



1997

T.C. MALTEPE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**SINAV ÇİZELGELEME PROBLEMİNİN OPTİMİZASYONU**

**Ülkü Burcu GÜREL**

Yüksek Lisans Tezi

**Tez Danışmanı**

**Doç. Dr. Oruç Raif ÖNVURAL**

**İSTANBUL – 2012**



**T.C.**  
**MALTEPE ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**  
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**SINAV ÇİZELGELEME PROBLEMİNİN OPTİMİZASYONU**

**YÜKSEK LİSANS TEZİ**

**Ülkü Burcu GÜREL**

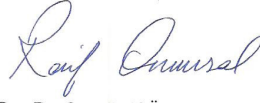
**Tez Danışmanı**  
**Doç. Dr. Oruç Raif ÖNVURAL**

**İSTANBUL – 2012**



Bu tez çalışması, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 21/09/2012 tarih ve 2012/18 sayılı kararıyla oluşturulan jüri tarafından **Bilgisayar Mühendisliği Tezli Yüksek Lisansı Tezi** olarak kabul edilmiştir.

JÜRİ



Doç. Dr. Oruç Raif ÖNVURAL

Üye

(Danışman)



Doç. Dr. Nihat KABAOĞLU

Üye



Yrd. Doç. Dr. Ali AKMAN

Üye

## ÖZET

**Yüksek Lisans Tezi**

**Ülkü Burcu GÜREL**

**SINAV ÇİZELGELEME PROBLEMİNİN OPTİMİZASYONU**

**T.C. Maltepe Üniversitesi, Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı: Doç. Dr. Oruç Raif ÖNVURAL**

Bu çalışmanın amacı, üniversitelerin sınav çizelgeleme problemine bir çözüm üretmektir.

Sınav çizelgeleme problemleri NP (Non-polinomal, polinomal olmayan) zor diye adlandırılan optimizasyon problemleridir. Bu sınıftaki problemlerin her zaman tam çözümü bulunamayabilir. Bunun için, yaklaşık sonuçlar üretecek çözümler kullanılır. Bu çalışmada, yaklaşık çözüm üreten üç farklı algoritma geliştirilmiş ve ürettikleri sonuçlar karşılaştırılmıştır. Ayrıca geliştirilen algoritmalar için görsel arayüzü geliştirilmiştir. Bu arayüz ile geliştirilen algoritmaların ürettiği sonuçlar elle kolayca değiştirilebilir.

**Anahtar Kelimeler:** Optimizasyon, sınav çizelgeleme, sınav çakışmalarının optimizasyonu, atama problemi.

2012, 60 sayfa.

## **ABSTRACT**

**Master's Thesis**

**Ülkü Bureu GÜREL**

**OPTIMIZATION OF EXAM SCHEDULING PROBLEM**

**T. C. Maltepe University, Institute of Science and Technology**

**Department of Computer Engineering**

**Dissertation Advisor : Assoc. Dr. Oruç Raif ÖNVURAL**

The purpose of this study is to develop heuristics to help scheduling exams in universities. Three different heuristics are developed and their results are compared. In addition, we developed a graphical user interface with features that allows manual intervention to the solutions produced with these heuristics.

The exam scheduling problems are known to be Non-Polynomial (NP) complete optimization problems. Accordingly, the best solution may not be found in polynomial time. As a result, heuristics are developed to provide good solutions to these problems..

**Key words:** Optimization, test/exam scheduling, optimization of exam coincidence, assignment problem.

2012, 60 pages.

## TEŐEKKÜR

Çalıőmalarımnda, ne kadar yoğun olursa olsun her zaman desteęini, sabrını ve yol gösterici ışığıını esirgemeyen deęerli hocam ve danıőmanım, T.C. Maltepe Üniversitesi Bilgisayar Mühendislięi Anabilim Dalı Bölüm Başkanı Sayın Doç. Dr. Oruç Raif Önvural'a, tecrübelerini ve derin bilgilerini hiçbir zaman paylaşmaktan sakınmayan deęerli hocam, T.C. Maltepe Üniversitesi Fen Bilimleri Enstitüsü Müdürü Sayın Prof. Dr. İlhami Yavuz'a, desteęini, sevgisini, emeęini esirgememiő, Daęıstanlıoęlu, Yüksel ve Gürel ailelerinin, aramızda olan ve olmayan tüm bireylerine teőekkürlerimi sunuyorum.

Ülkü Burcu Gürel

İstanbul - 2012



## İÇİNDEKİLER

ÖZET .....	i
ABSTRACT .....	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
KISALTMALAR DİZİNİ.....	vii
ŞEKİLLER LİSTESİ.....	viii
GİRİŞ .....	1
<b>1 ÇİZELGELEME PROBLEMLERİ VE OPTİMİZASYON .....</b>	<b>3</b>
<b>1.1 Çizelgeleme Problemleri .....</b>	<b>3</b>
1.1.1 Çizelgelemenin temel amaçları .....	6
1.1.2 Çizelgelemenin kullanıldığı problemler .....	6
<b>1.2 Sınav Çizelgeleme Problemi ve Problem Çözümündeki amaçlar .....</b>	<b>7</b>
1.2.1 Sınav Çizelgelemenin temel amaçları .....	7
<b>1.3 Sınav Çizelgeleme Probleminin Optimizasyon İşlemi .....</b>	<b>8</b>
<b>2 LİTERATÜRDE KAYNAK ARAŞTIRMASI VE KULLANILAN ALGORİTMA ÖZETLERİ.....</b>	<b>11</b>
<b>2.1 Graf Tabanlı Teknikler.....</b>	<b>12</b>
<b>2.2 Kısıt Tabanlı Teknikler.....</b>	<b>13</b>
<b>2.3 Yerel Arama Teknikleri.....</b>	<b>13</b>
<b>2.4 Tabu Arama .....</b>	<b>13</b>
<b>2.5 Tavlama Benzetimi.....</b>	<b>14</b>
<b>2.6 Genetik Algoritmalar .....</b>	<b>15</b>
<b>2.7 Karınca Algoritmaları .....</b>	<b>15</b>
<b>2.8 Üst Sezgisel Teknikler .....</b>	<b>15</b>

<b>2.9</b>	<b>Yazılım piyasasındaki sınav programı yapan benzer yazılımlar .....</b>	<b>18</b>
<b>3</b>	<b>LİTERATÜRDE KULLANILAN ÇÖZÜM YÖNTEMLERİ.....</b>	<b>21</b>
<b>3.1</b>	<b>Genetik arama algoritması .....</b>	<b>21</b>
3.1.1	Genetik algoritmaların tarihçesi.....	21
3.1.2	Genetik algoritmanın tanımı ve çalışma prensibi.....	23
3.1.3	Gen .....	23
3.1.4	Kromozom .....	24
<b>3.2</b>	<b>Genetik algoritmaların akış diyagramı .....</b>	<b>25</b>
3.2.1	Akış diyagramının açıklaması.....	25
<b>3.3</b>	<b>Çaprazlama yöntemi .....</b>	<b>26</b>
<b>3.4</b>	<b>Popülasyon büyüklüğü.....</b>	<b>28</b>
<b>3.5</b>	<b>GA'nın uygulama alanları .....</b>	<b>29</b>
<b>4</b>	<b>GELİŞTİRDİĞİMİZ ALGORİTMALAR VE YAZILIM .....</b>	<b>30</b>
<b>4.1</b>	<b>RSINA Rastgele Sınav Atama Algoritması.....</b>	<b>30</b>
<b>4.2</b>	<b>SINA1 Sınav Atama Algoritması.....</b>	<b>30</b>
<b>4.3</b>	<b>SINA2 Sınav Atama Algoritması.....</b>	<b>30</b>
<b>4.4</b>	<b>SINA1 ve SINA2 AKIŞ DİYAGRAMI.....</b>	<b>31</b>
<b>4.5</b>	<b>KISITLAR.....</b>	<b>32</b>
<b>4.6</b>	<b>SİSTEM MİMARİSİ VE GELİŞTİRİLMESİ.....</b>	<b>32</b>
4.6.1	Sistem tasarımı .....	32
4.6.1.1	Veri Girişi.....	33
4.6.2	SINA1 ve SINA2 algoritmaları için Çakışma Matrisi Bulma .....	34
4.6.3	Kullanıcı Arayüz ekranları.....	36
4.6.3.1	Öğrenci Bilgileri .....	36
4.6.3.2	Sınav Tarihleri Belirleme ve Listeleme .....	38
4.6.3.3	Sınav Rezervasyon.....	39
4.6.3.4	Sınav Atama .....	41
4.6.3.5	Ders Atama fonksiyonu, SINA1 ve SINA2 algoritmalarının çalışması 45	
4.6.4	System - Ayarlar.....	51
4.6.5	Raporla / Yazdır .....	52
<b>5</b>	<b>TARTIŞMA VE SONUÇ .....</b>	<b>53</b>

