S. Teaching Mathematics

(141100919930009)

THESIS
FOR THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING PROGRAMME

SUPERVISOR
Assist. Prof. MERT DEMIR

ISTANBUL 2006

MARMARA UNIVERSITY
THE INSTITUTE FOR
GRADUATED STUDIES IN PURE AND APPLIED SCIENCES

ACCEPTANCE AND APPROVAL DOCUMENT

INVESTIGATING THE USABILITY OF INTEGER PROGRAMMING FOR THE SCHEDULING PROCESS IN AN EDUCATIONAL INSTITUTE

Established committee listed below, on..................and...................by the *INSTITUTE FOR GRADUADTE STUDIES PURE AND APPLIED SCIENCES'* Executive Committee, have accepted Mr. Bayhan Temur's Master of Science Thesis, titled as ". Investigating The Usability Of The Integer Programming For The Scheduling Process In An Educational Institute" in Industrial Engineering.

COMMITTEE

Advisor : Asist.Prof.Dr. Mert Demir

Member : Prof.Dr. M.Akif Eyler

Member : Assoc.Prof.Dr. Borahan Tümer

Date of thesis'/ dissertation's defense before the committee: 18.07.2006

APPROVAL

Mr. Bayhan Temur has satisfactorily completed the requirements for the degree of Master of Science in Industrial Engineering at Marmara University.

Mr. Bayhan Temur is eligible to have the degree awarded at our convocation on ....Diploma and transcripts so noted will be available after that date.

Istanbul **DIRECTOR**

ACKNOWLEDGEMENTS

The person that I have to express my gratefulness is my thesis supervisor Assist. Prof. Mert Demir. He provided me with great support in my study and valuable feedback. My ambiguous thoughts evolved into an original study with his continuous effort and endless patience. Thank you for believing in me.

I would like to thank Lale Kara and Hans Peter Neuschmid, Principles of ALEV Schools, for allowing me time to study.

I would like to thank Tayfun Isguden, vice principle of ALEV Schools, for helping for receiving the data of school.

Finally I would like to express my gratitude to my wife Zeynep and our son Levent for their support and patience.

June/2006
Bayhan Temur

CONTENTS

ÖZET

EĞİTİM KURUMLARINDA DERS ÇİZELGELEME SÜRECİNDE TAMSAYILI PROGRAMLAMANIN KULLANILABİLİRLİĞİNİN ARAŞTIRILMASI

Günümüzde eğitim sektöründe ders çizelgeleme problemi, dönem başlarında idareciler için önemli bir yük getirmekte, çoğunda çizelgeleme el ile yapılmaktadır. Bu da eğitim sürecini olumsuz etkilemektedir. Bazı firmaların hazırladığı programlar da ya yeterli olmamakta ya da pahalıya mal olmaktadır. Okullarımızda bilgisayar bulunduğuna göre bir tamsayılı programlama yazılımı bulunan her kurum, kolaylıkla çizelgeleme yapabilecektir. Bu çalışmanın amacı, kurumlarımızın böyle bir olanaktan faydalanmasını sağlayacak bir rehber program sunmaktır.

Bu amaçla, bir eğitim kurumunun elle yaptığı bir çizelgeleme örneği temel olarak alınmıştır. Bu çizelgede geçen öğretmenler, sınıflar, branşlar, ders süreçleri günler ve saatler veri olarak değerlendirilmiştir.

Problemin matematiksel bir modeli oluşturulmuştur.Bu modelde, block ders kısıtları, öğretmenlerin ders ve gün istekleri ile diğer bazı yönetimsel istekler dikkate alınmıstır. Bir tamsayılı modelleme programı olan GAMS ve CPLEX çözücü olarak kullanılmıştır. Örnek bir zaman çizelgesi oluşturulmuştur. Programa veri girişi ve giriş kuralları hakkında bilgi verilmiştir.

Yapılan program, ilgili kurumda uygulanmış ve günler alan zaman çizelgeleme işinin kısa bir sürede tamamlandığı görülmüştür.

Anahtar Sözcükler: İlköğretim okullarında ders çizelgeleme, tamsayılı programlama

Haziran / 2006 **Bayhan Temur**

ABSTRACT

INVESTIGATING THE USABILITY OF INTEGER PROGRAMMING FOR THE SCHEDULING PROCESS IN AN EDUCATIONAL INSTITUTE

Nowadays in education sector, course-scheduling problem, at the beginning of year to reward administrators, mostly scheduling is performed manually. It adversely affects the education process. Commercial software is not adequate or expensive. The schools have computers hence anyone with integer programming software; can easily schedule courses on it. The aim of this study is to develop a program to prepare a guide for the schools to avail these.

For this reason, a scheduling sample that is prepared manually is taken from an institute as the main data. At the schedule, teachers, classes, branches, days and lessons are taken as the database.

The mathematical model is prepared. The problem has been modeled in GAMS and solved by the CPLEX solver. The model has accommodated block lessons, preferences of educators about course timings and some other administrative constraints. A sample course scheduling was constituted. Data requirements and their formats have been identified.

It has been applied in the institute, and the scheduling, which needed days to get over, has been completed very quickly.

Keywords: Course timetabling in first-grade education, integer programming

June / 2006 **Bayhan Temur**

CLAIM FOR ORIGINALITY

The main originality of this study is the first study about a timetabling process of Turkish school, which is an element of M.E.B. and private primary and secondary level school, in the literature. The sample school is a private school that in Turkey gathering data is highly problematic because of their usage of curriculum and extra-disappeared curriculum affairs.

June / 2006 **Assist.Prof. Mert Demir** **Bayhan Temur**

LIST OF SYMBOLS / ABBREVATIONS

a : Assignment of teacher, classroom and courses

ASAP : Automated Scheduling. Optimization and Planning Research Group

b : Auxiliary variable for controlling block periods

bs : The block section parameter

Cmp : Computer

CP : Constraint Programming

CRM : Culture of Religion and Morality Information

CTc : Class Teaching

Ctz : Citizenship

Eng : English

Gdn : Guidance

Grm : German

HTR : History of Turkish Repuplic

I : The set of teachers

IP : Integer Programming

IT : Information Technology

K : The set of classes

L : The set of courses

LP : Linear Programming

Msc : Music

MIP : Mixed-Integer Programming

M.E.B. : Milli Eğitim Bakanlığı, Ministry of National Education

Mth : Mathematics

M : The set of days

N : The set of days

OR : Operations Research

Pnt : Paint

SGr : Study German

Scn : Science

SIn : Socail Information

Spr : Sport or Physical Education

Std : Study

Trf : Traffic

Trk : Turkish

TTMN : TimeTabling Markup Language

WEd : Work Education

x : Timetable binary variables for automated assignment

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

At the beginning of education process, course scheduling is a major administrative activity for variety of institutions. A course-scheduling problem of assigning of a number of teachers into a limited number of time periods.

In this study, we will be concentrated on the school-timetabling problem. (Werra, 1985) During the last twenty years many contribution related to timetabling have appeared and it will probably continue with the same rate for years. One reason for this may be the huge variety of problems, which are included in the field of timetabling; another reason is the fact that education methods are changing, so that the models have to be modified. Computing facilities are now available in most schools and as a consequence, the approach to timetabling has to take this phenomenon into account. This means in particular that interactive methods are now becoming more important and, furthermore, they have to be adapted to microcomputers.

Such problems can be divided into three main categories: (Drexl, Salewski, Frank, 1997) School timetabling, academic course timetabling and other timetabling. In this study, it will be concentrated on a school timetabling problem, in other words class-teacher timetabling, in the case that a class will consist of a set of students who follow the same course program (Aubin and Ferland 1989). The timetable must be satisfying compactness constraints for pupils and teachers. In addition, the large teaching units have to be distributed over week in order to get acceptable timetables. Other optional issues might be accommodated in specific situations. Academic course timetabling considers course scheduling problem, which does not allow take care of rooms of different sizes, but which considers the professors preferences for periods (Tripathy, 1980). Other timetabling models; no general guideline can be established. It depends on the specific timetabling problem under consideration which optional issues become necessary. The far most relevant problem is the examination-scheduling problem.

The job of school course scheduling is a complex system that operates under multiple constraints. Including for teacher, classes, braches, days and hours. For example, a teacher may have a fixed number of hours in a week, at most two lessons can be scheduled at the same class in a day, and they may require at least one afternoon free. These are conditions on a real school system, by using an Integer Programming, for helping to administrative staffs that are engaged to the course scheduling activity.

The large number of events (courses) to be scheduled and a wide variety of constraints imposed on timetabling makes the set of all possible solutions (i.e., the search space of the problem) very large indeed. The construction of a timetable can be an extremely difficult task and its manual solution can require a lot of effort. Timetabling problems have attracted the attention of the scientific community from a number of disciplines (including OR and Artificial Intelligence) for about 40 years and over the last decade or so there has been an increased interest in the field.

A wide variety of approaches to the course-scheduling problem have been described in the literature. E.g. genetic algorithm, heuristic method, coloured mapping, linear programming and integer programming. Here, we investigate how to use IP to solve the problem, how to enter the data and how to present the output in the most simplest way.

A mixed-integer programming model for the problem is formulated in section three and solution algorithms and computational results are presented in the section. Test instances based on real class and school data for eight grades are solved directly by the GAMS/CPLEX-MIP Package. Finally, section five concludes the thesis with a summary and some implementation recommendations.

2. LITERATURE REVIEW

Educational timetabling is a major administrative activity for a wide variety of institutions. A timetabling problem can be defined to be the problem of assigning a number of events into a limited number of time periods. Wren (1996) defines timetabling as follows:

‘‘Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way as to satisfy as nearly as possible a set of desirable objectives.’’

A wide variety of approaches to timetabling problems have been described in the literature and tested on real data and cases at universities and high schools. They can be roughly divided into four types (Carter, 1986; Carter and Laporte, 1998): (1) sequential methods, (2) cluster methods, (3) constraint-based methods, and (4) meta-heuristic methods.

(1) Sequential methods: These methods order events using domain heuristics and then assign the events sequentially into valid time periods so that no events in the period are in conflict with each other. In sequential methods, timetabling problems are usually represented as graphs where events (courses) are represented as vertices, while conflicts between the events are represented by edges. For example, if some students have to attend two courses there is an edge between the nodes that represents this conflict. The construction of a conflict-free timetable can therefore be modeled as a graph-colouring problem. Each time period in the timetable corresponds to a color in the graph-colouring problem and the vertices of a graph are coloured in such a way so that no two adjacent vertices are coloured by the same colour. A variety of graph colouring heuristics for constructing conflict-free timetables is available in the literature (Carter and Laporte, 1998). These heuristics order the events based on an estimation of how difficult it is to schedule them. The heuristics that are often used are:

- Largest degree first. Events that have a large number of conflicts with other events (i.e., a large degree) are scheduled early. The rationale is that the events with a large number of conflicts are more difficult to schedule and so should be tackled first.
- Largest weighted degree. This is a modification of the largest degree first which weights each conflict by the number of students involved in the conflict.
- Saturation degree. In each step of the timetable construction an event, which has the smallest number of valid periods available for scheduling in the timetable constructed so far, is selected.
- Colour degree. This heuristic priorities those events that have the largest number of conflicts with the events that have already been scheduled.

Once the events are ordered for scheduling, a number of approaches can be employed to select a valid period for each event. For example, the earliest valid period can be selected, or the ‘‘best’’ valid period in the context of a defined objective function, etc.

(2) Cluster methods: In these methods the set of events is split into groups that satisfy hard constraints and then the groups are assigned to time periods to fulfill the soft constraints. Different optimization techniques have been employed to solve the problem of assigning the groups of events into time periods (Balakrishnan, 1992). The main drawback of these approaches is that the clusters of events are formed and fixed at the beginning of the algorithm and that may result in a poor quality timetable.

