TECHNOLOGY

Technology, 13(3), 159-172, (2010)

SOLVING OF SCHEDULING PROBLEM WITH HEURISTIC OPTIMIZATION APPROACH

Abdullah ELEN[*], İbrahim ÇAYIROĞLU[**]
[*] Karabuk University, Graduate School of Natural and Applied Sciences, Karabuk, TURKEY
[**] Karabuk University, Engineering Faculty, Karabuk, TURKEY

Abstract

In this study, solution of course scheduling problems in student affairs automation used in the universities was carried out using Genetic Algorithm method which is one of Heuristic Optimization approaches. Scheduling problem is the process of placing the works to be done in time intervals at optimum level within the constraints. The solution of this problem at universities becomes impossible with analytic methods since solution space is too wide and there are many constraints. Therefore, giving the best solutions for this type of problems, Genetic Algorithm method was used with a new approach by examining the parameters giving the best results.

The performance of the Algorithm was measured by conducting applications over the actual data from the university (504 instructors, 4163 course, 203 classrooms and 10525 students).
Algorithm was developed using ASP.NET, C# programming language and SQL Server Database. Interface and reporting pages of student affairs automation were also added to the program for practical use of the Algorithm.

Key words: Heuristic approach, Artificial Intelligence, Genetic Algorithm, Time Scheduling

1. Introduction

Scheduling problem is the process of placing the works to be done in time intervals at optimum level within the constraints. Weekly course and exam programs are one of the scheduling problems at universities.
Very high probabilities result when calculations of probabilities of course placement are made at a university. With an approximate calculation, placement of all courses in weekly schedule in a faculty where 3-hour 240 courses are present results in a quite high probability calculation like 400240. Moreover, when necessary containing data are added to this process, the problem becomes even more complicated. Therefore, it is almost impossible to solve such problems with conventional methods and mathematical algorithms.

Time-Scheduling problems have been studied since 1960 by many researchers. The reason is that the time scheduling problems in education is a vast field. Solution approaches also vary according to the size of the problem. However, a combination problem that must be solved lies at the heart of this entire scheduling problem. Therefore, the techniques known for discrete optimization problems might also be used in time scheduling problems [1].

Discrete optimization problems might be called “in general, easy-to-define, hard-to-solve problems”. It is generally hard to express these problems as a certain mathematical function. Therefore, it is difficult to solve this type of problems by classic optimization methods. Heuristic algorithms which generate simple solutions are needed for a solution. Based on genetic science and natural selection, genetic algorithms are this type of algorithms, they have been used for various discrete optimization problems and good results have been obtained [2].

Genetic algorithms generate a solution set of different solutions rather than a single solution to the problems.

Corresponded Author;aelen@karabuk.edu.tr

Thus, many points are evaluated at one time in searching space and consequently the probability of reaching holistic solutions rises. The solutions in solution set are completely independent from one another. Genetic algorithms imitate the evolutionary process in computer environment for the solution of the problems.

In this study, an algorithm was developed using actual data of 2009–2010 within Karabük University (504 instructors, 4163 courses, 203 classrooms and 10525 students) and application details have been indicated. The objective of the prepared program is to eliminate the challenges experienced during preparation of course and exam schedulings at universities, to lighten workload on instructors, to ensure efficient use of resources, and to raise pedagogical efficiency considering the satisfaction of students and instructors.

The first application study of genetic algorithms to scheduling problem was conducted by Davis in 1985. Liepins et al., have researched the scheduling problem of works with certain delivery time and processing time in 1987. This problem is referred to as the simplest scheduling problem. In 1993, Gupta et al. published their study on single machine model which aimed to minimize the flow time. Lee and Kim presented their study which also involved penalty for delay and sagging in 1995. Cheng et al. published their study on the model composed of identical, parallel machines in the same year [3]. Apart from that, Biegel and Davern carried out a study for workshop schedulings in 1990, Badami and Parks conducted a study for flow workshop problem 1991, Vancza and Markus has a published study on process planning problem in 1991 [4]. In general, genetic algorithms have found solutions close to optimal for scheduling problems. Solution times were much faster compared to other solution methods.

Among the studies conducted lately, AL-Milli used a new hybrid algorithm based on Genetic Algorithm and Great Deluge algorithm in the solution of course scheduling problem [5].
In their study, Wijaya and Manurung produced a solution successfully meeting all the constraints of a scheduling problem modeled with actual constraints for Indonesia University Faculty of Computer Sciences using Genetic Algorithm [6].

In their study, Jat and Yang obtained satisfactory experimental results in the solution of university course scheduling problem using a research strategy guided with Genetic Algorithm and a local searching technique [7].

Atanak and Hocaoğlu developed an automation which uses the size of courses, number of courses, the number students who should be registered for the course and instructors who will give the lectures as input and which prepares the schedule under the constraints given [8].

In their study, Bratkovic et al. handled a rather limited laboratory sample and used it in the solution of time scheduling problem with genetic algorithm [9].

Aldasht et al. used evolutionary algorithms similar to genetic algorithms in the solution of university course scheduling problem in their study. It resulted in more relevant results compared to those from manually prepared schedules [10].

Aydın used genetic algorithms in solution of university course scheduling problem in his study. Using actual data from Bahcesehir University, Faculty of Science and Letters, he obtained optimum results without ignoring any constraint [11].

In the study carried out by Kalender, an application preparing automatic course scheduling program in which solutions could be generated and changed, data could be kept, the application was intended to be used in Yeditepe University Computer Engineering with super-heuristic algorithms [12].

Juang et al. developed adaptive computer-aided scheduling system with Genetic Algorithm for planned employee training programs for training and appointment of employees in machine industry [13].

In his study, Baç developed a new mathematical model providing efficient solutions of the problem of making academic schedules. However, the model developed by him requires use of a computer with high processing power or work stations [14].

In the study conducted by Temiz and Erol, the solution of m-machine flow type scheduling problems was handled using genetic algorithm [15]. Dealing with the situations in which time parameters such as process time and deadlines are undefined, the objectives such as maximum delay time and total flow time were intended to be optimized. It was indicated that the algorithm generated efficient solutions at a reasonable time for medium and large scale problems.

Gülcü used genetic algorithm and taboo search algorithm in order to solve combinatorial optimization problems in his study. Weekly schedules were prepared for educational institutions and it was found that genetic algorithms resulted in good results for this type of problems [16].

Datta et al. used a multipurpose evolutionary algorithm method in the solution of course-time scheduling problem in their study. The solved problem is rather demanding and complicated since majority of the courses is divided or grouped. Obtained results were considered to be better than the ones prepared manually [17].

Çayıroğlu and Dizdar developed an expert system-based course placement application in their study. Prepared program gives all the necessary documents by preparing whole weekly schedule in such a manner that no conflict will occur [18].