<b>KAYNAKLAR.....</b>	<b>56</b>
<b>ÖZGEÇMİŞ.....</b>	<b>60</b>

## **KISALTMALAR DİZİNİ**

GA : Genetik Algoritma

RSINA : Rastgele sınav atama algoritması

SA : Simulated Annealing Algoritması

SINA1 : Büyükten küçüğe çakışma sıralı optimizasyonu algoritması

SINA2 : Küçükten büyüğe çakışma sıralı optimizasyonu algoritması

TA : Tabu Arama Algoritması

VLSI : Very Large Scale Integration

## ŞEKİLLER LİSTESİ

Şekil 1.1 Çizelgeleme problem çözümüne örnek yöntemler .....	6
Şekil 3.1 Sınav programı çizelgeleme gen yapısı.....	24
Şekil 3.2 Gen dizisi.....	24
Şekil 3.3 Genetik algoritmalar için genel akış diyagramı .....	25
Şekil 4.1 Algoritmalarımız için genel akış diyagramı .....	31
Şekil 4.2 Dersler tablosu .....	34
Şekil 4.3 Derslerin birbirleriyle çakışma matrisi .....	35
Şekil 4.4 Öğrenci görüntüleme, ekleme, güncelleme ekranı .....	36
Şekil 4.5 Database _OGRENCILER tablosu .....	37
Şekil 4.6 Öğrenciler listesi .....	37
Şekil 4.7 Sınav saati giriş ve güncelleme ekranı .....	38
Şekil 4.8 Database _SINAV_TARIHLERI tablosu.....	39
Şekil 4.9 Sınav saatleri listesi.....	39
Şekil 4.10 Sınav oturumu rezervasyon ekranı.....	40
Şekil 4.11 Database SINAV_REZERVE tablosu.....	40
Şekil 4.12 Sınav rezervasyonu listesi .....	40
Şekil 4.13 Sınav atama ekranı 1 .....	42
Şekil 4.14 Sınav atama ekranı 2 .....	42
Şekil 4.15 Sınav atama ekranı 3 .....	43

Şekil 4.16 Sınav takvimi öğrenci sorgulama.....	44
Şekil 4.17 Derslerin sınav takviminde ders_id ve öğrenci detayları .....	44
Şekil 4.18 Derslerin sınav takviminde ders_id ve öğrenci detayları .....	45
Şekil 4.19 Derslerin çakışma matrisi .....	45
Şekil 4.20 Derslerin çakışma matrisi devamı.....	46
Şekil 4.21 SINA1 algoritması sınav atama matrisi.....	46
Şekil 4.22 SINA1 ile yeni sınav takvimi.....	47
Şekil 4.23 2012-Raif manuel düzenlenmiş takvim.....	47
Şekil 4.24 SINAV_ATAMA tablosu son hali.....	48
Şekil 4.25 SINA2 algoritmasından sonraki sınav çizelgesi .....	48
Şekil 4.26 2011-Exam2 sınav tablosu.....	49
Şekil 4.27 2011-Exam2 için çakışma matrisi.....	49
Şekil 4.28 2011-Exam2 için SINA2 algoritması ile sınav çizelgelemesi .....	50
Şekil 4.29 Sistem ayarları ekranı.....	51
Şekil 4.30 Örnek sınav saatleri detay listesi.....	52
Şekil 4.31 Örnek öğrenci bilgileri listesi .....	52

## GİRİŞ

Üniversitelerdeki ön lisans ve lisans öğrencilerinin vize, final vb. sınavları belirli zaman diliminde yapılmak zorundadır. Binlerce öğrenciye ait dersler, yüzlerce oturumluk sınavlar oluşturmaktadır. Sınavların, salonlara yerleştirilmesi, kimlerin hangi oturumlarda görev alacağı, bazı derslerin kendine has özellikleri (örneğin teknik resim dersinin mutlaka teknik resim salonlarında yapılması gibi), gözetmenlerin ve ders yürütücülerinin özel istekleri, salonların durumu, sınav sürelerinin farklı olması, gibi onlarca farklı durum göz önüne alınarak sınavlar salonlara yerleştirilmeli ve salonlarda gözetmenlik görevi alacak öğretim elemanlarına adaletli bir şekilde dağıtılmalıdır. Üniversitelerin, özellikle meslek yüksekokullarında, yüksekokullarda ve yeterli araştırma görevlisi olmayan fakültelerde sınav yürütücülüğünün yanı sıra gözetmenlik görevlerini öğretim elemanları yapmak zorundadırlar. Ancak gözetmenlik görevleri, öğretim elemanlarının ünvanlarına bağlı olarak dağıtılmalıdır.

Sınav yerleşim işlemi halen pek çok fakülte, yüksekokul ve meslek yüksekokulunda excel gibi elektronik tablo programı yardımıyla el ile birkaç kişiden oluşan bir ekip tarafından günlerce mesai harcanarak yapılmaya çalışılmaktadır. Bu da, sınıf çakışması, gözetmen çakışması, öğrencilerin verilen salona sığmaması vb. gibi sorunları da beraberinde getirmektedir. Yine öğretim elemanlarının birtakım istekleri olmaktadır. Örneğin kendi dersinin sınavı olduğu ve bir önceki veya bir sonraki oturumda gözetmenlik istememektedirler. Ayrıca öğretim elemanları mümkün olduğu kadar kendi bölüm öğrencilerinin derslerinde gözetmenlik yapmak istemektedirler. Bazı sınavların süreleri ortalamanın çok üzerinde olabilmektedir. Örneğin teknik resim dersinin sınavı 200 dakika gibi. Bu uzun süreli sınavların da, dengeli dağılması arzu edilmektedir. Bazı öğretim elemanları gözetmenlikten muaf tutulması gerekebilmektedir. Özet olarak sınav atamalarında pek çok parametre ve kısıt vardır.

Bu çalışmada, bu kısıtların bazılarını göz önüne alarak üç değişik algoritma geliştirildi. Bu algoritmalar Greedy tekniği kullanılarak geliştirilen algoritmalar

sınıfına girebilir. Geliştirilen algoritmalar nesnel bir ara yüzü ile kullanılır ve günlük sınav çizelgelerini üretir.

Geliştirilen algoritmalar C#.net programlama dili ve MS SQL veri tabanı kullanılarak geliştirilmiştir. Algoritmanın performansı, Maltepe Üniversitesi veri tabanında bulunan öğrenci ve ders tabloları kullanılarak yapılmıştır. Bu yazılımı kullanarak sınav çizelgeleme işlemi çabukça gerçekleştirilmektedir.

Hemen hemen her iş kolunda ( eğitim, ulaştırma, üretim, hizmet, makina/insan işgücü, finans vb.. ) planlama yaparken, 'Zaman Çizelgeleme Problemi' ile karşılaşmaktayız. Farklı yaklaşımlarla çözülmeye çalışılan zor bir problem türüdür. Literatürde önerilmiş yöntemler, Graf tabanlı yaklaşımlar, yerel arama teknikleri, kısıt programlama yaklaşımları, popülasyon tabanlı yaklaşımlar, melez yaklaşımlar ve üst sezgisel yaklaşımlar olarak özetlenebilir. Son yıllarda, sezgisel algoritmalar kullanılarak çözüm yöntemleri oldukça geliştirilmiştir.

Sezgisel algoritmaların kullanılma sebebi, farklı hareketlerden en iyi olanına karar verebilmeyi sağlayabilmesidir. Yani, kesin çözümü bilinmeyen ama en yakın çözümde nelerin olmasını hedeflediğimizi söyleyebildiğimiz bir problemde, çözümlerin kalitelerini ölçmek için de iyi olana karar vermek amacıyla tanımlanan yöntemlerdir. Bu yaklaşımlardan bazılarında, Genetik Algoritma, Tabu Arama, Karınca Kolonisi Algoritması, Tavlama Benzetimi teknikleri kullanılarak değişik matematiksel modeller ve sonuçlar elde edilmiştir. Bu modeller, uygulanan metodlar birbirlerinden çok farklıdır. Bu nedenle de literatüre girmiş olan bu sonuçları, birbirleriyle karşılaştırmak bazen çok zor olabilmektedir.

Bilgisayar yazılımlarına uygulanan bu yapay zeka yöntemleri aslında, elimizde var olan problemin çözümünü spesifik olarak sağlayabilmektedir. Fakat literatürdeki diğer benzer problemlerin çözümü için birebir çözümdür diyemeyiz. Bu nedenle, son yıllarda birçok araştırmacı çözüm yaklaşımlarını üst sezgisel algoritmalara (hyper heuristics algorithm) yoğunlaştırmıştır.



# 1 ÇİZELGELEME PROBLEMLERİ VE OPTİMİZASYON

## 1.1 Çizelgeleme Problemleri

Çizelgeleme problemleri, 20. yy başlarından bu yana literatürde çok yeri olan bir konudur. İlk olarak Henry Gantt ve diğerleri tarafından üzerinde çalışmalar başlamıştır. Bu çalışmalar, ilk önceleri, problemleri bir araya getirerek ortak bir problem yapısını anlamaya yönelik olmuştur. Basit modellere çözümler üretilmeye çalışılmıştır. 20. yy'ın sonlarına doğru ise sezgisel yöntemler ön plana çıkmıştır. Bu yöntemler probleme özel sezgisellerdi. 1990'lardan bu yana genel amaçlı modern sezgisel yöntemler kullanılmaya başlanmıştır.