(3) Constraint-based approaches: In these methods a timetabling problem is modeled as a set of variables (i.e., courses) to which values (i.e., resources such as rooms and time periods) have to be assigned to satisfy a number of constraints (Brailsford, 1999). Usually a number of rules are defined for assigning resources to events. When no rule is applicable to the current partial solution a backtracking is performed until a solution is found that satisfies all constraints.

(4) Meta-heuristic methods: Over the last two decades a variety of meta-heuristic approaches such as simulated annealing, tabu search, genetic algorithms and hybrid approaches have been investigated for timetabling. Some very good results have been reported in (Burke and Ross 1996), (Burke and Erben, 2001) and the timetabling section. Meta-heuristic methods begin with one or more initial solutions and employ search strategies that try to avoid local optima. All of these search algorithms can produce high quality solutions but often have a considerable computational cost.

As early as the sixties simulation approaches, other heuristics and graph colouring methods appeared in the literature while they were attempting to solve some variations of the problem in an effective way. Specifically, Welsh and Powell (1967) use the graph colouring technique to solve a timetabling problem, while Schmidt and Strohlein (1979) report on simulation techniques. More than 30 years later, there is still the need to approach the problem in a systematic way that facilitates the inclusion of real-world variants of the problem along with automation and solution of very big problems.

Among the first approaches in mathematical programming, Lawrie (1969) and Akkoyunlu (1973) present linear and integer programming models for some versions of the problem and thus succeed in computing optimal solutions for a school and a university timetabling problem, respectively. More specifically, (Lawrie, 1969) the timetabling process starts with the generation of the so-called ‘‘layouts’’ for the different groups of students and an integer programming formulation attempts to find a set of optimal ‘‘arrangements’’, one for each group. Conversely, in (Akkoyunlu, 1973) the problem is solved as an assignment problem between a set of courses and the time periods of a weekly timetable, with central focus on preventing conflicts throughout this assignment. A solution is provided for the faculty assignment problem, a problem closely related to the timetabling problem, using linear programming models. The same problem was studied and a solution was attempted, again with the help of mathematical programming. More recently, in the teacher assignment problem is formulated as a MIP problem and is solved as a special case of the fixed charge transportation problem. In this case the proposed specialized algorithm, perform better than publicly available packages.

Using goal programming, the teacher assignment problem is combined with a form of the timetabling problem and solved through commercial software for goal programming. In a similar manner, a linear programming formulation is provided for the classroom allocation problem, a sub-problem of the university timetabling.

IP formulations for the school and the university timetabling problems as optimization problems are also given (Werra, 1985), where the NP-completeness of both problems is shown even for simple versions. The author, however, chooses graph theory approaches for the solution of the problems under consideration. Extensions to this work, especially with regards to the so-called conflict matrix, are presented by Tripathy (1984). Given the difficulties of those days to solve large IP problems, the Lagrangean relaxation is proposed as a possible solution approach for the resulting model. However, in other real-world applications (Tripathy, 1992) a heuristic approach is proposed instead. Along with the mathematical formulation, several authors divide requirements into two groups; the hard ones, which are included in the constraints and they define the search space, and the soft ones, which are included in some way in the objective function. A similar strategy for a timetabling problem for universities is solved in (Aubin and Ferland, 1989) by grouping sub-problems. A solution approach for the same problem, however with lectures of different length is provided in (Ferland 1985); in this formulation approach lectures may last one, two or three periods.

Apart from the classical mathematical programming approaches, several new and most efficient techniques for combinatorial problems have also being used for the timetabling problems. Among these, tabu search was used in (Costa 1994) for the solution of the school and university timetabling problems; constraint logic programming is presented. Last but not least, genetic algorithms have been utilized as an effective tool for the solution of timetabling problems. For example in (Drexl and Salewski, 1997) university timetabling problems are solved through these techniques.

An introduction to timetabling is written by Werra (1985) that includes a short review of timetabling models, the needs of automated and interactive timetabling methods. His purpose is to introduce the reader to some basics problems, which occurs in most of the real cases; class-teacher problem and its variations, then course-scheduling model together with some closely related problems.

Approaches for the solution of the timetabling problem have been from pure simulation of the hand construction of a schedule to graph theoretic approaches, three-dimensional transportation models, quadratic assignment model etc. The complex requirements of timetabling often result in a large size problem when formulated mathematically. This becomes a great hindrance to solving the timetabling problem employing mathematical programming techniques. Due to this mathematical programming techniques were rarely used for timetabling till 1970's. N.L. Lawrie (1969) describe the mathematical formulation given to the timetabling problem and the computational methods used in obtaining solution. Availability of faster and larger computers along with better algorithms has resulted in some approaches to solving timetabling problems using mathematical programming techniques.

A. Tripathy (1984) formulated a timetabling problem as a large integer linear programming problem in 0-1 variables. A solution method based on Lagrangean relation coupled with subgradient optimization is presented. The solution method also incorporates a branch and bound procedure that takes advantage of special sets of ordered variables. He had got difficulties such that large storage requirement and long computational time. He computed a large timetabling problem involving 900 subjects in a one-year graduate program. A. Tripathy (1992) did computerized decision aid for timetabling as a large complex problem. In the absence of a precise objective function, the solution process mostly aims at a feasible solution. This is the fact that the quality aspect of timetabling that is often not care of explicitly. This is due to the fact that the quality is largely the perception of the decision maker. This necessitates interaction of the decision maker during the generation of timetable through a computer. Case data of a timetabling problem have been analyzed. There is a quality parameter for a timetable, which is reflected by the preference for one timetable over the other. It is often difficult to express the quality parameter precisely. This remains largely the perception of the decision maker. Accordingly it is desirable that the decision maker is involved actively in the timetable generation process. This has been achieved by providing a decision aid in the form of schedule generation to the decision maker. The requirement of the timetabling problem has been mathematically and these have been taken care of at the various stages of timetabling. The special characteristics of the problem are the highly dense conflict matrix. The computer is used in aiding the decision maker to arrive at a desirable timetable.

J.J. Dinkel, J. Mote and M.A. Venkataramanan (1989) describe a network-based decision support system approach to the most general form of the academic course-scheduling problem. The dimension of faculty, subject, time and room are considered by incorporating a penalty function into a network optimization approach. The approach, based on a network algorithm, is capable of solving very large problems. This methodology can be applied to other scheduling situations where there are competing objectives and multiple resources. Such situations include: scheduling exams, times and rooms in an academic setting, and scheduling of clients, times, and facilities for physicians, hospitals, dentists, counselors, and clinics. Common problems in such settings include the utilization of available space, and dissatisfaction with assigned times and locations.

Aubin, and Ferland (1989) deal with large-scale school timetabling problems. As in the approach used by practitioners to handle the problem manually, two different components are analyzed sequentially; a timetabling is generated and student are assigned to course sections. The timetable is derived taking into account student registrations and classroom availabilities. The objective is to reduce the number of conflicts where lectures taking place simultaneously have common students or lecturers, or require the same classroom. The two components are strongly interrelated since the assignment of students to section influences the timetable derivation, and vice-versa, the timetable inflorescences the assignment of students to sections. The computerized approach iterates to modify successively the timetable and the assignment until the heuristic produces to deal with the subproblems can no longer improve the objective. The proper scope of the these results is to use the approach within a decision support system providing interactive functions to analyze closely the solution generated and to modify it into the suitable solution. This problem has two main components. In the timetabling subproblem a master timetable is derived taking into account student registrations and lecturer and classroom availabilities. The grouping subproblem specifies groups of students for large courses that have several times during the week. Aubin and Ferland purposed an approach to handle both subproblems and to deal with strong relationship. The solution approach several heuristic procedures to handle the subproblems that are formulated as assignment type problems in which entities are assigned to resources by using penalty terms for conflict and excessive use of classrooms.

D. Abramson (1991) studied on constructing school timetabling using simulated annealing, sequential and parallel algorithms method. Simulated annealing is a relatively new technique for solving optimization problems. The traditional solutions for the timetabling problem have either involved linear programming or heuristic search techniques. The result of serial algorithm is very promising and warrants further work. The speed of algorithm can be further improved by implementing a parallel program. It has been implemented on shared memory multiprocessor. This method requires further investigation.

Asratian and Werra (2002) consider a theoretical model which extends the basic "class-teacher model" of timetabling and which corresponds to some situations, which occur frequently in basic training programs of universities and schools. A generalizing class-teacher model, which is able to handle situations where there are several disjoint groups of classes which have to take some group-lectures. This model is suitable for our case study because of the sample school is mostly use class-teacher modeling for timetabling.

Costa (1994) has a paper on a tabu search algorithm for computing an operational timetable. Costa thinks that the many constraints of different types that must be taken into account and the volume of data make the timetabling problem very hard to solve. Various specific requirements are considered, and after having described precisely the problem to solved. He presented a general technique based on tabu search for finding an acceptable timetable. Costa says that the tabu search technique, developed independently by F. Glover and P. Hansen.

A number of papers in the literature have suggested various algorithmic approaches for solution of the school-timetabling problem.

The timetabling problem for a typical high school environment was modeled and solved using a constraint programming (CP) approach. In addition, operations research (OR) models and local search techniques were also used in order to assist the constraint programming search process by effectively reducing the solution search space. Relaxed models that can be solved using minimum cost matching algorithms were used in order to calculate problem lower bounds at various instances of the solution process. These bounds were in turn used to prioritize the search options of the constraint programming process. The use of minimum cost matching model in the search process is an economical and efficient mechanism for the creation of effective search strategies and it is a competitive manner of introducing problem domain information in the CP environment. By including in the solution process a sequence of local search steps, the solution quality was further improved. Several large problems were solved and actual computational results for specific problem instances are presented. Valouxis (2003) remind that there are three main timetabling problem instances-university, school and examinations-according to this classification fits in the school timetabling class of problems.

The job of school course scheduling for classes is a complex system which operates under many constraints that are usually divided into two categories: ‘‘hard’’ and ‘‘soft’’ (Burke and Ross 1996).

Hard constraints are rigidly enforced. Examples of such constraints are:

- Collisions may not be permitted. In a timetable a collision occurs when two or more courses are scheduled at the same period for the same teacher, for the same class. No resource (students or staff) can be demanded to be in more than one place at any one time.

- A timetable has to be complete. A timetable is complete when all courses planned for every class appear in the timetable, with the right amount of time periods for every course and also, when every person of the teaching staff is assigned the total number of teaching periods that the institution requires. For each time period there should be sufficient resources (e.g., rooms, invigilators, etc.) available for all the events that have been scheduled for that time period.

- A timetable should accommodate requests for sessions of consecutive teaching periods. Depending on the course and the number of teaching periods assigned per week, a teacher may choose to give the course in single-period or multi-period sessions. The timetable has to be able to schedule a given course in any scheme that the responsible teacher may choose.

Soft constraints are those that are desirable but not absolutely essential. In real world situations it is, of course, usually impossible to satisfy all soft constraints.

Examples of soft constraints (in both exam and course timetabling) are:

- Time assignment. A course may need to be scheduled in a particular time period.

- Time constraints between events. One course may need to be scheduled before/after the other.

- Coherence. Teachers may prefer to have their entire course in a number of days and to have a number of lecture-free days. These constraints conflict with the constraints on spreading events out in time.

- Resource assignment. Teachers may prefer to teach in a particular room.

L. Kwok and S. Kong, Y. Kam (1997) work on the timetabling problem to depict a complete picture of the timetabling exercise in Hong Kong and to provide some suggesting solution for improving exercises, Their research addresses; understanding the problem domain, setting up an evaluation frameworks. This survey was focused on understanding the timetabling problems in secondary schools in Hong Kong. The five intrinsic difficulties of the timetabling problem, namely filling 100% of scheduling slots, extremely high room utility rate, high teaching loading, class splitting and blocking of time slots should be carefully incorporated into the design of software based solution of the scheduling problem in Hong Kong secondary schools.