In a study, Yiğit established weekly schedules for vocational high-schools with the aid of genetic algorithms [19]. In this study, “repair operator” was used for eliminating the disorders formed in the structure of chromosomes after mutation operator used in genetic algorithm.

Dammak et al. obtained successful and basic results by developing a simple heuristic procedure in the solution of a nonconflicting exam assignment problem which is appropriate for the classrooms with certain capacities [20].

In their study, Kazarlis et al used Micro-GA’s and Developed Genetic Algorithm method including heuristic local search operators in the solution of University time scheduling problem [21]. In the study conducted by Biroğul, workshop scheduling problem was solved by means of genetic algorithm. Gantt chart was drawn according to workshop details entered [22].

In the study carried out by Head and Shaban, the heuristic function based system developed by them for simultaneous course/student scheduling was tried at United Arab Emirates University, it was found that physical and human sources of the university were used at an optimal level [23].

2. Material and Method

2.1. General Structure of Student Affairs Automation

Student affairs automation is one of the most important automations used in a university. All of the works performed by thousands of students and hundreds of instructors every year are carried out over this automation. Scheduling problem performed in the scope of this study is used at two points within this automation. These points are placing the courses in weekly schedules and setting the dates and times of the exams. For the developed algorithm to use actual data and contribute to the practical life, it is required that the data needed by algorithm have been entered and the outputs generated by the algorithm are reported. For this purpose, student affairs automation in which algorithm will be installed has been programmed and integrity has been ensured. Otherwise, it cannot be possible to benefit from practical results of the algorithm developed.

Student affairs automation is composed of three modules. These modules are Administrative module, Course Module and Exam Module. Links and queries between the tables were formed by forming the database structure of these three modules. Data were saved on SQL Server Database using C# programming language in ASP.NET environment. The codes generated by the algorithm were programmed as a Windows Application. Software was developed using a dynamic menu system as object-based. If the codes used in inquiring the data are used extensively, it has been coded as Stored Procedure inside database, if used less it has been coded as object. All subprocesses were programmed using dynamic parameters in object structure in order to minimize the number of codes. Data flows and connections are indicated below on database of these three modules.

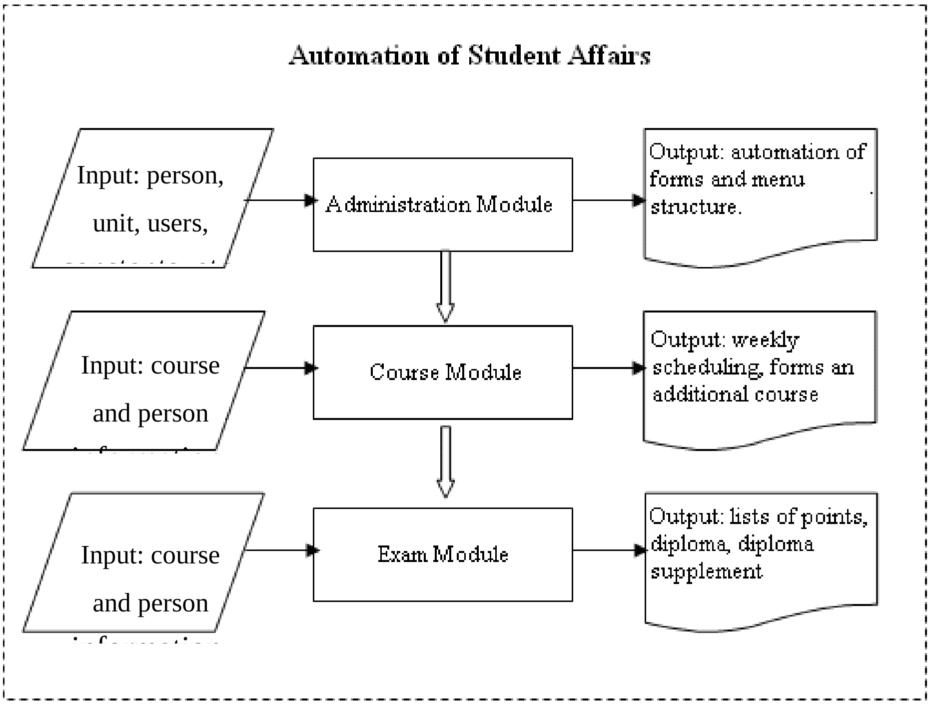

Figure 1. General Structure of Student Affairs Automation Developed

<u>Administrative Module:</u> Administrative module is the part in which basic data are saved in saving system of the program and which involves interface and admin administration parts. This module is composed of sub-divisions such as Units, Individuals, Users, Menus, Constraints, Classrooms and Academic Calendars.
<u>Course Module:</u> This is the module for performing basic processes such as creating curriculum for a Department, opening course, assigning instructors.
<u>Examination Module:</u> Processes of students' selecting their courses, setting examination dates of the courses and assigning proctors to these examinations, entering their grades and getting diploma and attachments as reports are carried out in this module.

2.2. Design of Genetic Algorithm Model

Course/Time scheduling is a hard problem required to be solved for each academic term at the universities. The problem might vary from an institution to another since universities have different structures, needs and priorities. In this study, a GA model was designed using actual data from Karabük University.

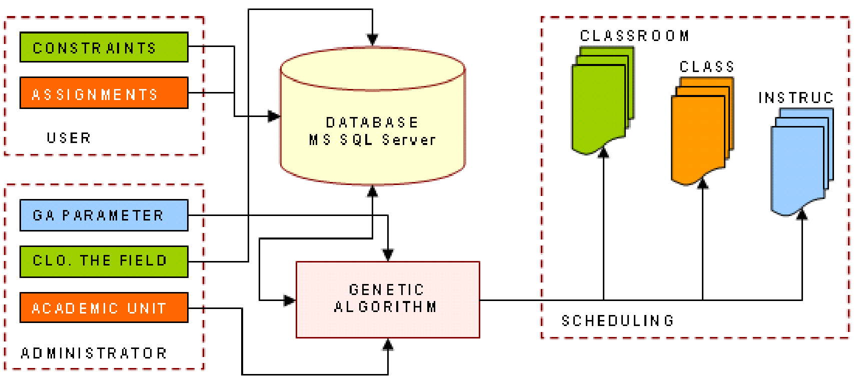

Figure 2. General Structure of the Developed Model

Structure of Gene

A gene stores seven different bits of information (Figure 3). Among these, G1, G2, G3 and G4 which are indicated with blue color are predefined information and cannot be changed by genetic operators. Indicated with green color G5 represents the classrooms belonging to relevant course which is also predefined. However, they can be changed partly by mutation operator (course-classroom relationship). G6 and G7 represent the places where day and time details are kept.