Bu çalışmalarda, tamamen çizelgeleme problemlerinin NP zor sınıfına girmesinden kaynaklı olarak sezgisel yöntemlere başvurulmuştur. Verimliliğin hedeflendiği, çakışmaların ise mümkün olduğunca en aza indirgenmesi amaçlanmıştır.

“Çizelgeleme, kıt kaynakların belirli bir zaman boyunca işlere tahsis edilmesiyle ilgilidir. Bu süreç bir veya daha fazla hedefin optimizasyonunu amaçlayan bir karar alma sürecidir (Pinedo, 2002'ye atfen Seçme, 2006, s.4.)" - [37]

Aslında bir zaman çizelgeleme problemi olan sınav atamalarında da amaç, kısıtlar gözönüne alınarak, en iyi performansla, en az çakışma ile bir çizelgeleme yapabilmektir. Bu problemin çözümü hiçbir zaman kesin budur diyemeyiz. Çizelgeleme problemi çok zor bir problemdir. Bu tür problemlerinin çözümü için değişik yaklaşımlar kullanılmıştır. Schmidt, bildirisinde bu konudaki ilk çalışmalardan bahseder [43] . Zaman çizelgeleme problemleri için geliştirilen modeller genelde birbirinden oldukça farklıdır. Bu nedenle de son yıllarda, çizelgeleme problem çözümlerinde, Tavlama Benzetimi (Simulated Annealing) [sayfa 14], Tabu Arama (Tabu Search) [sayfa 13] ve Genetik Algoritmalar (Genetic Algorithms) [sayfa 15] gibi yapay zekâ yöntemlerine dayandırılarak geliştirilen programlar yazılmaktadır.

Kısıtların belirlenebilmesi için de kullanıcı dostu ara yüz ekranları tasarlanmaktadır.

Bu tezde ele alacağımız sınav çizelgeleme problemi de bir çizelgeleme problemidir. Amacımız, bu problemdeki çakışmaları en aza indirerek, sonucumuzu optimize etmektir.

Örneğin, Maltepe Üniversitesinde sınav programının belirlenmesi için bütün sınav tarihlerinin ve saatlerinin belirleniyor olması gerekmektedir.

“Etkin bir çizelgeleme sayesinde belirli faaliyetlerin daha az kaynak kullanarak daha kısa zamanda yapılabilme olanağı ortaya çıkmaktadır (Güldalı, 1990’a atfen Seçme, 2006, s.4)” - [22].

“Çizelgeleme problemleri, yerine getirilmesi gereken bir grup görev ve bu görevlerin gerçekleştirilmesinde kullanılacak uygun kaynakları içerirler. Bu iki temel girdinin nitelikleri iyi belirlenmeli ve alacakları değerler mümkün olduğunca kesin ve doğru olarak hesaplanmalıdır. Ancak bu şekilde doğru zaman planları ortaya çıkarılabilir. Kaynaklar belirlendiğinde, çizelgeleme probleminin sınırları etkin bir şekilde çizilmiş olmaktadır. Bununla beraber her bir görev; kaynak gereksinimi, süresi, başlama ve bitiş zamanları gibi bilgiler cinsinden açıklanır. Aynı zamanda, bu görevler bütünü arasında her hangi bir teknolojik kısıt varsa belirtilir (Özkazanç, 1999’a atfen Seçme, 2006, s.5)” - [36]

“Yapılacak çizelgelemede aşağıdaki gibi bir yol takip edilmelidir (Biroğul, 2005, s.5)” - [6] :

1. Problemin anlaşılması,
2. Problemin tanımlanması,
3. Problemin daha önceden çözülmüş teorik ve uygulamalı problemlerle olan benzerliklerinin ve farklılıklarının belirlenmesi,
4. Problemin zorluk derecesinin belirlenmesi,

5. Zorluk derecesine göre, probleme en uygun algoritma yada çözüm yapısının belirlenmesi,
6. Problemin çözülmesi,
7. Çözümün değerlendirilmesi.

“Bu adımların takibinden sonra elde edilen çözümün istenilen düzeyde olup olmadığının kararı verilmelidir. Eğer çözüm yeterli değilse 5., 6., 7. adımlar en uygun çözüm bulunana kadar tekrarlanmalıdır (Biroğul, 2005, s.5)” [6].

“Problemin çözüm yöntemi belirlendikten sonra, problem bilgisayar ortamına aktarılır. Problem için veri girdi ve çıktılarının kararından sonra problemin çözüm yönteminin yada algoritmasının oluşturulması gerekir (Özel, 1999’a atfen Birogul, 2005, s.6)” - [35]

Aşağıdaki tabloda [Şekil 1.1 **Çizelgeleme problem çözümüne örnek yöntemler**, sınav ve ders çizelgeleme problem çözümleri için geliştirilmiş yöntemler, yıllara ve çalışmayı yapanlar bazında gösterilmektedir.

S. No	Yazar	Yılı	Yöntem	Çalışma Alanı
1	Burke E. K., Newall J.P.	1999	Genetik Algoritmalar	Sınav Çizelgeleme
2	Adamidis P., Arapakis P.	1999	Genetik Algoritmalar	Ders Çizelgeleme
3	Schaerf, A.	1999	Yerel Arama Tekniği	Ders Çizelgeleme
4	Zhaohui F., Lim A.	2000	Sezgisel Yaklaşım	Sınav Çizelgeleme
5	Abbas, A.M., Tsang, E.P.K.	2001	Kısıt Tatmin Teknikleri	Ders Çizelgeleme
6	Blum, C., Correia, S., Paechter, B., Rossi-Doria, O., Snoek, M.	2002	Genetik Algoritmalar	Ders Çizelgeleme
7	Burke E.K., Newall J.P.	2002	Yerel Arama Tekniği	Sınav Çizelgeleme
8	Tam V., Ting D.	2003	Sezgisel Yaklaşım	Ders Çizelgeleme
9	Sigl B., Golub M., Mornar V.	2003	Genetik Algoritmalar	Ders Çizelgeleme
10	Mahdi, O., Aïnon, R.N., Zainuddin, R.	2003	Genetik Algoritmalar	Ders Çizelgeleme
11	Legierski W., Widawski R.	2003	Kısıt Programlama	Okul Çizelgeleme
12	Merlot L.T.G., Boland N., Hughes B.D., Stuckey P.J.	2003	Kısıt Programlama	Sınav Çizelgeleme
13	Daskalaki S., Birbas T., Housos E.	2004	Tamsayı Programlama	Ders Çizelgeleme
14	Dimopoulou M., Miliotis P.	2004	Tamsayı Programlama	Ders Çizelgeleme
15	Piechowiak S., Kolski C.	2004	Karar Destek Sistemi	Ders Çizelgeleme
16	White G.M., Bill S. Xie B.S., Stevan Zonjic S.	2004	Tabu Arama	Sınav Çizelgeleme
17	Parthiban, P., Ganesh, K., Narayanan, S., Dhanalakshmi, R.	2004	Karar Destek Yaklaşımı	Ders Çizelgeleme
18	Yang Y., Petrovic S.	2004	Metasezgisel Yaklaşım	Sınav Çizelgeleme
19	Bhatt, V., Sahajpal R.	2004	Genetik Algoritmalar	Ders Çizelgeleme
20	Muller T., Rudova H.	2004	İteratif İleri Arama Algoritması	Ders Çizelgeleme
21	Kendall G., Mohd Hussin N.	2004	Tabu Arama	Sınav Çizelgeleme
22	Ross, P., Marin-Blazquez, J.G., Hart, E.	2004	Hipersezgisel Yaklaşım	Sınav Çizelgeleme
23	Daskalaki S., Birbas T.	2005	Tamsayı Programlama	Sınav Çizelgeleme
24	Zhang L., Lau S.K.	2005	Kısıt Tatmin Teknikleri	Ders Çizelgeleme
25	Cambazard H., Demazeau F., Jussien N., David P.	2005	Kısıt Programlama	Ders Çizelgeleme

**Şekil 1.1** Çizelgeleme problem çözümüne örnek yöntemler

### 1.1.1 Çizelgelemenin temel amaçları

Çizelgeleme problemlerinin çözümündeki temel amaçlar, kalitenin, kısıtların uygulanması ve düzenlemelerin en optimum seviyede sağlanmasıdır.

1. Kaynakların en etkin biçimde kullanılması
2. Sürenin en etkin biçimde kullanılması
3. Kısıtların belirlenmesi ve gerektiği şekilde kullanılması

### 1.1.2 Çizelgelemenin kullanıldığı problemler

Çizelgeleme birçok problemin çözümünde kullanılır.