Yen-Zen Wang (2002) suggests a genetic algorithm method that is employed to solve for teacher assignment problem within a school setting. It is demonstrated that genetic algorithms could significantly reduce the time spent on teacher appointments, avoid many improper solutions, speed up the searching efficiency and flexibility to solve the problem.

The genetic algorithm is often used as a search algorithm, which is based on the biological principle of selection, reproduction and mutation (Mayer, 1999). It uses these principles to improve the fitness of a population of chromosomes and finds the solution space related with a problem. Using genetic algorithm to solve an optimization problem into the parameter's optimization problem (Wang 2002). The goal of optimization is to find a proper combination of a series of parameters to achieve the most satisfaction, either minimum or maximum, depending on the requirement of the problem.

There are some differences between genetic algorithms and the other optimization searching algorithms, shown as follows:

1. Genetic algorithms work with a combination of coded parameters. Not the parameters themselves.

2. Genetic algorithms are global searching algorithms. They search multiple combinations of different parameters. Thus, they reduce the probability of obtaining local optima. With the functioning of genetic algorithms, combination of parameters are searched simultaneously are called ‘population’ which is the basic functioning unit of genetic algorithms.

3. Genetic algorithms use objective function information only and are in need of no auxiliary mathematical knowledge, such as gradient, derivatives. Etc. There are wider ranges of applications than mathematical-based methods.

4. Genetic algorithms use rules for probabilistic transition to find the optimal solution.

Wang (2002) says that by using coding and searching in a population, without auxiliary information and randomized operators, the above features provide the efficiency and robustness for genetic algorithms, thus resulting in more advantages than other searching methods.

Özcan and Alkan developed Stable Situation Genetic Algorithm (SSGA) for university scheduling problem. SSGA tried as multi population multi constraints, they claim that multi population stable situation genetic algorithm is increased together to hybridize and mutation. E.Özcan (2003) worked on a standard data format for timetabling problem end to design, TimeTabling Markup Language (TTMN).

Edmund K. Burke and S. Petrovic (2002) give a brief introduction to some recent approaches to timetabling problems that have been developed in the Automated Scheduling Optimization and Planning Research Group (ASAP). The research consists of three parts. Firstly, recent heuristic and evolutionary timetabling algorithms are discussed. Secondly, an approach that considers timetabling problems as multicreateria decision problems is presented. Thirdly, they discuss a case-based reasoning approach that employs previous experience to solve timetabling problems.

Dimopoulou, Milliotis (2004) examine the design and implementation of a computer network based system to aid the construction of a university course timetable. The system uses on integer programming model that assigns courses to time slots and rooms to construct each department's timetable. An automated procedure is generated to link the integer-programming model with each department's data and to resolve conflicts produced by the distribution of process.

Salem M. Al-Yakoob and Hanif D. Sherali (2005) present a mathematical model for assigning faculty members to classes including, among typical academic-class scheduling issues. The time-slots for classes are initially assumed to be given and an integer-programming model (CFAM) is introduce to solve the resulting problem.

Models and algorithms based on integer programming are proposed by S.Daskalaki, T. Birbas and E.Hous in (2004). The model is treated as an optimization problem; the objective is a linear cost function. With this objective, it is possible to consider the satisfaction of expressed preferences regarding teaching periods or days of the week or even classrooms for specified courses. Moreover, with suitable definition of the cost coefficients in the objective function, it is possible to reduce the solution space and make the problem tractable. The model is solvable by existing software tools with IP solvers, even for large departments. The case of a large number of courses and teachers is presented along with its solution as resulted from the presented IP formulation.

In recent years, because of the advancements in computer software and hardware, IP and MIP formulations have started being again an acceptable approach for many combinatorial problems. The new technologies in information systems, the availability of reliable software and the ability to solve relatively large problems in relatively short time are the main reasons for making this traditional modeling approach attractive for the solution of realistic problems. Two decades back the problems that were solvable by classical IP techniques, mainly branch-and-bound, carried tens of integer variables. Now a problem with many thousands and in special occasions millions of binary variables is not necessarily a problem. In regards with the timetabling problems, IP models have been presented in (Dimopoulou and Miliotis, 2001 and 2004) for the university timetabling problem and in (Daskaladis and Birbas, 2004) for the school timetabling problem; the solutions produced with commercial software presented no real problem in terms of computation times.

School timetabling problems often admit multidimensional assignment formulations, whose solutions depend to a large extent upon intricacies of the scheduling environment and the size of the problem in terms of the number of variables and constraints. A wide spectrum of related multidimensional optimization problems have been investigated. Two-dimensional assignment problems, in particular, have been extensively studied in the literature, and fast and efficient solution algorithms have been developed for certain classes of these problems. Three-dimensional assignment problems, although notorious for their complexity, have also drawn considerable attention due to the fact that many real-world problems admit such formulations (Al-Yakoob S.M., Sherali H.D., 2005). Four- and higher-dimensional assignment problems normally require heuristic procedures and massive computing power in order to derive good quality feasible solutions.

Class scheduling and timetabling problems have been widely addressed in the literature, and many mathematical models have been developed to tackle such problems. However, due to the specific nature of each individual problem, there is no universal model that can be utilized to solve all such problems. Among many such specialized problem treatments that are relevant to this paper, we refer the reader to Astratian and de Werra (2002), Aubin and Ferland (1989), Ferland and Roy (1985), Costa (1994), McClure and Wells (1984), Wright (1996), de Werra (1985), Lawrie (1969), Hertz (1992), Welsh and Powell (1967), and Tripathy (1980,1984 and 1992), Kang and White (1992).

There exist various school timetabling problems depending on the environment and the characteristics of the particular school level (Valouxius 2003). In this study the first eight grade school situation in which the teachers teach in several different class sections during the day and the students remain in their classrooms is modeled and solved. The school timetabling problem is combinatorial and there are several strict organizational and sequence-related rules that must be respected. The problem specifications used in the study, although they closely describe the situation of a typical Turkish school, are quite general and abstract, which makes the findings of this work applicable to wider set of school timetabling problems. The specifications mainly focus on the fact that each teacher is scheduled to lecture for a given number of hours at a fixed set of class sections and the requirement that all the class sections must be always in session without any idle periods in their daily schedules.

The timetabling problem instance, we present in this study attempts to represent the main peculiarities of the Turkish schools although we strongly feel that the algorithmic methodology and the problem solution experiences presented here also hold for other instances of this problem. The solution methodology presented in this study is facilitated by the use of GAMS programming. In particular, the GAMS solver library was used as the primary base for the implementation and experiments. The large number of relatively tight constraints and the addition of operations research techniques for the calculation of lower bounds effectively assist the solving process. The definition of effective sequences of subproblems, based on our previous experience on similar scheduling problems (Daskalaki, Birbas and Housos 2004), has further improved the solution process.. The basic elements of this work are operations research and GAMS/Cplex 9.0 programming usability that is declare before (See Appendix A).

In this study, school-timetabling problem is modeled as binary problem using 0-1 variables. The model provides constraints for a large number of different rules and regulations that exist in school environments. The model succeeds in creating timetables that are free from collisions between courses, teachers and classrooms and they are complete from all aspects; moreover it supports the scheduling of courses that require consecutive time periods. Courses that require sessions that are repeated several times to accommodate different groups of students.

The IP formulation followed in this study. The binary variables are defined in such a way that the structural elements: teachers, days, time periods, classrooms and courses, are preserved in the model. This choice was made knowing that in reality the assignment is performed between the set of triplets (teachers, classrooms and courses) and the set of pairs (day, period). Of course this choice leads to a very large number of possible variables, however with the introduction of suitable subsets of the basic sets of teachers, days, time periods, classrooms and courses, this number reduces to manageable sizes. On the other side, the proposed definition of binary variables leads to very flexible elements as far as the modeling is concerned. Carrying all the important information along with the variables provided great flexibility in the modeling process, so that every rule at the school timetabling process presented was easily turned into mathematical equations.

Moreover, the introduction of the auxiliary variables for the consecutiveness constraints was also quite successful in managing the complexity of these issues. It is known that requests for consecutive time slots in timetabling problems increase complexity and make the problem hard. The emphasis of the paper is on modeling the problem using IP and GAMS programming, the models that resulted from the real-world problem in our case study were still solvable by commercial packages using the classical branch and bound approach. The solutions found through the presented model are optimal in the sense that the objective function was chosen to be the means of introducing preferences for certain time periods, days and classrooms for all courses involved and teachers that assigned to the class and courses.

3. STATEMENTS AND MODELLING THE SCHOOL TIMETABLING PROBLEM

3.1. SAMPLE

This section presents a mathematical model and solution methodology and solution of the problem. In the schools, a number of teachers with various educational specializations are available. Every teacher is qualified to teach a certain number of courses. In addition, the students are organized eight school years with every year having a number of classrooms. The lectures that must be given to every class are predefined and common for the whole country.

The teaching days are Monday to Friday and the daily teaching hours for the students are either six or seven or eight. Every class sections have its own permanent room that implies that there is no need for the existence of room constraints in this version of the high school timetabling problem. The teachers are assigned to the class sections before the creation of the timetable and this indeed is one of the main inputs of this model.

The school for which the system was created is a comprehensive school in Istanbul, which contains over 700 pupils of all grades 1 to 8 according to year-groups. The Private ALEV Schools, which belong to The Aware of St.George High School's Graduates.

There are over 70 teachers, whom work full-time. The subject taught includes the full range of National Curriculum. The school operates for five days of the 37 weeks a year. There are eight time periods each day (five in the morning and three in the afternoon) and hence 40 time-slots per week.

The first four years (years 7-11) are known as the ‘lower school’ and the year-groups is divided into three or four classes (according to the number of registered students). The rest the school (years 12-15) forms the ‘upper school’ is divided into three or four classes. More details will be sown in School Profile (Table 3.1)

Table 3.1: School Profile	number
Students	700
Teachers	75
Administrators	8
Guidance Consolers	5
Secretary	5
Other Staffs	24
Administrator offices	4
Classrooms	32
Music Classrooms	2
Paint Classrooms	2
Orff Classroom	1
Glass work classroom	1
Ceramic work classroom	1
Science Laboratories	3
Computer Laboratories	2
Sport Fields	1 closed,3 open
Multi-Media Rooms	2
Teachers room	1
Library	1
Graphic rooms	2
Conference Saloon	1
Cafeteria	1

There are nineteen different main courses in the school curriculum from the first year grade to the eight-year grade; the each one has different periods of time. Those are shown in Table 3.2.

Table 3.2: Courses , Levels and Time Periods Requirements

Courses \ Levels	1st	2nd	3rd	4th	5th	6th	7th	8th
Class Teaching (CTc)(LI,Trk,Mth,EE,Gd)	27	27	27	26				
Study German (SGr)	4	4	4	3	3	3	3	3
German (Grm)				2	2	4	4	4
Music (Msc)	2	2	2	1	1	1	1	1
Paint (Pnt)	2	2	2	1	1	1	1	1
Work Education (WEd)				3	3	2	2	2
Computer (Cmp)	1	1	1	1	1	1	1	1
Sport (Spr) (Physical Education)	2	2	2	1	1	2	2	2
Turkish (Trk)					6	5	5	5
Study (Std)					5	5	5	5
Guidance (Gdn)					1			
Science (Scn)					4	3	3	3
Traffic (Trf)					1	1		1
Mathematics (Mth)					4	4	4	4
Socail Information (SIn)					3	3	3	
Citizenship (Ctz)							1	1
History of Turkish Repuplic (HTR)								2
Culture of Religion and Morality Info. (CRM)					2	2	2	2
English (Eng)						1	1	1

3.2. RULES FOR THE TIMETABLING PROBLEM

Building a weekly timetable for an institution is a tedious process that administrators usually undertake after spending days and hours. In this effort timetablers follow several rules, some of which are so important that they may never be violated (hard constraints) and others that are not as important and usually are obeyed only if all hard constraints are satisfied and there is still have for better solutions under some objectives for quality (soft constraints).