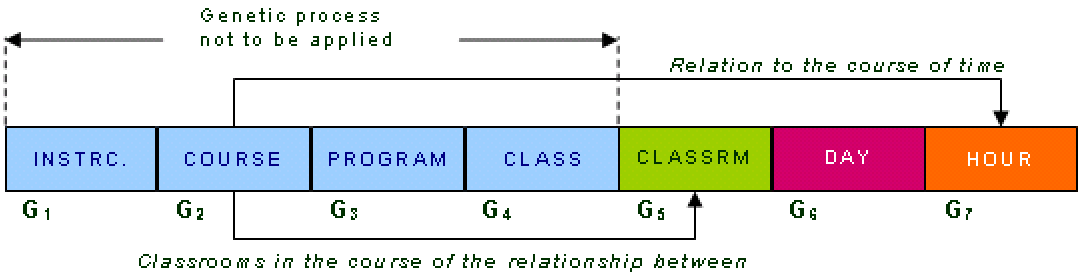

Figure 3. Structure of Detailed Gene

Structure of Chromosome

Since every academic unit at the university might open different number of courses, genetic parameters have been designed dynamically (Figure 4). For example, while Faculty of Engineering defines 150 courses for fall term, Faculty of Science and Letters might define 120 courses. Therefore, the number of genes in the chromosome will also be different.

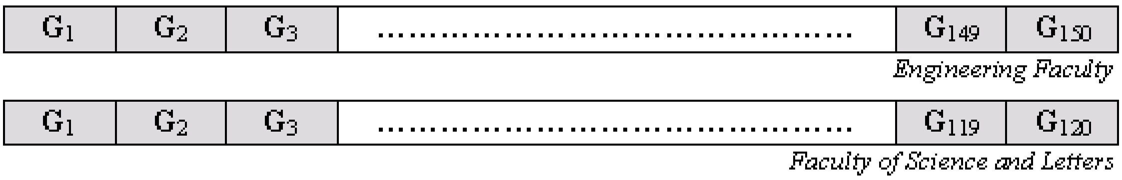

Figure 4. Number of Different Gene Have Chromosomes

Constraints Used in Solving the Problem

Weekly course scheduling constraints are required to be arranged according to the needs of the universities. Constraints were formed based on improving education quality, improving the performance of students and instructors, meeting requests of instructors and responding to administrative requirements [19].

Education/Training Productivity

According to pedagogical principles, ability of students to understand varies according to the days. Accordingly, education-training productivity levels differ. The first and last days of the week are generally the days on which productivity is less [25]. Days were given weight values considering the productivity curve according to the days indicated in Figure 5 for calculating fitness value of the days. Also, according to pedagogic researches, productivity levels of working hours within the day also differ. The productivity is low in the first and last parts of the working hours of the day. Working hours were given weight values considering the productivity curve according to the working hours of the day indicated in Figure 5 for calculating fitness value of the working hours. In the graph, number of parts of working hours was taken as 9 [24].

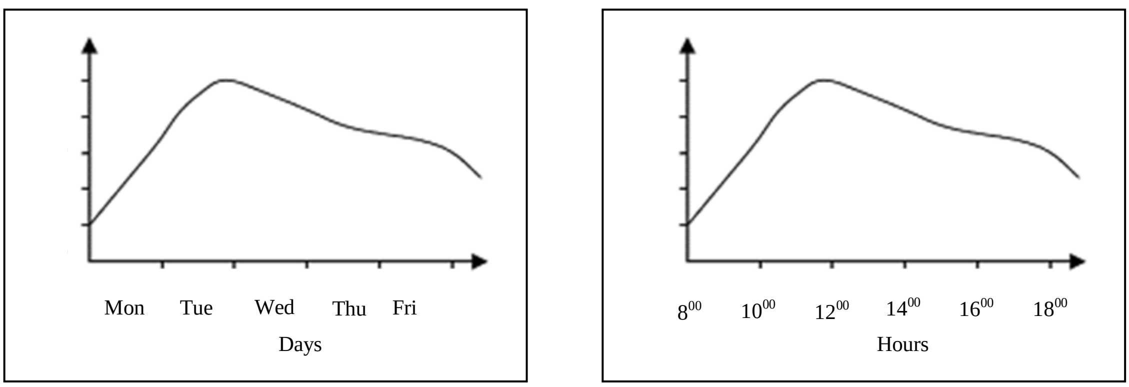

Figure 5. Pedagogical Day and Hour Productivity

Instructor Satisfaction

Instructors' setting the day and hour they want to teach their lessons help form appropriate times for academic study, consultancy, administrative duties etc. A constraint chart was formed for the satisfaction of the instructor as indicated in Figure 6. The instructor is asked to make a selection over the constraint chart as many as the number of the courses given by the instructor. Constraint point is calculated according to the days and hours detected and fitness value of the chromosome is updated.
In addition, the instructors will be able to use Area Closure constraint on the days and hours as allowed by the administration of the university.

ÇİZELGELEME KISITLARI (Yrd.Doç.Dr. İbrahim ÇAYIROĞLU)

GÜN / SAAT		PAZARTESİ	SALI	ÇARŞAMBA	PERŞEMBE	CUMA
1	08:00 - 08:45					
2	08:00 - 08:45					
3	08:00 - 08:45	İstenen Alan	İstenen Alan	İstenen Alan	İstenen Alan	
4	08:00 - 08:45	İstenen Alan	İstenen Alan	İstenen Alan	İstenen Alan	
5	08:00 - 08:45	İstenen Alan	İstenen Alan	İstenen Alan	İstenen Alan	
6	08:00 - 08:45	İstenen Alan	İstenen Alan	İstenen Alan		
7	08:00 - 08:45					
8	08:00 - 08:45					
9	08:00 - 08:45					
10	08:00 - 08:45					

Figure 6. Schedule of Instructor Satisfaction

Weight of Lesson Load

Distribution of weight of lesson load might be set in three different ways:

1. Dense: In some academic units, this type of placement might be preferred since number of department/program are excessive compared to physical building conditions and/or there are few classrooms.

2. Balanced: It ensures distributing the number of lessons of students and instructors to every day of the week equally as much as possible.

3. Free: Apart from two situations stated above, lesson placement process might be carried out without controlling the weight of any lesson load. Briefly, when such a selection is made, no constraint point will be calculated for lesson load weight.

Meeting Times of Splitted Courses

It might be demanded to split a course as theory, application and laboratory or split it into groups, or different instructors might be assigned to each part of this course. In such cases, it might be demanded that placement times are on different days or on the same day for splitted courses.
Distribution of placement times for splitted courses might be set in three different ways:

1. Different Day: This type might be preferred for assigning placement times to different days for splitted courses.

2. Same Day: This type might be preferred for assigning placement times to the same days for splitted courses.

3. Free: Apart from two situations stated above, lesson placement process might be carried out without controlling any constraint. Briefly, when such a selection is made, no constraint point will be calculated for placement times of splitted courses.