1. Sınav tarih ve zamanlarına ders atanması.
2. Sınavların dersliklere atanması.
3. Sınavlara gözetmen atanması.
4. Ulaştırımda personel ve güzergah, saat planlaması
5. Projelere kaynak ve zamanlama ataması
6. İş çizelgelemeleri

## 1.2 Sınav Çizelgeleme Problemi ve Problem Çözümündeki amaçlar

Üniversitelerde, final sınavları için gerekli olan çizelgelemede karşılaşılan en önemli zorluk, elle yapılmaya çalışıldığında yaşanmaktadır.

Öğretim üyeleri açısından bakarsak, belirli gün ve tarihlerde, belirli dersler için ayırdıkları zamanlara göre sınav çizelgelemelerinin maximum oranda düzenlenmesi beklenir. Yönetsel olarak bakıldığında da derslik, zaman, öğretim üyesi kaynaklarından maximum fayda beklenmektedir. Öğrenciler açısından da sınavlara hazırlanma, sınavlar arasındaki gün/saat farkları, aynı gün ve saatte birden fazla sınavın olmaması gibi beklentiler bulunmaktadır.

Tüm bunların manuel olarak A4 kağıtlarının bir araya getirilmesiyle veya basit excel programlarıyla yapılması günlerce sürecek, hatta çok da maliyetli ve hataya açık bir iş olacaktır.

### 1.2.1 Sınav Çizelgelemenin temel amaçları

1. Kaynakların en etkin biçimde kullanılması
2. Sürenin en etkin biçimde kullanılması
3. Kısıtların belirlenmesi ve uygulanabilmesi
  - a. Yumuşak kısıtlar ( mümkün olduğunca uyulması beklenen kısıtlar)  
Örneğin; öğrencinin sınav saatlerinin çakışması.
  - b. Sert kısıtlar ( mutlaka uyulması gereken kısıtlar)  
Örneğin; sınav tarihlerinin belirlenmiş tarihlerde yapılması.

### 1.3 Sınav Çizelgeleme Probleminin Optimizasyon İşlemi

Optimizasyon işlemi, verilen bir problem için belirli kısıtlamalara dikkat ederek, mümkün olan en iyi çözümü bulmaktır. Başka bir deyişle, olası en büyük ya da en küçük  $f(x)$  değerini veren, belirli kısıtlamalar dahilinde olan bir  $x$  değerini bulmaktır.

Optimizasyon işlemlerinde, çözülmesi istenen probleme, uyulması istenen kısıtlamalar göz önünde tutularak; bir amaç fonksiyonu (objective function) yardımıyla uygun (feasible) çözümler bulunması hedeflenir. Gerçek hayatta karşılaşılan çoğu optimizasyon problemi matematiksel formülasyonlarla çözülemeyecek kadar karmaşıktır (April ve diğ., 2001) - [5].

Klasik yöntemleri kullanarak böyle bir problem çözülmeye çalışıldığında, çözüm günler sürebilir ve buna rağmen bulunan sonuç istenen şartları sağlayamıyor olabilir. Bu tip problemlere teknolojinin de desteğiyle sezgisel (heuristic) yaklaşımlar ile belli çözümler getirilebilmektedir.

Sezgisel algoritmalar, bir problemi çözmek için, çeşitli alternatif hareketlerden en iyi olana karar vermek amacı ile tanımlanan yöntemlerdir. Kesin çözümü bilinmeyen ancak çözümde olması istenen özelliklerin neler olması gerektiği bilinen, bu şekilde bulunan çözümün kalitesinin ölçülebileceği problemlerde sezgisel algoritmalar kullanılır.

Sezgisel algoritmalar genel olmaları, kolay uygulamaya geçirilebilmeleri ve en önemlisi çok değişik alanlardaki problemlere uyarlanabilmeleri nedeniyle ilgi çekmişlerdir. Genetik algoritmalar - GA, ısı işlem (simulated annealing ) ve tabu arama (tabu search), sezgisel algoritmaların en çok kullanılanlarındandır. Bu üç kombinasyonel optimizasyon algoritmasının bir çok ayırt edici özellikleri olmakla beraber aşağıda sayılan benzerlikleri vardır (Youssef ve diğerleri, 2000) - [50].

1. Optimal çözümlü bulmayı garanti etmezler,
2. Durmaları gereken zamanı bilmediklerinden durma kriteri verilmelidir,
3. Araştırmayı olumsuz etkileyecek bir durumu da kabul edebilirler,
4. Genel olduklarından herhangi bir kombinasyonel optimizasyon problemine
5. uygulanabilirler,
6. Şartlar sağlandığında optimal çözüme yakınsarlar.

Adı geçen sezgisel algoritmalar aynı problem sınıfına uygulandığında aşağıda belirtilen kriterlere göre karşılaştırılabilirler:

1. Her bir algoritma ile tanımlanmış en iyi çözümün kalitesi,
2. Başlangıç çözümünden, durma kriterlerinin sağlanmasına kadar araştırmanın izlediği seyir,
3. Çözüm bulmanın maliyeti.

Bir üniversite için final sınavı sınav çizelgesi hazırlanması gibi bir optimizasyon problemi için optimum çözümü elde edilmek isteniyorsa yukarıda adı geçen metodlara ihtiyaç vardır çünkü ders saatleri, akademik personel, derslikler gibi mevcut kaynaklar çoğu kez çakışmasız bir çizelgeye olanak tanımaz. Bu tezde incelenen sezgisel algoritmalarından genetik algoritma ve tabu arama teknikleri çizelgeleme atama probleminde 3 farklı algoritma ile ele alınacaktır.

1. Rastgele dağıtımın sürekli yinelenerek optimize edilmeye çalışılması
2. Çakışmaların en fazla olduğu dersten başlayarak dağıtım yapılması
3. Çakışmaların en az olduğu dersten başlayarak dağıtım yapılması

Bu tez çalışması için geliştirdiğimiz, çakışmaları optimize etmeye çalıştığımız algoritmada kullanılan veriler Maltepe Üniversitesi veritabanından test amaçlı sağlanmıştır. Excel olarak alınan veriler, SQL Server kullanılarak oluşturulan veritabanına ait ilgili tablolara aktarılmıştır. Kullandığımız belli başlı veriler şöyledir;

Herbir fakülteye ait ders kodları, bu derslerin şubeleri, öğretim elemanları, bu dersleri alan öğrenci numaraları, öğrenci adı soyadı, bölümü. Bölüm-Ders-Öğrenci ilişkileri bu ilişkisel veritabanında tutulmaktadır. Geliştirmiş olduğumuz algoritmalar, bu ilişkileri gözününe almaktadır.

Öncelikle kullanıcı dostu önyüz ekranlarından girdiğimiz sınav tarih ve saatlerini de veritabanında Sınav Dönemi bazında yaratıyoruz. Belirli bir oturum için belirli bir dersi rezerve edebiliyoruz. Tüm dersleri alan öğrencileri bir dizi de sınıflandırarak tutuyoruz. Kullanmak istediğimiz yöntemle, otomatik olarak sınav çizelgeleme yapabiliyoruz. Çakışmaları, günüçi ve aynı saattekiler farklı olacak şekilde renklendirebiliyoruz. Çakışan öğrencileri ve dersleri listeleyebiliyoruz.



## 2 LİTERATÜRDE KAYNAK ARAŞTIRMASI VE KULLANILAN ALGORİTMA ÖZETLERİ

Ross [48], “Tipik zaman çizelgeleme problemleri çeşitli kısıtları yerine getirerek, bir dizi aktivite yada olayın kaynaklara ve zaman periyoduna atanmasından ibarettir. Kısıtlar zaman çizelgelemenin doğasından ve kuruluşun kendine özgü karmaşıklığından meydana gelir. Başka bir deyişle, zaman çizelgeleme süreçlerin bir sıraya konması veya belirli bir amacı sağlamak için olayların zamana ve kaynak kısıtlarına göre kısmen sıralanmasıdır ve bazen belirli bir kısıt performansı sağlayarak olayları zaman süresince kaynaklara atama süreci olan çizelgeleme ile karıştırılır (bununla birlikte, çoğu bilim adamı çizelgelemenin zaman çizelgelemenin özel bir durumu olduğunu veya tam tersini düşünürler). (Ross, 1995’e atfen Norberciak, 2007, s.156)” - [33].

Wren, zaman çizelgelemeyi “hedeflere verilen kaynakların arzulan tüm amaçları mümkün olduğu kadar sağlayacak şekilde boş zamanlara yerleştirilmesi” olarak tanımlamıştır " ( Wren, 1996, s.53 )" - [48] .

“Zaman çizelgeleme problemleri genellikle katı (hard) ve yumuşak (soft) olmak üzere iki kategoriye ayrılan çok sayıda kısıta tabidir. Katı kısıtlar siddetle uygulanan ve zaman çizelgeleme düzenlemesinde mutlaka yerine getirilmesi gereken kısıtlardır. Yumuşak kısıtlar arzu edilen fakat mutlaka yerine getirilmesi gerekmeyen kısıtlardır " ( Norberciak, 2007, s.156 )" - [33] .