There are some assumptions before listing the constraints of the problem as follows:

Assumption 1 (A1): All lecturers are preassigned to teachers in the given subjects.

Assumption 2 (A2): All classes do not change rooms.

Assumption 3 (A3): All teachers know their preference or penalty coefficient.

Assumption 4 (A4): Every set of students (classes) should have an activity (club affairs) periods as two sessions at the special days.

The list of the rules considered in this model in order to be able to create an acceptable and viable timetable is;

Hard constraints for the school timetabling are regulated by the following basic rules:

Hard Constraint 1 (H1): Collisions may not be permitted. In a timetable a collision occurs when two or more courses are scheduled at the same period for the same teacher, for the same class.

a) No teacher can meet two class sections at the same day and hour.

b) Two teachers should not be assigned to the same class section at the same day and hour.

c) Two teachers cannot be scheduled to same special room (laboratories).

Hard Constraint 2 (H2): A timetable has to be complete. A timetable is complete when all courses planned for every class appear in the timetable, with the right amount of time periods for every course and also when every person of the teaching staff is assigned the total number of teaching periods that the institution requires.

a) Every class section must have in the timetable the specified number of lectures by the proper teachers.

b) Every teacher must have in the timetable the specified lectures for the proper class sections and working days.

c) There must not exist non-teaching hours (idle hours) in the middle of a sequence of daily teaching hours.

Hard Constraint 3 (H3): A timetable must be suitable for club activities at a special day and at the special time periods of the day. And a course has at most two time periods in a day.

a) Every level of classes must have an activity (clup activities) periods, last two lesson on a certain day of the week.

b) There exist at most two periods for a type of course in one day.

Hard Constraint 4 (H4): A timetable should accommodate requests for sessions of consecutive teaching periods. Depending on the course and the number of teaching periods assigned per week, a teacher may choose to give the course in single-period or multi-period sessions. The timetable has to be able to schedule a given course in any scheme that the responsible teacher may choose to follow.

a) Some courses should be preferred block time periods (sport, art,..etc.).

b) Block periods must not be broken by the lunchtime.

c) The teaching schedule of every class section must be continuous and always starting on the first hour of the day.

Similarly, as soft constraints the list of rules considered in this model for the school timetabling are regulated by the following quality rules;

Soft Constraint 1 (S1): Each teacher may have a free afternoon as a half-day, in the weekly timetable.

Soft Constraint 2 (S2): Preferences for teaching in specific time periods are obeyed as much as possible. Each teacher may express an opinion regarding the preferred period for teaching his/her courses. For example, one may prefer to teach a given course during morning periods, while another during afternoon periods.

3.3 MODELLING THE SCHOOL TIMETABLING PROBLEM

Following similar approach (Valouxius C., Houses E., 2003), (Al-Yakoob S.M., Sherali H.D. 2005), (Wang Y. -Z., 2002), we use IP to build the model that may construct the school timetables. In this modeling philosophy the equations for the actual IP model may vary from school to school to reflect the special requirements imposed by each one of them. However, the main structure of the model remains the same.

3.3.1. General Features of the Model

The necessary data sets, parameters, variables and scalar needed for the model definition are following;

Five sets are considered as the basic structural elements for this approach. These are:

I: The teachers, lecturer or other teaching staff, from now on called teacher who is going to teach a course in the timetable. For our problem, it is assumed that the assignment of courses to teachers precedes the timetabling process, and the teaching load of each teacher is an input to the timetablers. The set of all teachers to be utilized in a given timetable is denoted with the letter i.

M: The day of the week, on which a course or part of a course may be scheduled. The set of all days possible for scheduling is denoted with the letter m.

N: The time period of a day, on which a period of a course may be scheduled. In our approach, there are n1 and n2 index sets for regulation of the block period courses.

A time period is any forty minutes period from 8:15 a.m. till 15:15 in case work school for the teaching. 15 minutes first break for a fast breakfast, and then three times five minutes break for changing teachers. And one course time lunch break, there is15 minutes break between sixth and seventh periods and the last five minutes break.

K: The group of students for which a course in the timetable is designed. It also represents the classroom name, is named by k.

L: The courses to be scheduled for a given set of groups of students. All different parts of a given course have to appear in a weekly timetable and in a desired scheme, different for each course. The lectures (courses) are named by l.

Three parameters are considered as the basic structural elements for the approach. These are;

a: Assignment of teacher, classroom and course. These are fixed and assigned by the head of departments and administrators of school. E.g.: $a_{i, k, l}$ is denoted teacher i, in the classroom k give the lecture l.

bs: The block session parameters. E.g.: $bs_{k,\text{'Mathematics'}}$=1 means that mathematics courses of classes k are done one block periods in every week. If 4 hours periods math course exits the periods will be done 2,1 and 1. bs(k,'Science')=2 means that 5 science courses of class k are done 2,2 and 1.

sr: The special room parameters. E.g.: $sr_{\text{'computer'}}$=2 means that the number of computer laboratories are two. If the number of special rooms (laboratories) is less than the number of teachers who need the room then same type of lessons cannot be scheduled at the same time.

Another parameter is defined by **p** for the preference for the teachers (i), who give the penalty coefficient for the day (m) and the period (n). It is denoted by $p_{i,m,n}$. It is preference coefficient of teachers or administrators about courses, between 1 and 8, 1 is the most wanted coefficient to is the most unwanted coefficient. For example, if a teacher wants to earlier periods then he/she will write small penalty coefficient as 1 or 2.

Three different sets of binary variables are adopted, throughout this study.

x: The basic set of binary variable and denoted by $x_{i,m,n,k,l}$ where $i \in I$, $m \in M$, $n \in N$, $k \in K$, $l \in L$. The variable $x_{i,m,n,k,l}$ takes the value of 1, when course l, taught by teacher i to the group of student in the classroom k is scheduled for nth period of the day m.

b: b is an auxiliary variables and will be denoted by $b_{l,m,n_1,k}$ where $m \in M$, $n_1 \in N$, $k \in K$, $l \in L$, while n_1 is a natural number for starting period of block courses. The variable $b_{l,m,n_1,k}$ will take value of 1, when course l, which require the block period that started n_1th periods, is scheduled for day m for the class k.

y: y is the half day binary variable, is denoted by $y_{i,m}$. Every teacher has a free afternoon. The variable $y_{i,m}$ will take value 1, the teacher i has the free afternoon at mth day of the week, the schedule shows empty mth days' 6^{th},7^{th} and 8^{th} periods. A scalar '**D**' is used for controlling the equation of half day as a value 10.

One integer variable is used in the model, which represent the total time assignment on the weekly timetable. It is denoted by **z**, it is also the objective equation of the model.

3.3.2 Constraints For The Model

In this part the constraints for the model along with corresponding rule are presented. The constraints are grouped into seven groups, each consisting of a certain number of equations.

3.3.3. Hard Constraints

Constraint 1: A teacher should have only one course at the same periods of time (basic rule H1.b):

$$\forall i \in I, \forall m \in M \text{ and } \forall n \in N, \sum_{k}\sum_{l} x_{i,m,n,k,l} \leq 1.$$

Constraint 2: One group of student (class) can have only one course at the same periods of time (basic rule H1.a):

$$\forall m \in M, \forall n \in N \text{ and } \forall k \in K, \sum_{i}\sum_{l} x_{i,m,n,k,l} \leq 1.$$

Constraint 3: Each course should be scheduled for as many teaching periods as the curriculum of each class requirement (basic rule H2.a):

$$\forall i \in I, \forall k \in K \text{ and } \forall l \in L, \sum_{m}\sum_{n} x_{i,m,n,k,l} = a_{i,k,l}.$$

$a_{i,k,l}$**:** Parameter that the assignment of teacher, classroom, course requirement.

Constraint 4: If the number of special rooms (laboratories) is less than the number of teachers who need the room then same type of lessons cannot be scheduled at the same time (basic rule H1.c):

$$\forall m \in M, \forall n \in N \text{ and } \forall l \in L, \sum_{i}\sum_{k} x_{i,m,n,k,l} \leq sr_l$$

sr_l: Parameter that the assignment of special rooms.

Constraint 5: A class, in a day should have at most two periods of time from a type of course (basic rule H3.b):

$$\forall m \in M, \forall k \in K, \forall l \in L, \sum_{i}\sum_{n} x_{i,m,n,k,l} \leq 2.$$

Constraint 6: This constraint ensure that the timetable may manage requests for multi-period sessions are quite common for school courses and may concern part of a course, that is the lecture, sport course or lab work (basic rule H4).

In the model, it is assumed that the teacher or administrator provides the desired split of course.

In order to facilitating this need, the auxiliary variable $b_{l,m,n_1,k}$ is introduced, for each course k for which there is request for at least one block session. The indices i,m,n_1,k are defined just for $x_{i,m,n,k,l}$ variables, while indices n_1 give the starting period of time, of block course (multi-period session).

For example, if course l (mathematics) require totally 4 teaching periods per week for lectures, the teacher or administrator may choose to split them in one session of 2 periods and one session of 2 periods by declaring for bs_l, mathematics 1 block (1+1+2=4) or in two sessions of 2 periods then bs_l, mathematics is 2 blocks (2+2=4).

The block courses would not be started 5th and 8th periods because of lunchtime and last period of the day (basic rule H4.c).

$$\forall i \in I,\ n \neq 5 \ and \ n \neq 8,\ x_{i,m,n_1,k,l} + x_{i,m,n_1+1,k,l} \leq 2 * b_{m,n_1,k,l},$$

Some courses should be preferred block time periods(basic rule H4.a).

$$\forall k \in K \ and \ \forall l \in L,\ \sum_{m} \sum_{n_1} b_{m,n_1,k,l} = bs_l \ ,$$

bs_l is the limits of block periods for course l.

3.3.4 Soft Constraints

Constraint 7: Every teacher should have a free afternoon (the half day) in one day of the weekly timetable. The auxiliary variable $y_{i,m}$ is created for the half day constraints of the teachers. If $y_{i,m}$ has the value of 1 for a teacher i , on a day m, the afternoon periods of time 6th, 7th and 8th will be free that $x_{i,m,n,k,l}$ is the value of 0 (S1).

And a scalar big multiplier D is used for the satisfying the constraint:

$$\forall i \in I, \sum_{m} y_{i,m} = 1$$

and

$$\forall m \in M, \sum_{n \geq 6} \sum_{k} \sum_{l} x_{i,m,n,k,l} \leq (1 - y_{i,m}) * D$$

3.3.5. Fixed Variables For Preallocated Assignment

Every set of students (classes) should have an activity (club affairs) periods as two sessions at the special days (A4). For example, class 1A, 1B,...,3D have free activity period last two hours of Monday. The activity sets such that E1={1A, 1B,...,3D} , E_2={4A.4B,...,5D} and E_3={6A,6B,...,8B} . The activity periods are fixed to 0 ;

If $k \in E_1$ *and* $n \geq 7$ *and* $m =' Monday'$ *then* $x_{i,m,n,k,l} = 0$,

If $k \in E_2$ *and* $n \geq 7$ *and* $m =' Tuesday'$ *then* $x_{i,m,n,k,l} = 0$,

If $k \in E_3$ *and* $n \geq 7$ *and* $m =' Thursday'$ *then* $x_{i,m,n,k,l} = 0$.

Fixed variables for preallocated assignment can be used for any teacher preferences for a free period. For example First periods of a teacher who has a transportation problem, the administrator can fix the first period as idle:

i.e. $x_{'Tch44',m,n,'1',k,l} = 0$

3.3.6. Objective Of The Model

The objective function of the problem constraints two multiplicative components, corresponding the assignment of $x_{i,m,1,k,l}$ and a penalty coefficient $p_{i,m,n}$;

$$z = \textit{Minimize}\ \left(\sum_{i}\sum_{m}\sum_{n}\sum_{k}\sum_{l} \mathrm{p}_{i,m,n} * x_{i,m,n,k,l} \right).$$

The objective function is to minimize the function of two multipliers p and x. In any given timetabling problem the coefficient in the objective function may take any value. If all coefficients take the same value, then the problem becomes degenerate and all feasible solutions will be optimal. Of course, in all practical situations this is not possible and a more specific analysis for the correct assignment of the coefficients is needed. In the model the coefficients are assigned values in such a way as to reflect preferences for specific time periods of the day for all courses and for specific days of the week for the courses with multiple time periods sessions.