2.3 The Developed Algorithm Interface

It was developed using C# programming language on Microsoft Visual Studio .NET platform. Main screen of the system is indicated in figure 7. On the main screen, there are labels indicating predefined parameter values, change values that occur during process, and bootstrap values. During the operation of the system, real time graphics of success rates calculated are drawn.

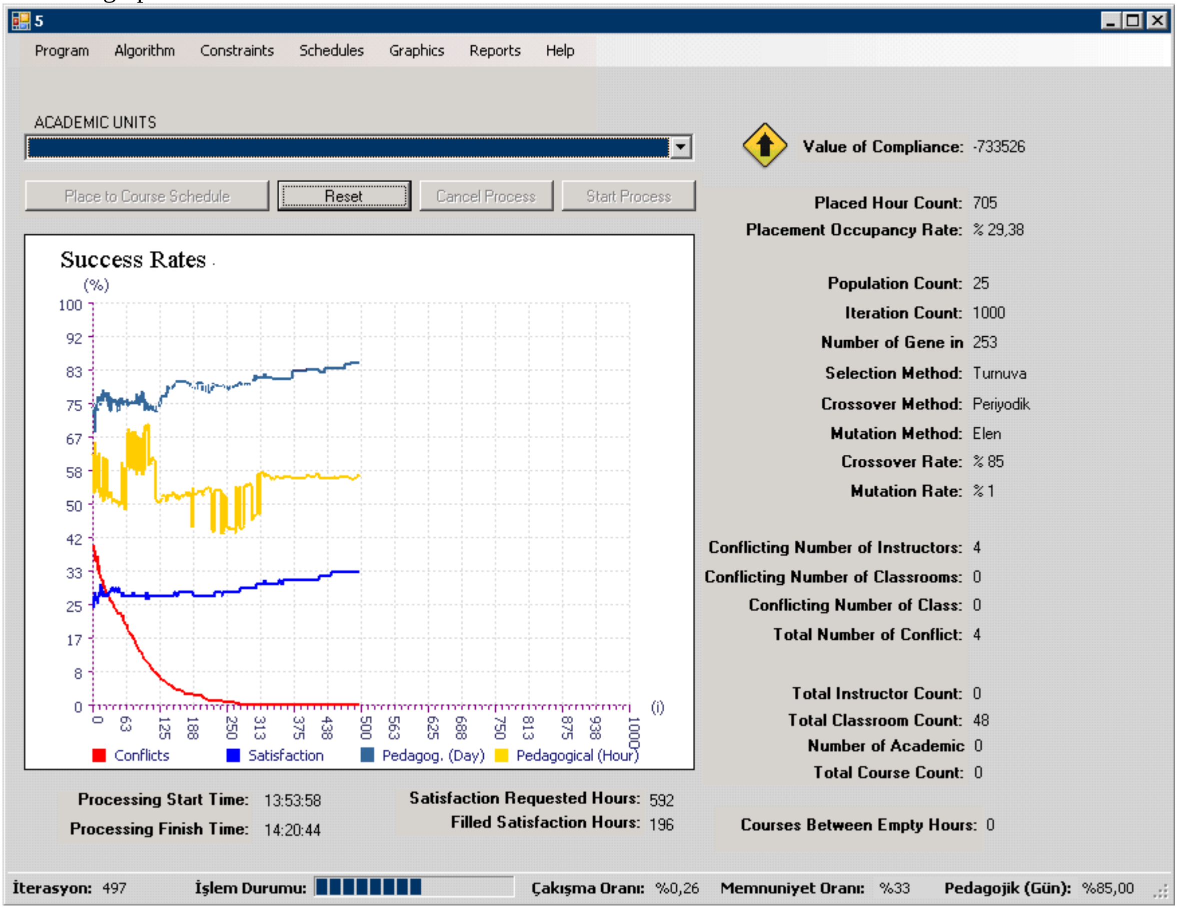

Figure 7. Application Interface

Firstly, relevant unit is selected from “Academic Units” multiple choice box (Faculty, Institute, College and Vocational High School) for operating the system. Then, parameter values of GA are set from Algorithm menu according to the unit selected. Next process step is constraints. “Configure” tab is clicked under constraints menu. Required adjustments are made considering the demands of the units. Thus, system will be made ready to run. Finally, System is initiated by clicking on Start button.

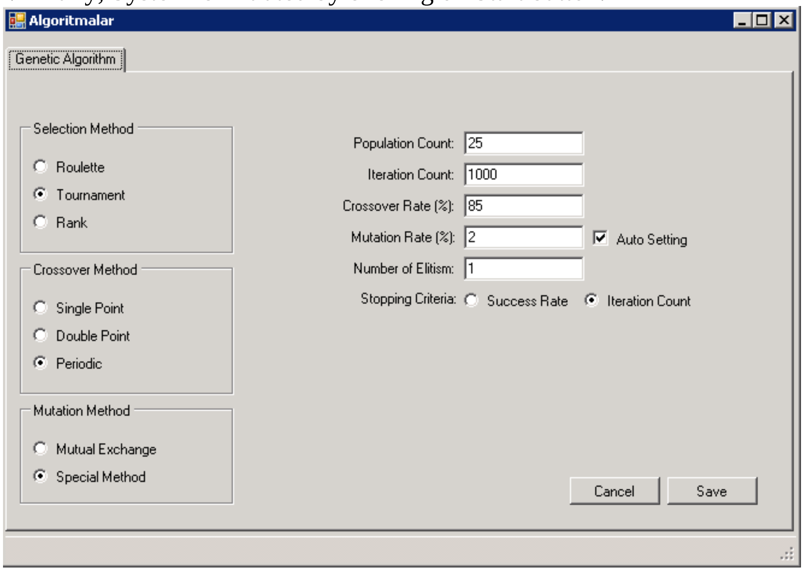

Figure 8. Selection of GA Parameters

The screen on Figure 8 is used in setting operators and parameter values of GA. Three most used preferences, Roulette, Tournament and Rank might be selected as Selection Method. Also three different selections are available in Crossing Method which are Single Point, Double (Two) Point and Periodic. It might be applied in two different ways as Intermutation and Special Method in Mutation Method.

Mutation rate might be controlled automatically by the system in parameter values. The reason is that the best chromosome value remains fixed during a certain iteration term. Finally, three different interruption criteria are present. These are called Iteration Number, Success Rate and operation time operator request.