Zaman çizelgelemenin en yaygın örnekleri; hemşire, doktor, asker nöbet çizelgelemesi, ders programları, sınav programları, havayolları uçuş çizelgeleri, personel çizelgeleme vb. 'dir. Sınav çizelgeleme ( Sınav programı - Examination timetabling ) : “Bir dizi üniversite dersinin sınavlarının ortak öğrencileri olan sınavların çakışmasından kaçınarak ve mümkün olduğu kadar öğrenciler için sınavların yayılmasını sağlayarak çizelgelemesidir ( Schaerf, 1995, s.2 )" - [41].

Qu ve diğerlerine göre ( 2006, s.3 ), sınav çizelgeleme problemi bir dizi sınavın  $E = e_1, e_2, \dots, e_n$  bir dizi kısıtlama altında sınırlı sayıda zaman periyoduna  $T = t_1, t_2, \dots, t_n$  atanması olarak tanımlanabilir. Zaman çizelgeleme problemlerinin karmaşıklığı ve zorluğu farklı kuruluşlarda tatmin edilmesi gereken, bazıları diğerleriyle çelisen çok çeşitli kısıtlardan ileri gelir. Katı kısıt örnekleri: Çakışan sınavlar aynı zamana çizelgelenemez. Sınavı alacak öğrencilerin sayısı sınavın atandığı sınıfın oturma kapasitesini aşmamalıdır. Zaman çizelgelemede tüm katı kısıtların tatmin edilmesi genellikle olurluluk olarak anılır. Pratikte bütün yumuşak kısıtları tatmin eden olurlu bir çözüm bulmak genellikle imkansızdır. Sınav çizelgeleme literatüründe en yaygın yumuşak kısıt, öğrencilerin yeterince gözden geçirme zamanına sahip olması için sınavların mümkün olduğu kadar sınav dönemi boyunca yayılmasıdır. Zaman çizelgelemenin kalitesi genellikle üretilen çözümde ne kadar yumuşak kısıtın çiğnendiğinin bir ölçüsüdür [37].

## 2.1 Graf Tabanlı Teknikler

Welsh ve Powell, 1967 : Welsh ve Powell graf boyama ile zaman çizelgeleme arasında bir köprü kurmuştur, bu önemli miktarda zaman çizelgelemede graf sezgiselleri üzerine araştırmalara yol açmıştır (Qu ve diğ., 2006, s.7) - [39].

De Werra, 1985'de sınıf, öğretmen ve ders çizelgeleme için çeşitli matematiksel modeller sunmuş, graf boyama ve ağ akış tabanlı yöntemleri özet olarak gözden geçirmiştir (Qu ve diğ., 2006, s.4) - [39].

Burke, Newall ve Weare 1998'de rasgele bir elemanın graf sezgisellerinin çalışmasına etkisini incelemiştir (Qu ve diğ., 2006, s.8) . Burke ve Newall 2004'de iteratif bir süreçte problem çözerken sınavların uygun sıralandığı bir dinamik sıralama stratejisi araştırmıştır.

Asmuni ve diğerleri, 2004'de Toronto'da Bulanık mantığın kullanılması verilerinde graf boyama tabanlı sınav çizelgelemeye uygulanmıştır. [35].

## 2.2 Kısıt Tabanlı Teknikler

David [13] bir Fransız okulunda sınav çizelgeleme modeline kısıt tatmini tekniğini uygulamıştır.

Brailsford, Potts ve Smith, [7] kısıt tatmini problemleri üzerine çeşitli arama yöntemlerini tanıtmışlar ve bu tekniklerin optimizasyon problemlerine uygulanabileceğini göstermişlerdir.

Reis ve Oliveira [40] da kısıt mantıksal programlamada çeşitli uzantılar geliştirmek için platform teşkil eden prolog tabanlı bir sistem olan ECLiPSe tabanlı bir sınav çizelgeleme sistemi geliştirmişlerdir (Qu ve dig., 2006,s.10) - [39].

Merlot ve diğ. [29] Benzer bir yolla bir optimizasyon programlama dili olan OPL'i kullanarak başlangıç çözümleri üretmiştir. Daha sonra tavlama benzetimi (simulated annealing) ve tepe tırmanma (hill climbing) yöntemi kullanılarak bu çözümler iyileştirilmiştir.

## 2.3 Yerel Arama Teknikleri

Casey ve Thompson zaman çizelgelemede nispeten yeni bir teknik olan açgözlü rasgele adaptif arama prosedürleri (Greedy Random Adaptive Search Procedures ) yaklaşımını incelemişlerdir ( Qu ve dig, 2006, s.15 ).

## 2.4 Tabu Arama

Di Gaspero ve Schaerf [14] katı ve yumuşak kısıtların ihlallerine katkıda bulunan komşuluklarla ilgilenen tabu arama tabanlı tekniklerin bir türü üzerine değerli bir araştırma yapmıştır.

White ve Xie [46] Ottawa Üniversitesi'nde sınav çizelgeleme problemi için her safhada çok sayıda kısıtlar dikkate alınarak çözümlerin giderek iyileştirildiği, OTTABU olarak bilinen dört safhalı bir tabu arama yöntemi geliştirmiştir.

Paquete ve Stutzle [38] sınav çizelgeleme probleminde kısıtlar için verilen önceliklere göre düzenlenen bir tabu arama yöntemi geliştirmişlerdir.

## **2.5 Tavlama Benzetimi**

Thompson ve Dowsland [45] ilk safhada olurlu çözümler üreten ve ikinci safhada yumusak kısıtlarla ilgili tavlama benzetimi ile çözümlerin iyileştirildiği iki aşamalı bir yaklaşım geliştirmek için değerli bir çalışma yapmışlardır.

Merlot ve diğ. [29] kısıt programlama teknikleri ile başlatılan bir tavlama benzetimi yaklaşımı kullanmışlar ve ardından tepe tırmanma ile çözümler daha da iyileştirilmiştir.

Duong ve Lam HoChiMinh [17] teknoloji üniversitesinde sınav çizelgeleme problemleri için bir tavlama benzetimi yönteminin başlangıç çözümlerini kısıt programlama kullanarak oluşturmuşlardır.

Burke ve diğ. [9] tavlama benzetiminin bir türü olan Great Deluge algoritmasını incelemiştir.

Abdullah ve diğ., [1] farklı optimizasyon problemleri için Ahuja ve Orlin tarafından geliştirilen graf yapısı orjinli iyileştirme yöntemlerine dayanan geniş bir komşuluk arama yöntemi geliştirmişlerdir.

## **2.6 Genetik Algoritmalar**

Corne, Ross ve Fang [14] genel eğitimsel zaman çizelgelemede genetik algoritmalar kullanımı üzerine özet bir araştırma sunmuş ve bazı konular ile gelecekte olası gelişmeleri göstermiştir.

Erben [19], uygun kodlama ve uygunluk fonksiyonları incelenen bir grupta genetik algoritması geliştirmiştir.

Sheibani [44], sınavlar arasındaki boş zamanı en büyükleme amacı ile eğitim merkezlerinde zaman çizelgeleme problemlerinin çözümü için matematiksel bir model ve standart bir genetik algoritma kurmuştur.

Wong, Cote ve Gely Cole de Technologie Supérieure'de [47] kısıt tatmini problemi olarak modellenen bir sınav çizelgeleme probleminin çözümü üzerine uyguladıkları genetik algoritma hakkında bazı konuları tartışmıştır.

## **2.7 Karınca Algoritmaları**

Naji Azimi [32], sınav çizelgeleme problemlerinin sistematik biçimde çözülmesi için tasarlanan bir karınca kolonisi sistemi oluşturmuş ve onu tavlama benzetimi, tabu arama ve genetik algoritma ile karşılaştırmıştır. Sonuçların analizinde karınca sisteminin en iyi performans gösterdiği ve tabu aramanın rastgele oluşturulan başlangıç çözümünü yüksek düzeyde iyileştirdiği belirtilmiştir.

## **2.8 Üst Sezgisel Teknikler**

Kendall ve Hussin [27] 2003'deki çalışmalarını temel alarak hem hareket etme stratejileri hem de düşük seviye sezgiseller olarak yapıcı graf sezgiselleri kullanılan tabu arama tabanlı bir üst-sezgisel (hyper-heuristic) incelemişlerdir.

Yang ve Petrovic [49] Literatürde raporlanan çeşitli Toronto örnekleri için great delague algoritması ve en iyi sonuçları almak için başlangıç çözümünün oluşturulmasında graf sezgisellerinin seçiminde durum tabanlı muhakeme kullanılmıştır.

Qu ve Burke [39] birleştirilmiş graf tabanlı üst sezgisel çerçeve içinde farklı yüksek seviye arama algoritmalarının (dik iniş-steepest descent, tabu arama-tabu search, yinelemeli yerel arama - iterated local search, değişken komşuluk arama - variable neighbourhood search) kullanılmasının etkileri incelenmiştir.