Assignment of values to $p_{i,m,n}$ coefficient according to requests for teachers, administrators or the department chairman for specific time periods of the day (S2). It is well accepted among teachers and the people that handle the timetabling process that certain days should have preferential treatment in terms of the time period that they are assigned. For each department there is a set of courses that are considered more difficult than others and require “prime time”, while other courses are less demanding and can be taught at any time of the day. The default value of p is 0 for all teachers. For example: if we write p(‘Tch55’,m,’1’)=1 then it means that the teacher55 does not want the first periods of week. If we write p(‘Tch55’,m,’1’)=-1 then the Teacher55 wants the first period of week. In general, negative values (-1,-2,...) indicate preference and positive values (1,2,3,..) indicate abstinence.

4. RESULTS

The problem is modeled and solved for the school mentioned in Chapter 3.

For a school year, there are seventy teachers, nineteen different courses (Table 2) and twenty-seven classes at eight level of teaching. In a day there are eight courses time periods and between 5th and 6th periods there is lunchtime. These are all shown on the school timetable.

All data were entered into the GAMS model. The solution found and it was proven to be optimal.

4.1. DATA REQUIREMENT FOR THE MODEL

In order to implement the model the following data were entered;

Teachers (i) : Seventy teachers were entered the model one by one.
Days (m) : Five days of the week from Monday to Friday were entered.
Time Periods (n) : Eight periods of time for a day were entered.
Classes (k) : Twenty seven classrooms that are refer to the group of students.
Courses (l) : Nineteen types of courses were entered.
The assignments of teacher, class and course (a): Teachers, his/her courses and which class will take the course were assigned. Number of periods of time in a week was assigned.
The assignment of block session (bs) : Which courses will have block sessions and the numbers of block.
The assignment of number special rooms (sr): The number of special rooms is assigned as laboratories.

4.2. THE SAMPLE SCHOOL

There are over seventy teachers, nearly of who work full-time. And there are more than ten teachers for activities as riding, football, theatre, etc. The subject taught includes the full range of National Curriculum. The school has got extra curriculum under the control of National Education Ministry about German language education.

The school has twenty-seven classes that represent the group of the students. For example 4A is the name of classroom and 4 is the fourth years of students in the education and A is the group name of the forth years students. The teachers change the classroom according to the course timetable and the students do not change the classrooms except Lab sessions, Art or Sport courses.

In the school there are one Chemistry, one Physics and one Life Intelligence Laboratories, there are two Computer Laboratories, one Sports Hall, one open Sport Field, one Multi Purpose Conference Room (for conference, theatre, gymnastics...), two Multi-Media classrooms, two Paint classrooms, one Orff classroom, two Music rooms, one Ceramics work room and a Glass work classroom (Table 3.1).

These classrooms occupation or timetabling are done by the responsible teachers of the departments or by administrators according to the needs of them or the needs of any staff who needs the classroom.

At the school, the data are collected from departments as art, science, mathematics, etc. and also from the administrators who are on the registration and accounting offices (Figure 4.2.1).

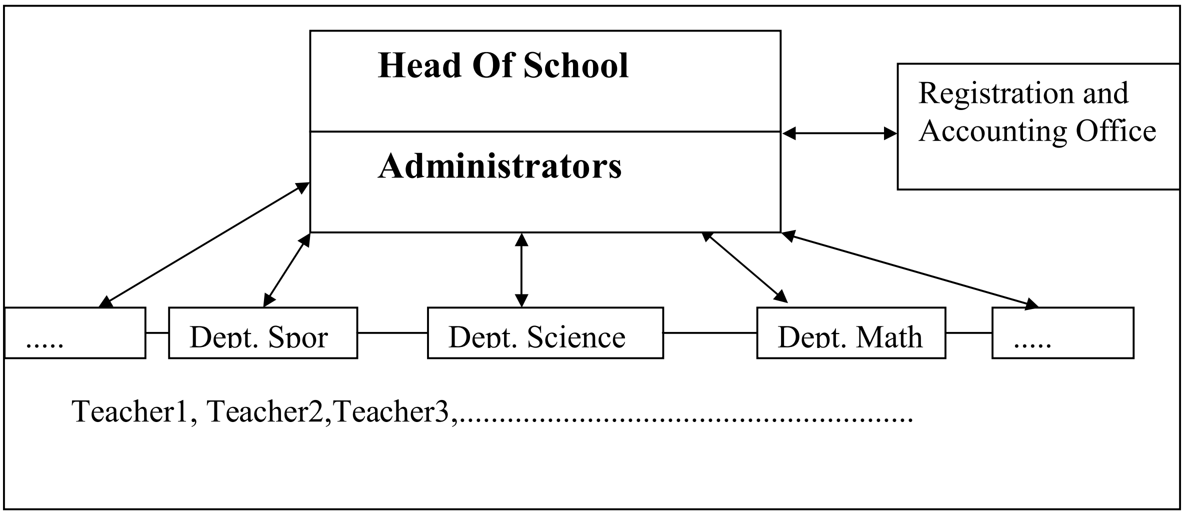

Figure 4.2.1: School Flow Chart

A staff that is an administrator starts to construct a timetable of the school one week or ten days ago before the beginning of the education year. The construction takes many hours in a day the constructor goes through stressful times during this activity. Recent days many schools are getting professional help from the commercials, but this causes some problems because of the commercials cannot leave their staffs in the school every day. If any problem or a need for change on the timetable, there is lack of time to overcome it.

An IP model based on the formulation described in this paper was developed for the construction of the 2005-2006 education period timetable. The penalty coefficient was assigned values according to the assignment of teachers and administrators discussed in the construction of objective function part. This causes the difficulties of calculation for the GAMS that are used for the programming as mixed integer programming minimization. The solver machine was used in an Intel Pentium-M740 processor and the solution found is proposed timetable for the school. The process time of the machine is approximately ten minutes. If a staff tries to construct a timetable for a school, has got twenty seven classrooms, it takes approximately five days.

4.3. PROBLEM INSTANCES AND RUNNING TIMES

For the model run on the computer, we try to solve the problem in different sizes, which give very good result for time saving. These sample running are shown at the Table 4.3 and also shown model statistics values.

Table 4.3 : Problem size and statistics

num. Of teachers	num. of classes	num. of courses	solution time (seconds)	optimality	iteration count
5	7	6	0.41	Yes	1475
56	13	18	34.0	Yes	13.695
72	27	19	377.2	Yes	270.581

Model Statistics (for the model with 72 teachers)

Blocks Of Equations	10	Single Equations	8930
Blocks Of Variables	4	Single Variables	16109
Non Zero Elements	69868	Discrete Variables	16108

4.4. GENERATED TIMETABLES

Loop statements and external files are used to present the solution in an understandable format that anybody can follow the resultant timetables. The timetables are displayed in three external files. The first timetable is "The General Timetable Of The School" (See Appendix C-I), the second timetable is "Timetables For Classes" (See Appendix C-II) and the third one is "Timetables For Teachers"(See Appendix C-III)

For the presentation purposes we selectively present the timetable for the first and sixth year level classes. The reason of this choose is the school program the first to fifth year levels have the similar courses and also the sixth to eight year levels have similar courses. The courses are shown on the Table 3.2.

After construction of general timetable has been done by the staff (timetabler), a declaration is done to each teacher for getting feedback. Table 4.4.1 shows a sample for general timetable.

Table 4.4.1: General Timetable Base Over Teachers. Teachers, days and periods are corresponding to the course and class.

2005-2006 Private .... School Timetable

			Monday								Tuesday							
Order	Teacher Name	Branch	P1	P2	P3	P4	P5	P6	P7	P8	P1	P2	P3	P4	P5	P6	P7	P8
1	Tch06	CTc	1A	1A	1A	1A					1A	1A	1A	1A	1A			
2	Tch07	CTc	1B	1B	1B	1B	1B				1B	1B	1B	1B	1B			
3	Tch08	CTc	1C	1C	1C	1C					1C	1C	1C	1C	1C			
4	Tch09	CTc	2A	2A	2A	2A	2A	2A				2A	2A					
5	Tch10	CTc	2B	2B	2B	2B					2B	2B	2B	2B	2B			
6	Tch11	CTc	2C	2C	2C	2C					2C		2C	2C	2C	2C		
	....	......																
21	Tch36	Grm	7C	6B				7C			6B					6B Std		7A Std
22	Tch46	Pnt	5C Wed	6C Wed	5A WEd	6D	4B	8B	5B			6C WEd	5c WEd	5B WEd				
23	Tch66	Mth	5B		6C		6C	6A			5B		6C	7C	7C			

This declaration consists of weekly timetable for each teacher as shown on the Table 4.4.2. It represents the teachers' name and classes details as duty days. The duty days are very important day of teachers who have got a place or a point to check the students in the breaks for unwanted events or accidents etc.

Table 4.4.2: Teachers Timetable
The Private .... School 2005-2006
Teacher's Name: Tch44.........

	Mon	Tues	Wed	Thu	Fri
1	8B WEd			7B WEd	
2	8A WEd	6A Pnt		8B WEd	4C Pnt
3	2A Pnt			6D WEd	8A WEd
4	4C WEd		1A Pnt	4c WEd	
5	7C WEd		3C Pnt	2A Pnt	6A WEd
6		4C WEd	3C Pnt	1A Pnt	6A WEd
7	6DWEd				7B WEd
8					7C WEd

The day of Duty: Thursday (Garden A)

After recovering of general timetable, a timetable for classrooms (the group of students) is written. This student's weekly course planning example is shown on the Table 4.4.3. Shows course time periods, and the name of the teacher who are assigned to the courses.

Table 4.4.3: Student Timetable
The Private .... School 2005-2006
Class Name: 6A

Periods	Mon	Tues	Wed	Thu	Fri
1	Trk	Grm	Mth	Scn	Trk
2	Grm	Msc	Scn	Grm	Eng
3	Cmp	Scn	Grm	Mth	SIn
4	Spr	Trk	Scn	Scn	Mth
5	Spr	Eng	SIn	Scn	Mth
6	Mth	Trk	Grm	Trk	Scn
7	Grm	Cmp	Grm	Act	Mth
8	SIn	CRM	Ing	Act	Scn

Course	Name	Teacher
Trk	Turkish	Tch56
Mth	Math	Tch66
Scn	Science	Tch59
CRM	Cul.Rel.Mor.	Tch72

4.5. CONCLUSION

The first year students or classes have six types of courses, which are shown in Table 4.5.1. Class Teaching; courses which have mostly used for reading and writing studies in Turkish and simple counting, simple addition in Mathematics. Life Information studies are also occupied. Music; which for learning and singing songs. Paint; which for drawing and painting studies. Sports (Physical Education); for development of physical abilities of students. Computer; the course of students for using computer. German study; for foreign language starting level German language introductions are given. In additionally two periods club activity exists, a student can choose any activity as ceramics, glass, chess, etc.