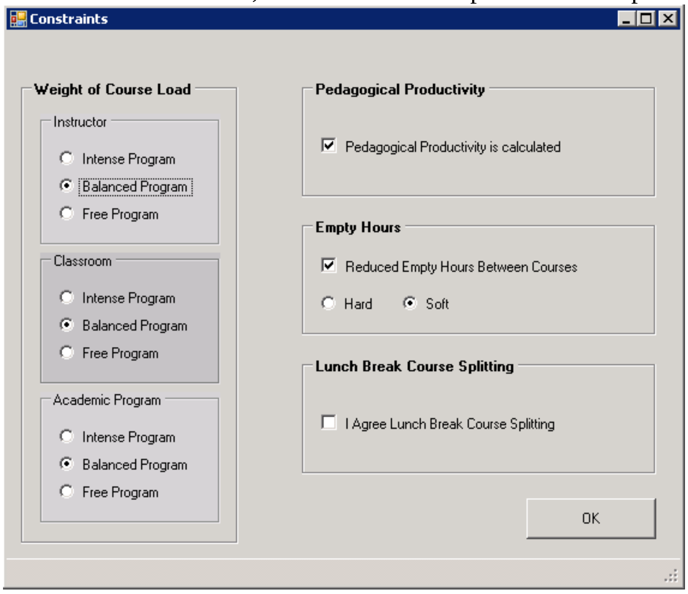

Figure 9. Constraint Identification Screen

On constraint identification screen indicated in Figure 9, separate definitions of weight of lesson load might be made for instructor, classroom and academic program. As stated before, they might be applied in three different ways as Dense, Balanced and Free programs. Selection is made from Pedagogical Productivity group in order to define whether Pedagogical Day and Hour productivity will be taken into consideration. Two different punishments, severe and soft punishment, might be applied in minimizing the idle time between the courses. The idle time between the lessons is minimized when severe punishment is applied. During forming course schedules, it is required to check from Lunch Break Course Splitting group in order to allow lunch break splittings of the courses. After the processes regarding constraint definitions, “Ok” button is clicked.

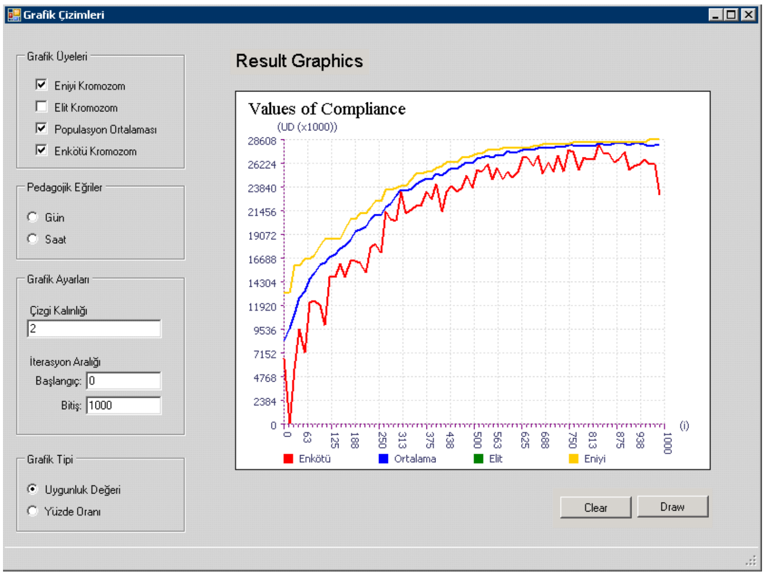

Figure 10. Result Graphics

After scheduling process is completed, genetic values and pedagogical day and hour values might be drawn individually as seen from result graphics on Figure 10. From graphic settings group, line weight and iteration range intended to be displayed might be set. The type of the value to be displayed in each iteration is determined from graphic type group.

After course scheduling process is completed, the best result obtained with Genetic Algorithm is saved in database clicking on "Place on Course Schedule" button from the main screen. Three different types of schedule are formed in a web-based environment as a result of this process: Instructor, Classroom and Academic Program.

Yrd.Doç.Dr. İbrahim ÇAYIROĞLU						
GÜN / SAAT		PAZARTESİ	SALI	ÇARŞAMBA	PERŞEMBE	CUMA
1	08:00 - 08:45					TLU401 (T) - MOB-2 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0420
2	08:00 - 08:45					TLU401 (T) - MOB-2 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0420
3	08:00 - 08:45	BLG211 (T) - ANFİ 2 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0425				TLU401 (T) - MOB-2 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0420
4	08:00 - 08:45	BLG211 (T) - ANFİ 2 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0425				
5	08:00 - 08:45	BLG211 (T) - ANFİ 2 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0425				
6	08:00 - 08:45			BLG403 (T) - ANFİ 10 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0425		
7	08:00 - 08:45			BLG403 (T) - ANFİ 10 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0425		BLG319 (T) - ANFİ 8 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0425
8	08:00 - 08:45			BLG403 (T) - ANFİ 10 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0425		BLG319 (T) - ANFİ 8 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0425
9	08:00 - 08:45					BLG319 (T) - ANFİ 8 [3] Yrd.Doç.Dr. İ. Çayıroğlu 0425
10	08:00 - 08:45					

Figure 11. Weekly Scheduling for Instructor

Course schedule prepared for the instructor is seen on Figure 11. In each time slot, course code, classroom code and academic program code are indicated.

BİLGİSAYAR SİSTEMLERİ ÖĞRETMENLİĞİ - 4. Sınıf						
GÜN / SAAT		PAZARTESİ	SALI	ÇARŞAMBA	PERŞEMBE	CUMA
1	08:00 - 08:45					
2	08:00 - 08:45		BLG407 (T) - ANFİ 12 Doç.Dr. R. Bayır (2)		BLG405 (T) - ANFİ 11 Yrd.Doç.Dr. B. Şen (3)	EGT401 (T) - ANFİ 12 Prof.Dr. A. Özçifçi (1)
3	08:00 - 08:45		BLG407 (T) - ANFİ 12 Doç.Dr. R. Bayır (2)		BLG405 (T) - ANFİ 11 Yrd.Doç.Dr. B. Şen (3)	
4	08:00 - 08:45	EGT403 (T) - OTO. 3 Yrd.Doç.Dr. A. Özyürek (2)			BLG405 (T) - ANFİ 11 Yrd.Doç.Dr. B. Şen (3)	
5	08:00 - 08:45	EGT403 (T) - OTO. 3 Yrd.Doç.Dr. A. Özyürek (2)				
6	08:00 - 08:45	BLG411 (T) - 101 Öğr.Gör. O. Erkaymaz (3)		BLG403 (T) - ANFİ 10 Yrd.Doç.Dr. İ. Çayıroğlu (3)		
7	08:00 - 08:45	BLG411 (T) - 101 Öğr.Gör. O. Erkaymaz (3)		BLG403 (T) - ANFİ 10 Yrd.Doç.Dr. İ. Çayıroğlu (3)	BLG401 (T) - ANFİ 9 Yrd.Doç.Dr. S. Görgünoğlu (3)	
8	08:00 - 08:45	BLG411 (T) - 101 Öğr.Gör. O. Erkaymaz (3)	EGT405 (T) - TES. 2 Prof.Dr. E. Candan (2)	BLG403 (T) - ANFİ 10 Yrd.Doç.Dr. İ. Çayıroğlu (3)	BLG401 (T) - ANFİ 9 Yrd.Doç.Dr. S. Görgünoğlu (3)	
9	08:00 - 08:45		EGT405 (T) - TES. 2 Prof.Dr. E. Candan (2)		BLG401 (T) - ANFİ 9 Yrd.Doç.Dr. S. Görgünoğlu (3)	
10	08:00 - 08:45					

Figure 12. Weekly Scheduling for Student

Similarly, for academic program and classrooms, it is as indicated on Figure 12 and Figure 13.