Holland [24], “Doğada, biyolojik yapılar hayatta kalma ve yüksek oranda üreme ile kendi çevreleriyle mücadelede çok başarılıdırlar. Biyolojistler, doğada gözledikleri yapıları, çevrelerinde bir dönem boyunca işleyen Darwinian doğal seçimin sonucu olarak açıklarlar. Başka bir deyişle, bir dönem boyunca, doğadaki yapılar çevreye uygunluklarının sonucudur. Uygunluk, bir dönem boyunca, doğal seçim, seksüel rekombinasyon (genetik çaprazlama) ve mutasyon yoluyla yapıların yaradılışına neden olur (Holland'a atfen, Koza, 1993, s.1) - [28] ”.

Çevrelerinde görevlerini yapabilme yetenekleri daha iyi olan varlıklar (yani daha uygun bireyler), hayatta kalırlar ve yüksek oranda ürerler, daha az uygun olan varlıklar hayatta kalamazlar ve çok düşük oranda üreyebilirler. Bu Charles Darwin tarafından “On the Origin of the Specifies by Means of Natural Selection (1859) adlı eserde tanımladığı en iyinin hayatta kalması ve doğal seçim kavramıdır (Koza, atfen 1993, s.17) - [28] ”.

Genetik algoritma Darwinian evrimsel süreci ve doğal olarak kromozomlara genetik operasyonların uygulanmasını taklit eder (Koza, atfen 1993, s.18) - [28] .

Genetik Algoritmalar doğal seçim ve doğal genetik mekanizmalarına dayanan arama algoritmalarıdır. Onlar, bir arama algoritması kurmak için rasgele bilgi değişimi ile dizi yapılarının arasında en iyinin kurtulması ve insan araştırmasının yenilikçi sezgisel gücünü birleştirirler. Her nesilde, önceki nesilin en iyisinin tesadüfi yeni bir parçası denenerek, bitlerinin ve parçalarının kullanılması sonucu yeni yapay

diziler oluşturulur. Genetik algoritmalar basit rasgele arama yapmazlar, onlar geçmiş bilgilerden yararlanarak yeni araştırma noktaları düşünürler (Goldberg, 1989, s. 1) - [21].

Genetik algoritma, yönlendirilmiş rasgele araştırma algoritmalarının bir türüdür. Tabii seçme (natural selection) ile canlılarda bulunan genetik gelişimin benzetimini gerçekleştirmektedir. Algoritma diğer evrimsel algoritmalar gibi araştırma uzayında bulunan çözümlerin bazılarının oluşturduğu bir başlangıç popülasyonunu (Initial population) kullanmaktadır. Başlangıç popülasyonu her yeni nesilde (generation), tabii seçme (natural selection) ve üreme (reproduction) işlemleri vasıtasıyla ard arda geliştirilir. En son kusağın en uygun yani en kaliteli (fittest) bireyi, problem için optimal çözüm olmaktadır. Bu çözüm her zaman optimum olmayabilir ama kesinlikle optimuma yakın bir optimal çözümdür (Karaboga, 2004, s.78) - [26].

Holland [24], basit bit dizileri (bit strings) kullanarak karmaşık yapıların kodlanabileceğini göstermiştir. Yapılar, çözülecek problem için çözümleri temsil etmektedir ve dizi, kromozom yada birey adı verilmektedir. Bunlar, problemin araştırma uzayından alınır ve bu dizilerin belirli bir miktarı genetik algoritmanın kullanacağı popülasyonu oluşturur. Daha sonra temel genetik operatörler kullanılarak çözümler geliştirilir ve yeni nesiller üretilir. Eğer bu işlemler iyi kontrol edilirse, çözülecek probleme çok iyi uyarlanmış yapıları içeren çözüm kümesinin ortaya çıkmasını sağlar.

Even [20], zaman çizelgeleme problemini çözülmesi NP zor bir problem olarak tanımlamıştır. Sınav çizelgeleme problemi, bir zaman çizelgeleme problemidir ve problemin çözümü için hedef, her şube için derslerin istenen kısıtların göz önünde bulundurulması ile en uygun sınav çizelgesinin elde edilmesi işlemidir. Çizelgeleme problemlerinde kısıtlar, zorunlu olanlar yani keskin kısıtlar ve tercihe bağlı olanlar yani esnek kısıtlar olarak iki guruba ayrılır. Bir sınıfın aynı anda farklı iki dersinin sınavının olamayacağı veya aynı derslikte farklı iki dersin sınavının olamayacağı

keskin kısıtlara birer örnek iken, sınavların ilk saatlerde olması, esnek kısıt için örnek verilebilir.

Abramson, tavlama yöntemiyle çizelgeleme problemlerini çözmeye çalışmış ve altı farklı soğutma tekniğini karşılaştırmıştır. Kısıt bazlı mantık programlama, bu tür problemleri çözmek için kullanılan diğer bir yaklaşımdır [2], [16], [30].

Hertz ve Schaerf, çizelgelemeyi, tabu arama yöntemiyle çözmeye çalışmışlardır [23], [42].

Colorni, yaptığı deneylerin sonucunda, Genetik Algoritmaların (GA) ve tabu arama yönteminin daha iyi çizelgeler ürettiğini rapor etmiştir [12].

Erben, birbiriyle etkileşen kütüphanelere dayalı problemin ikili düzen ile temsil edildiği bir GA kullanmıştır ki muhtemelen testlerin yavaşlığının sebebi bunlardır [19].

Cladeira, problemi çözmek için, nesil ötesi Genetik Algoritma kullanmıştır. Burke, çalışmalarında üniversiteler için ders programı hazırlama ve sınav hazırlama problemleri için sezgisel çaprazlama ve mutasyon yöntemlerini kullanmışlardır. Çaprazlama operatörü olarak tek ve iki noktadan sezgisel çaprazlama, mutasyon operatörü olarak da tek noktan sezgisel mutasyonu kullanmışlardır [11].

Bu bölümdeki kaynakçalar : [4], [6], [34], [37].

## **2.9 Yazılım piyasasındaki sınav programı yapan benzer yazılımlar**

Piyasada sınav programı oluşturma işlevini gören birçok yazılım vardır. İşlevlerine göre maliyetleri artan yazılım örnekleri şöyledir.

Genetik algoritmayla ders çizelgeleme programı: ([www.hmyazilim.com/masa\\_ustu\\_yazilimi.php](http://www.hmyazilim.com/masa_ustu_yazilimi.php)). Geliştirilmiş olan bu program, kaynak kısıtlı ders



izelgelemeye y6nelik olarak genetik algoritma ile hazırlanmıřtır. Ama, belirli kısıtlar kapsamında ilgili d6nemde aılan b6t6n derslerin, dersi verecek 6ğretim elemanı ve dersin verileceėi řube ile eřleřtirilerek zaman izelgesine belirlenen dersliklere g6re yerleřtirilmesidir. Bu y6zden bu ders izelgeleme programı ( ders programı oluřturma programı ) genetik algoritma y6ntemi ile geliřtirilmiřtir

Lantiv timetabler 6 yazılımı: (<http://www.lantiv.com>) Lantiv Timetabler tahminen en oturmuř izelgeleme aracı olup, b6t6n eėitim kurumları iin kullanılabilir. Bu program b6t6n gerekli verileri (6ğretmen, 6ėrenci, malzeme vb.) veritabanına kaydedip daha sonra hatalardan ayıklayarak kullanıcıya g6stermektedir. B6t6n istenilen bilgileri ara y6z6nde g6steriyor olup kullanıcının bulmak istediėi bilgilere kolayca ulařmasını saėlamaktadır. Fakat yazılımın web desteėi yoktur. İhtiya duyulduėunda iletiřim bilgileri vardır. Bununla beraber web sayfası ierisinde bir forum bulunmaktadır. Bu sayede kullanıcıların birbiri ile haberleřmeleri ve sorunlarını kolayca paylařmaları saėlanmaktadır. Bu yazılım Lantiv International tarafından geliřtirilmiř olup kiři sayısı arttıka maliyet deėiřmektedir.

Mimosa scheduling yazılımı: ([www.mimosasoftware.com](http://www.mimosasoftware.com)) Mimosa b6t6n eėitim kurumlarının deėiřik eřit ve boyutta kapasiteleri iin uluslar arası izelgeleme programı oluřturma yazılımıdır. Bununla beraber bu yazılım konferans, iř yeri ve askeri kurumlar iin izelgeleme de yapabilmektedir. Bu sayede Mimosa iř ve end6stri evresinde de tercih edilen bir programdır. Bu programın hızlı, kullanıcı dostu ve ok b6y6k kapasiteli olduėu iddia edilmektedir. Bununla beraber program ok zengin eniyileme araları ve kullanıcı ile beraber yapılabilen birok izelgeleme bilgisine de sahip olduėu iddia edilmektedir. Kullanıcı dostu olan program yaygın kullanıcı hatalarını en alta indirmekte olduėu, kullanıcılara birok ipucu vermekte olduėu ve kullanıcı sorularına makalelerle veya web sayfasından cevap vermekte olduėu belirtilmektedir. Kullanıcı birok izelgeleme programını otomatik ya da man6el olarak oluřturabilmekte isterse interaktif olarak da oluřturabilmekte olduėu belirtilmektedir. Bu yazılım Mimosa Software Ltd. tarafından geliřtirilmiř olup kiři sayısına g6re yazılım maliyeti deėiřmektedir.