Table 4.5.1: General Timetable Base Over Classes . Teachers , days and periods are corresponding to the courses. For the first year classes

2005-2006 Private .... School Timetable

		Monday								Tuesday								Wednesday							
Order	Class	P1	P2	P3	P4	P5	P6	P7	P8	P1	P2	P3	P4	P5	P6	P7	P8	P1	P2	P3	P4	P5	P6	P7	P8
1	1A	LI	Mth	Trk	Trk	SGr	SGr	EE	Std	LI	Trk	Trk	Mth	Trk	Std	Act	Act	Msc	LI	Trk	Pnt	Trk	Trk	Std	Std
2	1B	LI	Trk	Mth	Trk	Trk	Msc	Std	Std	LI	Trk	Mth	Trk	Trk	SGr	Act	Act	Spr	Spr	LI	SGr	Trk	Trk	EE	Std
3	1C	LI	Mth	Trk	Trk	Spr	Spr	SGr	SGr	LI	Trk	Trk	Mth	Trk	Std	Act	Act	LI	Mth	Trk	Trk	Msc	Std	Std	Cnp

	Thursday								Friday							
Class	P1	P2	P3	P4	P5	P6	P7	P8	P1	P2	P3	P4	P5	P6	P7	P8
1A	LI	Trk	Trk	Mth	Msc	Msc	Cmp	SGr	LI	Spr	Spr	Trk	Mth	Trk	SGr	Gd
1B	LI	Trk	Mth	Trk	Trk	Msc	Std	Std	LI	Trk	Mth	Trk	Trk	SGr	Act	Gd
1C	LI	Mth	Trk	Trk	SGr	SGr	Pnt	Std	LI	Trk	Trk	Msc	Mth	Cmp	Std	Gd

Course	Name
LI	Life Information
Mth	Mathematics
Trk	Turkish
Msc	Music
Pnt	Paint
Std	Study
EE	Elective Extra
Act	Activity (Club Affairs)
Cmp	Computer
Gd	Guidance

The course names and abbreviations, levels and time requirements are shown in School Flow Chart (Figure 4.2.1). The first year classes' timetable is shown in Table 4.5.1 base over Classes that classes, days and periods are corresponding to the courses and teachers.

Judging the resulting timetable for the first year, one should note the following:

- All courses required by the curriculum appear timetable with no conflicts among them.

- Each course appears in so many periods as required from the curriculum, assigning courses and teachers in a regular classrooms (indicate 1A,1B,...). Also the session periods follow the requirements for each course. The sports lessons are scheduled block manner.

- All courses are scheduled from 1st period to 8th period following the preferences for certain time periods given as input from the timetabler.

- The block courses are not interrupted by lunch time period.

- Due to the penalty coefficients in the objective function it is possible to satisfy the preference given day and periods the course of teacher could be scheduled.

Similarly, in Table 4.5.2, one may observe the curriculum and the timetable for the students of the sixth year. Many comments that were put forth in the previous discussion could be repeated for this timetable too. The main difference here is the distinction between students of different courses and teachers for all courses. In fact, each courses has different teacher, provides more complexity of timetable for its own students, with the exception of a number of courses shared between divisions. As a result, overlapping (in time) between courses of different divisions is not allowed.

In those cases there should be no overlapping with any of the courses of either classes. The sixth year students have thirteen different courses as; Turkish, German, English, Mathematics, Science, Social Information, Music, Paint, Sport, Work Education, Traffic, Computer, Culture of Religion and Morality Information.

Table 4.5.2: General Timetable base over Classes . Teachers , days and periods are corresponding to the Courses. For the sixth year classes

2005-2006 Private .... School Timetable

	Monday								Tuesday								Wednesday							
Class	P1	P2	P3	P4	P5	P6	P7	P8	P1	P2	P3	P4	P5	P6	P7	P8	P1	P2	P3	P4	P5	P6	P7	P8
6A	Trk	Spr	Spr	Cmp	Grm	Mth	Std	Std	Grm	Pnt	Scn	Trk	Msc	Trk	Std	CRM	Mth	SIn	Grm	Scn	SIn	Std	SGr	SGr
6B	Mth	Grm	SIn	Spr	Spr	Cmp	Eng	Std	Grm	Scn	Mth	Trk	Std	SGr	SIn	Std	Msc	Mth	Grm	Trk	SGr	Trk	CRM	Std
6C	Scn	WEd	Mth	SIn	Mth	SGr	Std	Std	Grm	WEd	Mth	Trk	CRM	Cmp	SIn	Trk	SIn	Scn	Grm	CRM	Trk	SGr	SGr	Std
6D	Trk	Grm	Mth	Pnt	Std	Trk	WEd	Std	Grm	SIn	Msc	Mth	SIn	Std	Scn	Std	Mth	Trf	Grm	Eng	CRM	Std	SGr	Std

	Thursday								Friday							
Class	P1	P2	P3	P4	P5	P6	P7	P8	P1	P2	P3	P4	P5	P6	P7	P8
6A	Trf	CRM	Mth	Grm	Scn	Trk	Act	Act	Trk	Eng	SIn	Mth	WEd	WEd	Std	SGr
6B	Mth	Grm	Scn	WEd	WEd	Trk	Act	Act	SIn	SGr	Trf	CRM	Scn	Pnt	Trk	Std
6C	Eng	Grm	Trk	Grm	Std	Trk	Act	Act	Mth	Scn	Trf	Msc	Pnt	Std	Spr	Spr
6D	Cmp	Grm	SIn	WEd	Trk	SGr	Act	Act	Scn	Scn	Mth	Trk	CRM	Trk	Spr	Spr

Course	Name
SIn	Social Information
Mth	Mathematics
Trk	Turkish
Msc	Music
Pnt	Paint
Std	Study
Grm	German
Act	Activity (Club Affairs)
Cmp	Computer
Eng	English
SGr	Study German
WEd	Work Education
Spr	Sport
Scn	Science
Trf	Traffic
CRM	Culture of Religions and Morality Information

5. SUMMARY AND CONCLUSIONS

In this study we presented an IP formulation of a timetabling problem seen in many schools by using GAMS programming. The problem is a hard one and very complex, however, the choices made through the modeling process result to solvable and flexible models. The flexibility offered is due to the multi-dimensional variables, which allow details of the educational system to be modeled as constraints of the IP model.

The flexibility offered is due to the multi-dimensional variables, which allow low details of the educational system to be modeled as constraints of the IP model. A variety of rules may be represented in the model with suitable constraints provided by this formulation.

Moreover, the choice made for the cost function allows the introduction of certain preferences regarding time periods, days and classrooms, so that timetables can be improved according to well-accepted quality measures.

The schools can have different requirements according to their vision. In fact many schools have similar programming requirements. But in private school sector preferences and visions and missions are different so that the courses and periods of time are different. These types of schools cause having a new organization and timetabling study.

The timetabling problem for a 1-8th grade private school was used as a case study and was solved successfully. The timetable construction required scheduling of lectures, each have type carrying different number of time periods. Courses with consecutive block time periods require satisfaction and have turned into hard constraints.

Creating timetables for schools is tedious process, however automation is possible. The use of computers was not widely accepted by teachers who did not like the scheduling by the machine. Today's generation is highly accustomed to the machines and computers; the course schedulers and the teacher can take crucial decisions during the construction process.

Results indicate that moderate sized problems can be solved in about 6 minutes that indicated that computerized solutions are feasible for most schools.

Besides the saving in time, the computer has the advantage that the timetables produced do not violate any constraints. Furthermore the computer facilities can be used to print the timetables of classes, teachers and classrooms, etc. Of course the use of computer is interesting only for large schools or universities where one should try to construct a timetable with a computer.

It is essential that the codes be made easy to use, that the method be almost transparent to the user; feeling and understanding how to procedure works may help the scheduler to reach a good solution in a reasonable number of runs. To increase the chances of survival of such a program, the user should have a direct access to it; he/she should definitely not need to go a computing center outside of the school.

The purpose of this study is to develop a program to prepare a guide for the schools to avail these. The field of timetabling is nice topic for the specialist in Operations Research for several reasons: there are many problems to be solved that include pure combinatorial mathematics, computer science and applications.

REFERENCES

Abramson D., 1991. Constructing school timetables using simulated annealing: Sequential and parallel algorithms. Management Science 37(**1991**) 98-113.

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

Al-Yakoob S.M., Sherali H.D. 2005. Mathematical programming models and algorithms for a class-faculty assignment problem. European Journal of Operation Research xxx (**2005**) xxx-xxx.

Aubin J.,Ferland J.A., 1989. A large scale of timetabling problem. Computers and Operations Research 16(1)(**1989**)67-77.

Astratian A.S., Werra de D., 2002. A generalized class-teacher model for some timetabling problems. European Journal of Operational Research.143(**2002**)531-542.

Costa D. 1994. A Tabu Search algorithm for computing an operational timetable. European Journal of Operational Research 76(**1994**)98-110

Valouxius C., Houses E., 2003. Constraint programming approach for school timetabling. Computers & Operation Research 30(**2003**)1555-1572.

Balakrishnan N., Lucena A., Wong R.T.,1992. Scheduling examinations to reduce second-order conflicts. Computers and Operation Research 19(**1992**)353-361.

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

Brooke A., Kendrick D., Meeraus A., Raman R., Rosenthal R.E., **1998**. GAMS A User's Guide. GAMS Development Corporation Washington, DC 20007, USA.

Burke E.K., Ross P,1996. The Practice and Theory of Automated Timetabling: Selected Papers from the 1st International Conference on the Practice and Theory of Automated Timetabling, Napeir University, August/September **1995**. Sprigers Lecture Notes in Computer Science Series, vol.1153.

Burke, E., Erben, W. (Eds.), 2001. The Practice and Theory of Automated Timetabling

III: Selected Papers from the 3rd International Conference on the Practice and Theory of Automated Timetabling, University of Applied Sciences, Constant, August 16–18, 2000. Springer Lecture Notes in Computer Science Series vol. 2079(**2001**).

Carter M.W. and Laporte G., 1998. Recent development in practical course timetabling. In Burke and Carter (**1996**) pp 3-21

Carter M.W.,1986.A survey of practical of examination timetabling algorithms. Operation Research 34,(**1986**)193-202.

Dimopoulou M., Miliotis P., 2001. Implementation of a university course and examination timetabling system. European Journal of Operation Research. 130 (**2001**) 203-213.

Dimopoulou M., Miliotis P., 2004. An Automated University Course Timetabling System developed in a distributed environment's case Study. European Journal of Operation Research 153(**2004**) 136-147.

Daskalaki S., Birbas T, Housos E. 2004. An integer programming formulation for a case study in university timetabling. European Journal of Operation Research 153 (**2004**) 117-135.

Drexl A., Salewski F., 1997. Distribution requirement and compactness constraints in school timetabling. 102 (**1997**) 193-214.

E.Kieron Burke E.k., S.Petrovic S., 2002. Recent research directions in automated timetabling. European Journal of Operation Research 140(**2002**)-266-280.

Dinkel J.J., Mote J., Venkatarmanan M.A., 1989. An efficient decision support system for academic course scheduling. Operations Research 37(6)(**1989**)853-864.

Ferland J., Roy S., 1985. Timetabling problem for university as assignment of activities to resources. Computer and Operations Research. 12(**1985**) 207-218.

Hertz A., 1992. Finding a feasible course schedule using Tabu search. Discrete applied Mathematics 35 (**1992**) 255-270.

Kang L., White G.M., 1992. A logic approach to the resolution of constraints in timetabling. European Journal of Operation Research 61 (**1992**) 306-317.

Kwok L.-F., Kong S.-C and Kam Y., 1997 Timetabling In Honk-Kong, Secondary School. (Received 1 January 1997; accepted 1 March 1997 Elsevier Science 1 Ltd).

Lawrie N. L., 1969. An integer linear programming model of a school timetabling problem. Computer Journal. 12(**1969**) 307-316.

Mayer M.K., 1999. A network parallel genetic algorithm for the one machine sequencing problem. Computers and Mathematics with Applications,37(**1999**)1061-1071.

McClure R.H., Wells C.E., 1984. Applications and Implementation A Mathematical Programming Model For Faculty assignments. Decision Sciences. 15(**1984**)409-420.

Ozcan E., Alkan A. 2003. Zaman Çizelgeleme Problemleri İçin Standart Bir Veri Formatı Çalışması:TTML. International XII, Turkish Symposium on Artificial Intelligence and Neural Networks-TAINN **2003**.