203

GÜN / SAAT		PAZARTESİ	SALI	ÇARŞAMBA	PERŞEMBE	CUMA
1	08:00 - 08:45					
2	08:00 - 08:45	MKM201 (T) [4] - 0210 Doç.Dr. O. Yiğit		BLM181 (T) [2] - 0201 Yrd.Doç.Dr. M. Özalp	MKM105 (T) [2] - 0201 Yrd.Doç.Dr. G. Sur	BLM181 (T) [2] - 0210 Yrd.Doç.Dr. M. Özalp
3	08:00 - 08:45	MKM201 (T) [4] - 0210 Doç.Dr. O. Yiğit	MAT181 (T) [3] - 0210 Yrd.Doç.Dr. M. Düz	BLM181 (T) [2] - 0201 Yrd.Doç.Dr. M. Özalp	MKM105 (T) [2] - 0201 Yrd.Doç.Dr. G. Sur	BLM181 (T) [2] - 0210 Yrd.Doç.Dr. M. Özalp
4	08:00 - 08:45	MKM201 (T) [4] - 0210 Doç.Dr. O. Yiğit	MAT181 (T) [3] - 0210 Yrd.Doç.Dr. M. Düz	TUR181 (T) [2] - 0201 Yrd.Doç.Dr. A. Öksüz	KIM181 (T) [2] - 0208 Yrd.Doç.Dr. M. Levent	
5	08:00 - 08:45	MKM201 (T) [4] - 0210 Doç.Dr. O. Yiğit	MAT181 (T) [3] - 0210 Yrd.Doç.Dr. M. Düz	TUR181 (T) [2] - 0201 Yrd.Doç.Dr. A. Öksüz	KIM181 (T) [2] - 0208 Yrd.Doç.Dr. M. Levent	
6	08:00 - 08:45					
7	08:00 - 08:45	GSM181 (T) [2] - 0201 Öğr.Gör. G. Hacısüleymanoğlu	FIZ181 (T) [3] - 0205 Prof.Dr. D. Paktaş	FIZ181 (T) [3] - 0208 Prof.Dr. D. Paktaş	FIZ181 (T) [3] - 0210 Prof.Dr. D. Paktaş	
8	08:00 - 08:45	GSM181 (T) [2] - 0201 Öğr.Gör. G. Hacısüleymanoğlu	FIZ181 (T) [3] - 0205 Prof.Dr. D. Paktaş	FIZ181 (T) [3] - 0208 Prof.Dr. D. Paktaş	FIZ181 (T) [3] - 0210 Prof.Dr. D. Paktaş	GSR181 (T) [2] - 0208 Yrd.Doç.Dr. A. Ertok atmaca
9	08:00 - 08:45		FIZ181 (T) [3] - 0205 Prof.Dr. D. Paktaş	FIZ181 (T) [3] - 0208 Prof.Dr. D. Paktaş	FIZ181 (T) [3] - 0210 Prof.Dr. D. Paktaş	GSR181 (T) [2] - 0208 Yrd.Doç.Dr. A. Ertok atmaca
10	08:00 - 08:45					

Figure 13. Weekly Scheduling for Classroom

3. Result and Discussion

Prepared scheduling program has been tried on a computer with Microsoft Windows Server 2003 installed, 1024 MB memory and Core2 Quad 2.40 GHz processor. The experiments have been conducted for selecting the most suitable algorithm operators and parameter values. The data used in the experiments are actual data and comprises 504 instructors, 4163 classrooms and 10525 students for the entire University. Thus, the obtained results have indicated the actual performances of operators and parameter values. The course scheduling solved using genetic algorithm belongs to 2009-2010 fall term of Karabük University, Faculty of Technical Education. Scheduling is composed of courses which are given 5 days a week, 10 lessons a day and each being 40 minutes.

While schedule of the faculty is prepared, obligatory and flexible constraints have been formed considering the requirements of academic programs. The data regarding the size of experiment data are indicated in Table 1.

Tournament selection operator has been used in the experiments. Crossing rate has been kept fixed as 85%. A mutation method complying with the structure of the problem was developed and experiments were conducted accepting mutation probability as 1%. In addition, if fitness rate of each individual in the population along 20 iterations doesn't change, mutation rate is increased by 1% gradually.

Table 1 Size of Experiment Data.

Number of Instructors	64
Number of Classrooms	48
Number of Classes	32
Number of Courses	253
Number of Course Hours	705
Number of Academic Programs	8

Iteration number was stated as 1.000 in the experiments. If there is no conflict while finding a possible solution in genetic algorithm, the fitness value of the best solution is positive integer valence point. Shortly, if a fitness value equal to or over 0 is reached during operation of the algorithm, conflict rate is 0%.
Searching process is ended depending on iteration number, conflict success rate or user demand. If demanded results are not obtained despite exceeding 1.000 iteration, the operation of the algorithm is interrupted. The solution which after 1.000 iteration still has conflicts and cannot fulfill some constraints is saved in database

with all error codes. Thus, if there is a situation to allow flexible constraints in the results, intended solution might be used as course scheduling of that term by educational institution.

Experiment No: 1

In this experiment, population number has been selected as 25, crossing rate as 85% and mutation probability as 1%. Periodic crossing operator has been used in the experiment. Private mutation method which is suitable for the model developed for mutation operator has been used.

Weight of lesson load in the schedule has been selected as balanced. Lunch break course splitting has not been allowed. Low punishment has been applied in minimizing the idle time between the courses, and pedagogical day and hour productivity has been added to the constraints.

A solution was formed in this experiment for both satisfaction of instructors and students considering pedagogical day and hour productivity. As seen on Figure 14.a, conflict rate was reduced to 0% at iteration 445. According to the results obtained, satisfaction of instructors was calculated as 32%, pedagogical day as 80% and pedagogical hour productivity as 57%.