Calculix exam planner yazılımı: <http://www.examplanner.com/> adresinden online olarak sınav çizelgeleme yapılabilmektedir.

Özellikle yurtdışında birçok akademik kurumda da profesyonel kullanılan <http://www.thoughtimus.com/default.htm> 'den bilgilerine ulaşılabilen ticari yazılım, çok kapsamlıdır.

Bu bölümdeki kaynakçalar : [4]

### 3 LİTERATÜRDE KULLANILAN ÇÖZÜM YÖNTEMLERİ

#### 3.1 Genetik arama algoritması

Genetik Algoritmalar (GA) stokastik bir arama yöntemidir. Darwin'in "en iyi olan yaşar (survival of the fittest)" prensibine dayalı olarak biyolojik sistemlerin gelişim sürecini simüle eden GA, ilk defa Holland (1975) [24] tarafından önerilmiştir. Sezgisel bir metot olan GA, problem için optimum sonucu bulamayabilir, ancak bilinen metotlar ile çözüm zamanı problemin büyüklüğü ile üstel artan problemlerde optimal çözüme çok yakın çözümler vermektedir. Başlangıçta doğrusal olmayan (non-linear) optimizasyon problemlerine uygulanan GA, sonraları gezgin satıcı, karesel atama, yerleşim, atölye çizelgeleme, ders veya sınav programı hazırlanması gibi birleşik optimizasyon problemlerine de başarıyla uygulanmıştır.

Genetik algoritma, rastlantısal ve deterministik arama tekniklerini kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan sezgisel bir arama tekniğidir (Goldberg, 1989) [21]. Genetik Algoritmalar mümkün olan çözümlerin bir popülasyonu üzerinde işlem yapan olasılıklı (stochastic) arama algoritmalarıdır.

Geleneksel programlama teknikleriyle çözülmesi güç olan, özellikle sınıflandırma ve çok boyutlu optimizasyon problemleri, bunların yardımıyla daha kolay ve hızlı olarak çözüme ulaştırılmaktadır. Genetik algoritmanın çalışması doğal evrim sürecine benzetilir. Genetik algoritmalar doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretir. Bunun için "iyi"nin ne olduğunu belirleyen bir uygunluk (fitness) fonksiyonu ve yeni çözümler üretmek için yeniden kopyalama (recombination), değiştirme (mutation) gibi operatörleri kullanır.

##### 3.1.1 Genetik algoritmaların tarihçesi

İlk defa 1960' larda psikoloji ve bilgisayar bilimi uzmanı olan John Holland tarafından ortaya atılan; kendisi, öğrencileri ve meslektaşları tarafından geliştirilen

(Louis, 1993) G.A.'lar, mühendislik dünyasında birçok alanda genişçe işlenmiş, deneyleri yapılmış ve uygulanmıştır (Mitehell ve Forrest, 1993) . Ayrıca 1975'de G.A.'nın teorik çatısının verildiği John Holland'ın "Doğal ve Yapay Sistemlerde Adaptasyon" (Adaptation in Natural and Artificial Systems) isimli kitabı yayımlanmıştır. Holland'ın G.A.'sı Basit Genetik Algoritma (B.G.A.) olarak adlandırılmaktadır. Holland'ın asıl amacı, adaptasyon olgusunu tabiatta meydana geldiği şekliyle resmi olarak incelemek ve özel problemleri çözmek için algoritmalar düzenlemek yerine, doğal uyum mekanizmalarının bilgisayar sistemlerine aktarılabilceği yollar geliştirmektir (Tanrıseven, 2000). 1992'de, John Koza Genetik Algoritma'yı, verilen bazı görevleri gerçekleştirecek programlar geliştirmek için kullandı. Koza, bu yöntemle genetik programlama ("Genetic Programming" (GP)) adını vermiştir [4].

Mekanik öğrenme (machine learning) konusunda çalışan Holland, Darwin'in evrim kuramında etkilenerek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünmüştür. Tek bir mekanik yapının öğrenme yeteneğini geliştirmek yerine böyle yapılarda oluşan bir topluluğun çoğalma, çiftleşme, mutasyon, vb. genetik süreçlerden geçerek başarılı (öğrenebilen) yeni bireyler oluşturabildiğini görmüştür. Çalışmalarının sonucunu açıkladığı kitabının 1975'te yayınlanmasından sonra geliştirdiği yöntemin adı Genetik Algoritmalar olarak yerleşmiştir. Ancak 1985 yılında Holland'ın öğrencisi olarak doktorasını veren David E. Goldberg adlı inşaat mühendisi 1989'da konusunda bir klasik sayılan "Genetic Algorithm in Search, Optimization and Machine Learning" kitabını yayınladığı kadar genetik algoritmaların pek pratik yararı olmayan bir araştırma konusu olduğu düşünülmüştür. Hâlbuki Goldberg'in gaz boru hatlarının denetimi üzerine yaptığı doktora tezi ona sadece 1985'te National Science Foundation Genç Araştırmacı ödülünü kazandırmakla kalmadı, genetik algoritmaların pratik kullanımının da olabirliğini kanıtlamıştır. Ayrıca kitabında genetik algoritmalara dayalı tam 83 uygulamaya yer vererek GA'nın dünyanın her yerinde çeşitli konularda kullanılmakta olduğunu göstermiştir (Goldberg, 1989) [21].

Son yıllarda genetik algoritma, optimizasyon problemlerinin çözümü için büyük önem kazanmıştır. Genetik algoritmanın son yıllardaki çalışmaları arasında haberleşme şebekeleri tasarımı, elektronik devre dizaynı, gaz boruları şebeke optimizasyonu, görüntü ve ses tanıma, veri tabanı sorgulama optimizasyonu, uçak tasarımı bulunmaktadır.

### **3.1.2 Genetik algoritmanın tanımı ve çalışma prensibi**

Genetik Algoritmalar (GA) ya da daha geniş kapsamıyla Evrimsel Algoritmalar (EA), doğadaki evrimsel süreçleri model olarak kullanan bilgisayara dayalı problem çözme teknikleridir. Geleneksel programlama teknikleriyle çözülmesi güç olan, özellikle sınıflandırma ve çok boyutlu optimizasyon problemleri, bunların yardımıyla daha kolay ve hızlı olarak çözülebilmektedir.

Algoritma ilk olarak popülasyon diye tabir edilen bir çözüm (kromozomlarla ifade edilir) seti ile başlatılır. Bir popülasyondan alınan sonuçlar bir öncekinden daha iyi olacağı beklenen yeni bir popülasyon oluşturmak için kullanılır. Yeni popülasyon oluşturulmasında iyinin seçilmesine olanak veren bir uygunluk (fitness) fonksiyonu kullanılır. Daha sonra bu oluşan popülasyondaki kromozomlar doğada olduğu gibi çaprazlama (cros-over) ve değişim (mutation) işlemine tabi tutulurlar. Bunun sonucunda oluşan popülasyonun en güçlü kromozomu (fitness fonksiyonunu maksimum yapan kromozom) bir tabloya yazılır. Bu bir önceki popülasyondan yeni popülasyon üretme ve popülasyonun en güçlü kromozomunu bularak tabloya ekleme işlemi defalarca yapılır. Bu işlemin yapılma sayısı ne kadar fazla olursa algoritmamız o kadar iyi bir sonucu elde etmiş olur. Tüm bu işlemler bitirilip her popülasyondaki güçlü kromozomların bulunduğu tablo oluşturulduktan sonra bu tablodaki kromozomlardan uygunluk değeri en büyük olan, yani fitness fonksiyonunu maksimum yapan kromozom sonuç olarak alınır.

### **3.1.3 Gen**

Tek başına anlamı olan en küçük veri birimidir. Sınav programı çizelgelemesi için gen dönem ders kodu, öğrenci sayısı, derslik kodu, sınav tarihi, başlama saati,

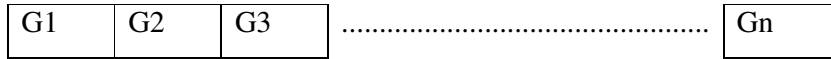
bitiş saati, sınıfı, şubesi, dönemi ve öğretim elemanı bilgilerinden oluşmuş bir veridir.