Schmidt G, Strohlein T, 1979 Timetable construction—An annotated bibliography, The Computer Journal 23 (4) (**1979**) 307–316.

Tripathy A., 1980. A lagrangean Relaxation Approach to Course Timetabling. Journal of the Operation Research Society. 31(**1980**)499-603.

Tripathy A., 1984. School timetabling-A case in large binary integer programming. Management Science 30(12)(**1984**)1473-1489.

Tripathy A., 1992. Computerized decision aid for timetabling_ A case analysis. Discrete Mathematics. 35(3)(**1992**)313-323.

Wang Y.-Z.,2002. An application of algorithm methods for teacher assignment problems. Expert Systems with Applications 22 (**2002**) 295-302.

Welsh D.J.A., Powell M.B., 1967. An upper bound for the chromatic number of a graph and its application ti timetabling problems. Computer Journal. 10(**1967**) 85-86.

Wren A., 1996. Scheduling, timetabling and restring- A special relationship. In Burke Ross (**1996**) pp.46-75.

Werra de D., 1985. An introduction to timetabling. European Journal of Operational Research. 19(2)(**1985**)151-162.

Wright M. , 1996. School Timetabling Using Heuristic Search. Journal of the Operational Research Society. 47(**1996**)347-357.

APENDIX A CPLEX 9.0

1. INTRODUCTION

GAMS/Cplex is a GAMS solver that allows users to combine the high level modeling capabilities of GAMS with the power of Cplex optimizers. Cplex optimizers are designed to solve large, difficult problems quickly and with minimal user intervention. Access is provided (subject to proper licensing) to Cplex solution algorithms for linear, quadratically constrained and mixed integer programming problems. While numerous solving options are available, GAMS/Cplex automatically calculates and sets most options at the best values for specific problems.

2. HOW TO RUN A MODEL WITH CPLEX

The following statement can be used inside your GAMS program to specify using

Cplex Option LP = Cplex; {or QCP, MIP, MIQCP, RMIP or RMIQCP}

The above statement should appear before the Solve statement. The MIP and QCP capabilities are separately licensed, so you may not be able to use Cplex for those problem types on your system. If Cplex was specified as the default solver during GAMS installation, the above statement is not necessary.

3. OVERVIEW OF CPLEX

3.1. LINEAR PROGRAMMING

Cplex solves LP problems using several alternative algorithms. The majority of LP problems solve best using Cplex's state of the art dual simplex algorithm. Certain types of problems benefit from using the primal simplex algorithm, the network optimizer, the barrier algorithm, or the sifting algorithm. If the parallel option is licensed, a concurrent option will allow solving with different algorithms in parallel. The solution is returned by the first to finish.

Solving linear programming problems is memory intensive. Even though Cplex manages memory very efficiently, insufficient physical memory is one of the most common problems when running large LPs. When memory is limited, Cplex will automatically make adjustments that may negatively impact performance. If you are working with large models, study the section entitled Physical Memory Limitations carefully.

Cplex is designed to solve the majority of LP problems using default option settings. These settings usually provide the best overall problem optimization speed and reliability. However, there are occasionally reasons for changing option settings to improve performance, avoid numerical difficulties, control optimization run duration, or control output options.

Some problems solve faster with the primal simplex algorithm rather than the default dual simplex algorithm. Very few problems exhibit poor numerical performance in both the primal and the dual. Therefore, consider trying primal simplex if numerical problems occur while using dual simplex.

Cplex has a very efficient algorithm for network models. Network constraints have the following property:

• each non-zero coefficient is either a +1 or a -1
• each column appearing in these constraints has exactly 2 nonzero entries, one with a +1 coefficient and one with a -1 coefficient

Cplex can also automatically extract networks that do not adhere to the above conventions as long as they can be transformed to have those properties.

The barrier algorithm is an alternative to the simplex method for solving linear programs. It employs a primal dual logarithmic barrier algorithm that generates a sequence of strictly positive primal and dual solutions. Specifying the barrier algorithm may be advantageous for large, sparse problems.

Cplex provides a sifting algorithm that can be effective on problems with many more variables than equations. Sifting solves a sequence of LP subproblems where the results from one subproblem are used to select columns from the original model for inclusion in the next subproblem.

GAMS/Cplex also provides access to the Cplex Infeasibility Finder. The Infeasibility finder takes an infeasible linear program and produces an irreducibly inconsistent set of constraints (IIS). An IIS is a set of constraints and variable bounds that is infeasible but becomes feasible if any one member of the set is dropped. GAMS/Cplex reports the IIS in terms of GAMS equation and variable names and includes the IIS report as part of the normal solution listing.

3.2 QUADRATICALLY CONSTRAINED PROGRAMMING

Cplex can solve models with quadratic constraints. These are formulated in GAMS as models of type QCP. QCP models are solved with the Cplex Barrier method.

QP models are a special case that can be reformulated to have a quadratic objective function and only linear constraints. Those are automatically reformulated from GAMS QCP models and can be solved with any of the Cplex QP methods (Barrier, Primal Simplex or Dual Simplex).

For QCP models, Cplex returns a primal only solution to GAMS. Dual values are returned for QP models.

3.3 MIXED-INTEGER PROGRAMMING

The methods used to solve pure integer and mixed integer programming problems require dramatically more mathematical computation than those for similarly sized pure linear programs. Many relatively small integer-programming models take enormous amounts of time to solve.

For problems with integer variables, Cplex uses a branch and cut algorithm which solves a series of LP, subproblems. Because a single mixed integer problem generates many subproblems, even small mixed integer problems can be very compute intensive and require significant amounts of physical memory.

GAMS and GAMS/Cplex support Special Order Sets of type 1 and type 2 as well as semi-continuous and semi integer variables.

Cplex can also solve problems of GAMS model type MIQCP. As in the continuous case, if the base model is a QP the Simplex methods can be used and duals will be available at the solution. If the base model is a QCP, only the Barrier method can be used for the nodes and only primal values will be available at the solution.

APPENDIX B THE GAMS MODEL

Big Program:

$ontext

{ This program construct a course timetable for the 1 to 8 grades schools, by using GAMS programming facilities. It generates a Weekly General Timetable, Weekly Timetable for each Teacher and Weekly Timetable for each Student. We advise the user for getting a GAMS A USER'S GUIDE by GAMS Development Corporation 1217 Potomac Street, N.W. Washington, DC 20007, USA. http://www.gams.com.}

$offtext

$ontext

***********WEEKLY TIMETABLING PROGRAM***********

$offtext

sets

i **Ogretmenler {Teachers}**

$ontext

{To this part of the program for entering the element of teachers set. You have to write the name of the teacher between the slashes /..../.Put commas between each teachers name and use English characters.

Examples :/ teachername1, teachername2, teachername3,............,teachernamen/ .}

$offtext

/FundaOzturk,VerdaYasar,PnarYldz,HaticeCebeci, SerapAbanozoglu,

....,
DouglasLankin, DanielWebber,CnurYavaslo /

m **Ders Gunleri {The Days of Courses}**

$ontext

{Enter the days of week which have courses between the slashes /.../

Examples: /Monday, Tuesday,.....,Friday/ .}

$offtext

/Pazartesi,Sali,Carsamba,Persembe,Cuma/

n **Ders Saatleri {Time-Slots of Courses}**

$ontext
{Enter the time slots from the first to the last number of slots between the slash /..../.
For explaining regular series you can use * , or you can write one by one using
comma between each one.

For Example: /1*8/ or /1,2,3,4,5,7,8/ .}
$offtext

/1*8/

k **Siniflar {Classes or Classrooms}**

$ontext
{Classes or classrooms of the students set elements are entered between the slashes
/..../ one by one, put comma between each one .

For example : /6a,6b,7a,7b,7c/ .}
$offtext

/1a,1b,1c,2a,2b,2c,3a,3b,3c,3d,4a,4b,4c,4d,5a,5b,5c,5d, 6a,6b,6c,6d,7a,7b,7c,8a,8b/

l **Branslar {Courses}**

$ontext
{The element of course set is entered between the slashes /..../ one by one,
put comma between each one .

For example : /Mathematics,Science,English/ .}
$offtext

/Snf,EtutA,Etut,Almanca,Muzik,Resim,Is,Bilgisayar,Beden,
Turkce,RehSE,Fen,Trafik,Matematik,Sosyal,Vatandaslk,TCInk,
DinKulturu,Ingilizce/

$ontext
{ This part for the activity periods, e1(k) is Monday activity set if you enter the
classes which have activities then the last two time-slots on Monday will be idle

for you at the timetable. e2(k) is tuesday activities and e3(k) is Thursday activities. Similar way you can add or arrange the program according to you need if you check **{Activity time-slots control}** part.

If you do not need any separation delete this part and **{Activity time-slots control}** part. Or you can close each line using (*) at he beginning of the lines.}

$offtext

e1(k) etkinlik1 /1a,1b,1c,2a,2b,2c,3a,3b,3c,3d/
e2(k) etkinlik2 /4a,4b,4c,4d,5a,5b,5c/
e3(k) etkinlik3 /6a,6b,6c,6d,7a,7b,7c,8a,8b/ ;

alias(n,n1,n2);

parameter a(i,k,l) **ogretmen sinif brans atamasi**

{Assignment of teacher class and course}

$ontext

{This part is for the assignment of the teachers, class and course, enter the name of teacher, class and course, put comma between each one then leave space and enter the number of time slots belong to the course in a week. To enter another one; put comma and go on. All these entries will be done between the slashes /.../.

For example : / name of teacher1.class.course the number of periods,
name of teacher2.class.course the number of periods,
....,
name of teachern.class.course the number of periods/

$offtext

/FundaOzturk.1a.Snf 27,
VerdaYasar.1b.Snf 27,
PnarYldz.1c.Snf 27,
....

....

....

....

```
DanielWebber.8a.Etut 2, DanielWebber.8b.Etut        2,
CNurYavaslol.5b.Etut 2, CNurYavaslol.5c.Etut         3     / ;
```

parameter bs (k,l) **limit blocks**

```
       /
       1a.Beden  1,
       1b.Beden  1,
       1c.Beden  1 ,
       2a.Beden  1,
       2b.Beden  1,
       ..................
       .................
       ................
       8b.Beden  1,

                /;
$ontext
```

{If you want to have block lessons, then you can use this part.

For example: bs(‘class’,’coursename’)=1, this make 1 block lesson.
bs(‘class’,’coursename’)=2, this make 2 block lesson.

If you write bs(k,’coursename’)=1, k refer to all class so this course will be done as 1 block lesson for all classes which has the course. Do not forget the put semicolon after each assignment.}

```
$offtext

   bs(k,'Is')=1 ;
   bs(k,'Matematik')=1;
   bs(k,'Fen')=1;
```

parameter p(i,m,n) **ceza parametresi**; **{Penalty Parameters}**

$ontext

{This part is about penalty parameters that teacher or administer can do a preference for each teacher, day and the periods of course. Big number show unwanted day and period small one show wanted day and period. You can use in range 1 to 8 coefficients or if you want you can use another range. According to this coefficient the most suitable timetable will be arranged for you.