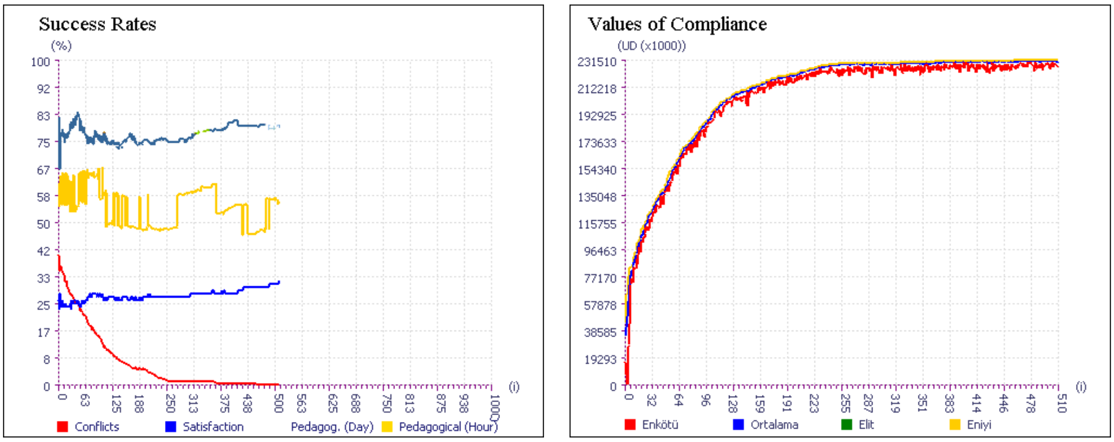

Figure 14. a) Success Rates Graphic, b) Compliance Values

Fitness values (UD) obtained as a result of experiment performed are indicated on Figure 14.b. Also as stated in the figure, the best UD used in the solution of the problem is indicated with yellow, the best solution in each iteration with green, the average value of chromosomes in each iteration with blue and the chromosome values with the lowest UD in each iteration with red.

The UD values indicated in fitness values graphics in the experiments were rescaled between 0 and UD value (UDMAX) of the best solution and negative values were not used. Conflict punishments used in the calculations were determined as negative point.

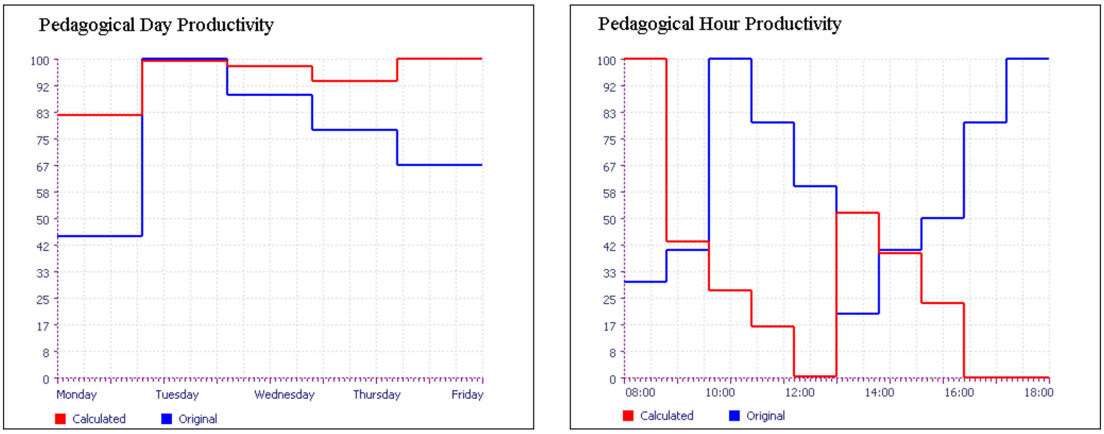

Figure 15. Pedagogical Day and Hour Productivity

In pedagogical day and hour productivity indicated on Figure 15, the values that must be are indicated with blue, the day and hour values calculated for the best solution found as a result of genetic algorithm processes are indicated with red. While the productivity has shown 80% similarity as a result of the experiment conducted, hour productivity was found as 57%.

In this experiment, no constraint was given priority. It might be expected that satisfaction of instructor and pedagogical day-hour productivity yield better results by prolonging iteration.

Experiment No: 2

In this experiment, population number has been selected as 20, crossing rate as 85% and mutation probability as 1%. Periodic crossing operator has been used in the experiment. Private mutation method which is suitable for the model developed for mutation operator has been used.

Weight of lesson load in the schedule has been selected as balanced. Lunch break course splitting has not been allowed. Low punishment has been applied in minimizing the idle time between the courses and pedagogical day and hour productivity were inactivated.

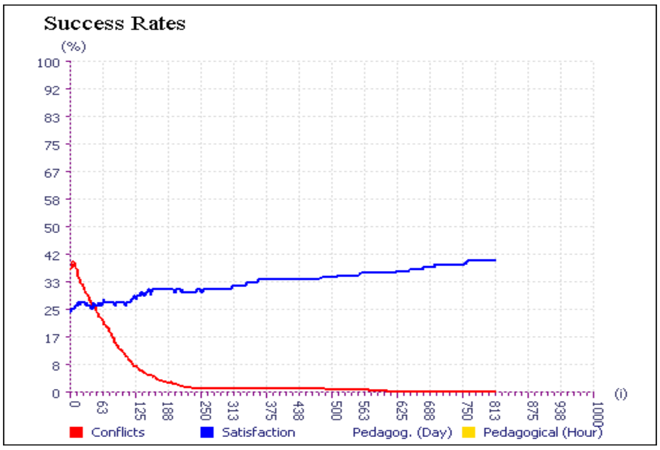

Figure 16. Success Rates Graphic

A solution was formed in this experiment considering satisfaction of instructors. As seen on Figure 14.a, conflict rate was reduced to 0% at iteration 607. According to the results obtained, satisfaction of instructors was calculated as 41%.

Experiment No: 3

In this experiment, population number has been selected as 10, crossing rate as 85% and mutation probability as 1%. Periodic crossing operator has been used in the experiment. Private mutation method which is suitable for the model developed for mutation operator has been used.

Weight of lesson load in the schedule has been selected as balanced. Lunch break course splitting has not been allowed. Low punishment has been applied in minimizing the idle time between the courses and pedagogical day and hour productivity were inactivated.

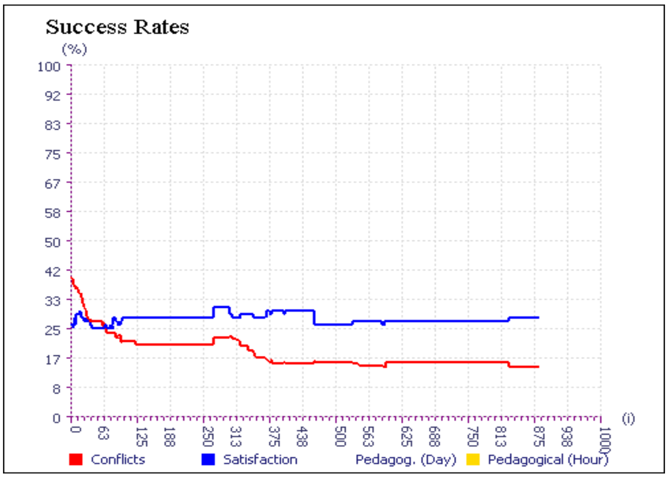

Figure 17. Success Rates Graphic

A solution was formed in this experiment considering satisfaction of instructors. As seen from Figure 17, conflict rate is still 14% at iteration 880. According to the results obtained, satisfaction of instructors was calculated as 28%.