For example: p('teachername','day','periods')=coefficient;

If you have got any preference than you must write p(i,m,n)=0.}

$offtext

parameter sr(l) **özel odaların (labaratuvar...) sayısı** ;

sr("Bilgisayar") = 2;

p('DanielWebber',m,n)=1;
p("BayhanTemur",m,n)=**ord**(n);
* p(i,m,n)=0;

scalar D /10/;

variables
x(i,m,n,k,l) **Zaman Tablosu {TimeTable}**
z **Toplam zaman yerlesimi {Total time Allocation}**
y(i,m) **Yarim gun {Half Day Assignment}**
b(l,m,n1,k) **n1 başlangıç saati olan blok ders ; {the lesson which starts from n1th period}**
binary variable x, y , b;

$ontext
{this part is about the controlling starting periods of block lessons. You can chose any periods that you do not want to start the block.}
$offtext

b.fx(l,m,"5",k)=0; **{block lessons cannot starts from 5th period}**
b.fx(l,m,"8",k)=0; **{block lessons cannot starts from 8th period}**

{Activity time-slots control }
x.fx(i,"pazartesi",n,k,l)$((ord(n)gt 6)and e1(k)) = 0;
x.fx(i,"sali",n,k,l)$((ord(n)gt 6)and e2(k)) = 0;
x.fx(i,"persembe",n,k,l)$((ord(n)gt 6)and e3(k)) = 0;

x.fx(i,m,n,k,l)$(a(i,k,l)=0) = 0;

equations

Obj **Hedef Fonksiyonu**,
BigD(i,m) **Yarimgun çarpanı**,
Yarimgun(i) **Her ögretmenin bir ögleden sonrası boş yarım gun olacak,**
Blok(l,m,n1,k) **Bir bransın bir günde bir sınıfa blok dersi olması,**
CheckBlocks(l,k) **k sınıfı için l branşında blok ders sayıları**,
Kisit3(i) **Ogretmenlerin ders saati toplami,**
Kisit4(m,k,l) **Bir gunde bir bransla ilgili en fazla 2 ders yapilir**,
Kisit5(i,m,n) **Bir ogretmen ayni anda sadece bir ders yapabilir**,

Kisit6(k,m,n) **Bir sinif ayni anda sadece bir ders yapabilir,**
Kisit7(m,n,l) **Ayni anda toplam lab sayisi oda sayisini asamaz;**

```
Obj ..  z =e= sum((i,m,n) , p(i,m,n)* sum((k,l)$(a(i,k,l)>0) , x(i,m,n,k,l)  ));
Yarimgun(i)..  sum(m,y(i,m))=e=1;

BigD(i,m)..sum((n,k,l)$((ord(n) gt 5) and (a(i,k,l)>0)),x(i,m,n,k,l))=l=(1-y(i,m))*D;

Blok(l,m,n1,k)$(bs(k,l) and sum(i,a(i,k,l)) >0 ) ..
        sum(i$(a(i,k,l)>0), x(i,m,n1,k,l)+x(i,m,n1+1,k,l)) =g= 2* b(l,m,n1,k);

CheckBlocks(l,k)$(bs(k,l) and sum(i,a(i,k,l)) >0 ).. sum((m,n1), b(l,m,n1,k)) =e=
bs(k,l);

Kisit3(i) ..    sum((m,n,k,l)$(a(i,k,l)>0), x(i,m,n,k,l)) =e= sum((k,l),a(i,k,l));

Kisit4(m,k,l)$(not sameas (l,"SNF"))..  sum((i,n)$(a(i,k,l)>0), x(i,m,n,k,l)) =l= 2;

Kisit5(i,m,n)..  sum((k,l)$(a(i,k,l)>0),x(i,m,n,k,l))=l=1;

Kisit6(k,m,n)..  sum((i,l)$(a(i,k,l)>0),x(i,m,n,k,l)) =l= 1;

set IL(I,L);
IL(I,L) = yes$(sum(k,a(i,k,l)) >= 1 );

Kisit7(l,m,n)$(sr(l)>0)..  sum((i,k)$IL(i,l), x(i,m,n,k,l)) =l= sr(l);

option limrow=5,limcol=0;
option mip=cplex;
option reslim=10000;
model timetabling /all/;
timetabling.holdfixed=1;
solve timetabling using mip minimizing z ;
option x:1:3:2; display x.l;
display y.l;
display b.l;

display IL;
file     zamancizelgesi /cizelgebig.txt/;
 put     zamancizelgesi              ;
zamancizelgesi.nw=5;
put     'Özel.......Okulları   ';
put     'HAFTALIK OKUL GENEL DERS CİZELGESİ'//;
parameter counter;

loop (m, put m.tl':'
```

```
    for (counter=1 to 8 ,
         put @(20+counter*5) counter:4:0
       );
      put /
 put'-----------------------------------------------------------------'/;
      loop(i,
       loop(l$IL(i,l), put i.tl , l.tl
         loop(n,
           loop(k$(x.l(i,m,n,k,l)=1), put @(23+ord(n)*5) k.tl  )
         )
          put /;
       )
     )
 put /;
 );

file zamanogretmen/table4.txt/;
put zamanogretmen

loop(i, put'Ogretmen : '  put i.tl '  HAFTALIK DERS CİZELGESİ'/

   loop(m, put@(10+ord(m)*11) m.tl)
      put/
 put'-------------------------------------------------------------------'/;

      loop(n,  put n.tl

         loop(m,

          loop(l$IL(i,l),

            loop(k$(x.l(i,m,n,k,l)=1),  put @(10+ord(m)*11) k.tl:3,l.tl:3)
          )

        ) put/;

     ) put/;

 );

file  zamanogrenci/table5.txt/;
put zamanogrenci

loop(k, put'Sınıf : ' put k.tl'  HAFTALIK DERS CİZELGESİ'//

    loop(m, put@(ord(m)*11) m.tl)
      put /
```

```
put'------------------------------------------------------------------'/;

        loop(n, put n.tl

           loop(m,

             loop(i,

                loop(l$(x.l(i,m,n,k,l)=1), put@(ord(m)*11) l.tl)
                )

           ) put/;
       ) put/;
);
```

APPENDIX C GENERATED TIMETABLES

Timetables:

I) A Sample Part General Timetable Of The School

Özel.......Okulları HAFTALIK OKUL GENEL DERS CİZELGESİ

Pazartesi	:	1	2	3	4	5	6	7	8
FundaOzturk	Snf	1a		1a	1a	1a			
....									
EbruUlutas	Snf		3c	3c	3c		3c		
....									
BayhanTemur	Matematik	6c	5b	5b					
....									
AzizBudak	DinKulturu		6c	6c	8a		8b	7b	5b

Sali	:	1	2	3	4	5	6	7	8
FundaOzturk	Snf			1a	1a	1a	1a		
....									
EbruUlutas	Snf	3c	3c	3c	3c		3c	3c	3c
....									
BayhanTemur	Matematik	6c	6a	6a					
....									
AzizBudak	DinKulturu		5a	5c	7c		5d	7c	

Carsamba	:	1	2	3	4	5	6	7	8
FundaOzturk	Snf	1a	1a				1a	1a	1a
....									
EbruUlutas	Snf			3c	3c	3c			
....									
BayhanTemur	Matematik	7c	7c	6a					
....									
AzizBudak	DinKulturu	5a		7a		6a	6a	8b	

Persembe	:	1	2	3	4	5	6	7	8
FundaOzturk	Snf	1a	1a			1a	1a	1a	1a
....									
EbruUlutas	Snf	3c	3c	3c			3c		3c
....									
BayhanTemur	Matematik	6a	7c	6c	6c				
....									
AzizBudak	DinKulturu	7b	7a			5b			

Cuma	:	1	2	3	4	5	6	7	8
FundaOzturk	Snf	1a	1a	1a	1a	1a	1a	1a	1a
....									
EbruUlutas	Snf	3c	3c	3c	3c	3c	3c	3c	3c
....									
BayhanTemur	Matematik	5b	7c	5b					
....									
AzizBudak	DinKulturu	8a	6b		5d	6d	5c	6b	6d

II) Sample Timetables For Classes

Sınıf : 5a HAFTALIK DERS CİZELGESİ

	Pazartesi	Sali	Carsamba	Persembe	Cuma
1	Etut	Sosyal	DinKultur	Trafik	EtutA
2	Fen	DinKulturu	EtutA	Almanca	Etut
3	Fen	Turkce	Turkce	Is	Sosyal
4	Matematik	RehSE	Fen	Is	Matematik
5	Is	Beden	Sosyal	Bilgisayar	Matematik
6	Turkce	Etut	Matematik	Etut	Fen
7	Turkce		Muzik	Turkce	Etut
8	Resim		Almanca	Turkce	EtutA

Sınıf : 7c HAFTALIK DERS CİZELGESİ

	Pazartesi	Sali	Carsamba	Persembe	Cuma
1	Turkce	Almanca	Matematik	Turkce	Turkce
2	Sosyal	Is	Matematik	Matematik	Matematik
3	Almanca	Is	Beden	Etut	EtutA
4	Vatandaslk	DinKulturu	Beden	Fen	Turkce
5	Sosyal	Etut	Almanca	Etut	Resim
6	EtutA	Etut	Muzik	Bilgisayar	Almanca
7	Turkce	DinKulturu	Etut		Fen
8	EtutA	Ingilizce	Sosyal		Fen

III) Sample Timetables For Teachers

Ogretmen : YesimSacbagl HAFTALIK DERS CİZELGESİ

	Pazartesi	Sali	Carsamba	Persembe	Cuma
1					
2	8b Etu	7b Alm	6c Etu	7b Alm	6d Etu
3			8b Etu		8b Alm
4				7b Alm	
5	8b Alm	8b Alm	6c Alm		
6		8b Alm	6c Alm	8b Etu	
7	6c Alm				
8	6c Alm	7b Alm			

Ogretmen : PAkarsu HAFTALIK DERS CİZELGESİ

	Pazartesi	Sali	Carsamba	Persembe	Cuma
1		4c Is	7b Is		
2		7c Is	7b Is		
3	6a Is	7c Is	4c Res		
4	6a Is	6a Res	1a Res	3c Res	8b Is
5			1a Res	3c Res	8b Is
6	6d Is		4c Is		2a Res
7	6d Is	8a Is	4c Is		2a Res
8		8a Is			

Ogretmen : OzcanAkkanat HAFTALIK DERS CİZELGESİ

	Pazartesi	Sali	Carsamba	Persembe	Cuma
1	7a Bed	1a Bed			
2	7a Bed	1a Bed		3b Bed	6c Bed
3	4b Bed		6b Bed	3b Bed	6c Bed
4			6b Bed		
5		5a Bed			
6				2a Bed	
7		3a Bed		2a Bed	
8		3a Bed			

Ogretmen : OzgeTurhan   HAFTALIK DERS CİZELGESİ

	Pazartesi	Sali	Carsamba	Persembe	Cuma
1		5d Etu	5c Tra		5d Tra
2		5d Fen	5d Fen	6a Fen	5d Reh
3	5d Etu	5d Fen		6a Fen	
4			5c Etu	5d Fen	
5	6a Fen				5c Etu
6			5c Fen		
7			5c Fen		5c Fen
8					5c Fen

Ogretmen : BayhanTemur   HAFTALIK DERS CİZELGESİ

	Pazartesi	Sali	Carsamba	Persembe	Cuma
1	6c Mat	6c Mat	7c Mat	6a Mat	5b Mat
2	5b Mat	6a Mat	7c Mat	7c Mat	7c Mat
3	5b Mat	6a Mat	6a Mat	6c Mat	5b Mat
4				6c Mat	
5					
6					
7					
8					

CURRICULUM VITAE

Name : Bayhan Temur

Address : Hamidiye Mh. Ataturk Cd. Ogretmenler ST. 3042 A1 D14 34782
ÇEKMEKÖY-İSTANBUL

Work Address : OZEL ALEV OKULLARI Ayvalı Ciftligi Mevkii
Omerli-Umraniye/ISTANBUL

Telephone : 0216 6424440
0532 2574807

E-mail : bt0185@yahoo.com

Personal Information

Birth Place : Artvin—Şavşat
Birth Date : 01.11.1968
Sex : MALE
Nationality : Turkish
Marital Status : Married
Blood Group : A RH +
Military Service : Discharged 22 March 1998

Educational Background

Graduated From: Boğaziçi Un.Dept.of Teaching Mathematics-ISTANBUL

High School Education: Kabataş Erkek High School Science Department-ISTANBUL

Primary School Education : Ismetpasa Primary School - ORDU