4. Conclusion

With the genetic algorithm developed for solving the university scheduling problem, schedules of faculty, high school, institute and vocational high schools were formed in a short term with 0% conflict rate. In addition, optimum solutions were obtained considering satisfaction of instructors and pedagogical day-hour productivity.

Karabük University, Faculty of Technical Training with maximum number of students was used as example in the experiments carried out. 253 courses (totally 705 hours) opened in 2009-2010 fall term were given by 64 different instructors in 48 different classrooms. Forming time of schedules was calculated as 15 minutes on average. As another example, 56 courses (totally 171 hours) opened in 2009-2010 fall term at Karabuk University, Faculty of Science and Letters were given by 33 different instructors in 5 different classrooms. Average forming time of schedules vary between 1-2 minutes.

When optimum genetic algorithm parameters and operators are examined, the following are found;
- The most appropriate selection method in the solution of the problem is Tournament,
- Periodic crossing method and 85% crossing rate,
- In mutation method, a method similar to intermutation method was developed by adding some special attachments to conventional structures and mutation rate was found as 1%.
- Consequently, the most appropriate value was determined between 25-25 for Faculty of Technical Training population number of which was used in the experiments.

References

1. Daskalaki, S., Birbas, T., Housos E., "An Integer Programming Formulation for A Case Study in University Timetabling", European Journal of Operational Research, 153: 117-135 (2004).
2. Bağış, A., "Genetik Algoritma Kullanılarak Ders Programının Optimum Şekilde Düzenlenmesi", Yüksek Lisans Tezi, Erciyes Üniversitesi Fen Bilimler Enstitüsü, Kayseri, 1996.
3. Wadhwa S., Chopra A., "A Genetic Algorithm Application: Dynamic Re-configuration in Agile Manufacturing Systems", Studies in Informatics and Control, 9: 1-9 (2000).
4. Yeniay, Ö., "An Overview of Genetic Algorithms", Anadolu Üniversitesi Bilim ve Teknoloji Dergisi, 2(1): 37-49 (2001).
5. Al-Milli, N. R., "Hybrid Genetic Algorithms with Great Deluge For Course Timetabling", IJCSNS International Journal of Computer Science and Network Security, 10(4): 283-288 (2010).
6. Wijaya, T., Manurung, R., "Solving University Timetabling As a Constraint Satisfaction Problem with Genetic Algorithm", In Proceedings of the International Conference on Advanced Computer Science and Information Systems (ICACSIS 2009), Depok, 2009.
7. Jat, S. N., Yang, S., "A Guided Search Genetic Algorithm for The University Course Timetabling Problem", Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2009), Ireland, 180-191 (2009).
8. Atanak, M. M., Hocaoğlu, F. O., "Genetik Algoritmalarla Ders Programı Hazırlama Otomasyonu Tasarımı", III.Otomasyon Sempozyumu ve Sergisi, Denizli, 2005.
9. Bratković, Z., Herman, T., Omrčen, V., Čupić, M., Jakobović, D., "University Course Timetabling with Genetic Algorithm: a Laboratory Exercises Case Study", Lecture Notes in Computer Science, 5482/2009: 240-251 (2009).
10. Aldasht, M., Alsaheb, M., Adi, S., Qopita, M. A., "University Course Scheduling Using Evolutionary Algorithms", 4th International Multi-Conference on Computing in the Global Information Technology, Cannes/La Bocca, France, 47-51 (2009).
11. Aydın, M. A., "Solving University Course Timetabling Problem Using Genetic Algorithm", Master Thesis, Bahçeşehir University Institute of Sciences Industrial Engineering, İstanbul, 2008.
12. Kalender, M., "Ders Çizelgeleme Programı", TMMOB Elektrik Mühendisleri Odası İstanbul Şubesi 2006-2007 Öğretim Yılı Proje Yarışması, 2007.
13. Juang, Y., Lin, S., Kao, H., "An Adaptive Scheduling System with Genetic Algorithms for Arranging Employee Training Programs", Expert Systems with Applications 33: 642–651 (2007).
14. Baç, U., "Akademik Ders Programlarının Yapılması Probleminin Matematiksel Modeller ve

Algoritmalarla Çözümü ve Uygulaması”, Yüksek Lisans Tezi, Hacettepe Üniversitesi Sosyal Bilimler Enstitüsü, Ankara, 2007.

15. Temiz, İ., Erol, S., “Bulanık Akış Tipi Çizelgeleme Problemi için Çok Amaçlı Genetik Algoritma”, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 22(4): 855-862 (2007).
16. Gülcü A., “Yapay Zeka Tekniklerinden Genetik Algoritma ve Tabu Arama Yöntemlerinin Eğitim Kurumlarının Haftalık Ders Programlarının Hazırlanmasında Kullanımı”, Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 2006.
17. Datta, D., Deb, K., Fonseca, C. M., “Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm”, KanGAL, Report No. 2006006, India, 1-10 (2006).
18. Çayıroğlu İ., Dizdar, E. N., “Uzman Sistem Destekli Online Ders Yerleştirme Programı”, Teknoloji Dergisi, 9(4): 283-293 (2006).
19. Yiğit, T., “Meslek Liseleri Haftalık Ders Çizelgelerinin Genetik Algoritmalar Yardımıyla Oluşturulması”, Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi Dergisi, 19: 25-39 (2006).
20. Dammak, A., Elloumi, A., Kamoun H., “Classroom Assignment for Exam Timetabling”, Advances in Engineering Software, 37: 659-666 (2006).
21. Kazarlis, S., Petridis, V., Fragkou, P., “Solving University Timetabling Problems Using Advanced Genetic Algorithms”, 5th International Conference on Technology and Automation, Thessaloniki/Greece, 131-136 (2005).
22. Biroğul, S., Güvenç, U., “Genetik Algoritma ile Çözümü Gerçekleştirilen Atölye Çizelgeleme Probleminde Ürün Sayısının Etkisi”, Akademik Bilişim 2007, Kütahya, 2007.
23. Head, C., Shaban, S., “A Heuristic Approach to Simultaneous Course/Student Timetabling”, Computers & Operations Research, 34: 919–933 (2007).
24. Daban, F., Özdemir, E., “Çok Parametreli Genetik Algoritma Kullanarak Ders Programlarının Hazırlanması”, II. Elektrik Elektronik Bilgisayar Mühendislikleri Eğitimi Sempozyumu, 2004.
25. Binbaşıoğlu, C., “Öğretim Metodu ve Uygulamaları”, Binbaşıoğlu Yayınevi, Ankara, (1973).