Dönem Dersi Kodu	Sınav Tarihi	Öğrenci Sayısı	Başlama Saati	Bitiş Saati
Sınıfı	Şube	Dönemi	Derslik	Öğretim Elemanı

**Şekil 3.1** Sınav programı çizelgeleme gen yapısı

#### 3.1.4 Kromozom

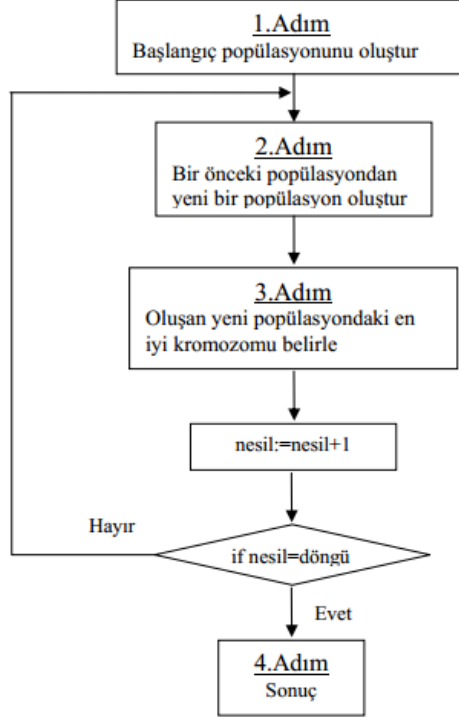
Bir veya birden fazla gen yapısının bir araya gelerek problemin çözümüne ait bilgilerin bir kısmını oluşturan dizilere kromozom denir (Biroğul, 2005). Kromozomlar alternatif çözümlerdir. Popülasyondaki bireylere karşılık gelirler. GA'da kromozom sayısının fazla olması çeşitlilik açısından avantaj sağlar fakat aynı zamanda fazla kromozom demek çözüm zamanının uzaması anlamına gelir. Kromozom sayısının tespiti problemlere göre değişse de literatürde kromozom sayısında 20-30 arası değerlerin kullanıldığı gözlenmiştir.



**Şekil 3.2** Gen dizisi

Tablo n adet gene sahip bir kromozom yapısı Sınav programı çizelgelemesi için kromozomdaki gen sayısı bölümdeki dönem derslerinin sınavlarının sayısı kadar olacaktır.

### 3.2 Genetik algoritmaların akış diyagramı



Şekil 3.3 Genetik algoritmalar için genel akış diyagramı

#### 3.2.1 Akış diyagramının açıklaması

- 1. Adım:** Bu adımda başlangıç için rastgele seçilen bir popülasyon oluşturulur.
- 2. Adım:** Bir önceki popülasyondan yeni bir popülasyon oluşturulur. İlk olarak mevcut popülasyondaki kromozomların seçilme olasılıkları hesaplanır ve bir tabloya aktarılır. Olasılık = kromozomun değeri / popülasyondaki kromozom değerleri toplamı Daha sonra bu kromozomlar seçilir. Bu seçimi yapabilmek için rulet tekerleği seçimi (roulette wheel selection), turnuva seçimi (Tournament Selection) gibi seçme yöntemleri vardır. Bu problemin çözümünde rulet tekerleği seçim yöntemi kullanılmıştır. Rulet tekerleği yönteminde seçilme olasılıklarını tuttuğumuz tablodaki sayılar birbirine eklenerek kümülatif toplamları alınır. Bu toplamlar aralığı her kromozomun seçilebilme aralığıdır. Daha sonra popülasyondaki kromozom sayısı kadar 0 ile 1 aralığında rastgele sayı üretilir. Bu üretilen rastgele sayılar hangi aralığa denk geliyorsa o birey

seçilmiş olur. Bu kromozomlar gen bazında belirlenen çaprazlama oranı ve şekline göre çaprazlanır. Genler belirlenen mutasyon oranı ve şekline göre mutasyona uğratılır. Tüm bu işlemler sonucu yeni bir popülasyon oluşturulmuş olur.

3. **Adım:** Oluşturmuş olduğumuz yeni popülasyondaki kromozomlar için fitness (uygunluk) fonksiyonu hesaplanarak bu popülasyondaki en iyi uygunluk değeri olan birey seçilir. Bu seçilen birey nesillerin en iyi bireylerinin yer aldığı tabloya eklenir.
4. **Adım:** Bu adım algoritmanın en son adımıdır. Nesillerin en iyi uygunluk değerine sahip kromozomlarının bulunduğu tablodaki en iyi uygunluk değeri olan birey seçilir. Yeni bireyler uyumluluğa göre veya rastgele olarak seçilebilir. Yeni bireylerin tamamen rastgele seçilme durumunda yakınsama zorlaşabilir. Tüm bireyler uyumluluğa göre seçildiğinde ise yeni kuşak içinde bölgesel yakınsamalar olabilir. Bu sorunların üstesinden gelmek için belli bir oranda uyumluluk seçimi belli bir oranda da rastgele seçim yapılabilir. Bu oran Kuşak farkı (Generation gap) ile ifade edilir. Kuşak farkı %100 olduğunda yeni bireylerin tamamı uyumluluğa göre seçilir.

### 3.3 Çaprazlama yöntemi

Zaman çizelgeleme problemlerine uygun altı farklı çaprazlama yöntemi test edilmiştir. Bunlar; Pozisyona dayalı, Sıraya dayalı, Kısmi planlı, Dairesel, Doğrusal ve Sıralı Çaprazlama yöntemleridir.

**a. Pozisyona dayalı çaprazlama yöntemi:** Bu yöntemde rastlantısal olarak seçilmiş pozisyondaki işler, bir ebeveynden çocuğa kalıtsallaştırılır. Diğer işler diğer ebeveynde buldukları sıra ile yerleştirilir. Öncelikle pozisyondaki sayılar,  $[1, n]$  rastlantısal tamsayılar şeklinde düzenlenir, daha sonra bu pozisyonlar rastlantısal olarak seçilir. Her pozisyonun çaprazlama olasılığı %50 dir (Murata vd, 1996) [31].

**b. Sıraya dayalı çaprazlama yöntemi:** Bu yöntemde bir grup nokta rastgele seçilir. Birinci kromozomun seçilen noktalara karşılık gelen karakterleri aynen



yerlerini korur. İkinci kromozomun seçilen noktalara ait karakterleri birinci kromozomun aynı noktalarındaki karakterlerin arkasına getirilir. Geriye kalan boş pozisyonlara ikinci kromozomdan aktarılan yeni karakterler de göz önünde bulundurularak ilk kromozomun kullanılmayan karakterleri sıra ile (soldan sağa) yerleştirilerek yeni bir kromozom elde edilir (Cheng, 1999) [10]. Bu tür çaprazlama, kromozomu oluşturan karakterlerin sayısı ve sıralarının önem taşıdığı durumlarda kullanılır.

**c. Kısmi planlı çaprazlama:** Goldberg tarafından geliştirilen bu çaprazlama ilk olarak gezgin satıcı probleminde (TSP) kullanılmıştır. Bu yöntemde iki ayrı iş sırasında rastlantısal olarak aralıklar belirlenir ve bu aralıkta yer alan işlerin yeri karşılıklı olarak değiştirilir (Goldberg, 1989) [21]. Bu yöntem aşağıda bir örnek üzerinde açıklanmaktadır: Çaprazlama için seçilen ebeveyn yapılar, A ve B olarak adlandırılmıştır ve sekiz iş içermektedir. A= 2, 8, 6, 4, 5, 7, 1, 3 ve B = 8, 7, 6, 4, 5, 4, 6, 5 'ye kısmi planlı çaprazlama operatörü uygulanır ise, A ve B.den ilk olarak ortak bir aralık rastlantısal bir şekilde seçilir. Daha sonra, seçilmiş iki aralıktaki elementlerin değişim planları belirlenir. İkinci olarak, A ve B. deki iki aralık karşılıklı değiştirilir. Bir dizide bir iş tekrarlı olarak yer aldığından her iki yapının da uygun olmadığı görülmektedir ve buradan elde edilen yapılar geçici sonuçlar olarak değerlendirilir.

Bundan dolayı, yeni yapıları uygun olmayan A ve B. de, değişim planının uygulanması gerekir. Bu örnekte, A yapısının 1, 7 ve 8 pozisyonlarında yer alan; 2, 1 ve 3 değerleri sırasıyla 6, 4 ve 5 değerleriyle değiştirilir. B yapısında ise 6, 7, ve 8 pozisyonlarındaki 4, 6 ve 5 değerleri sırasıyla 1, 2 ve 3 tarafından değiştirilir. Yeni yapıda A= 6, 8, 2, 1, 3, 7, 4, 5 ve B = 8, 7, 6, 4, 5, 1, 2, 3 şeklinde olur.

**d. Dairesel çaprazlama:** Davis, Goldberg ve Lingle tarafından geliştirilmiş bir yöntemdir [18]. Bu yöntemde ilk kromozomdan en baştaki gen seçilir ve bu gen yeni diziyeye yerleştirilir. Bu gene karşılık gelen ikinci kromozomdaki gen belirlenir bu değer de yeni kromozom üzerine yerleştirilerek dairesel bir şekilde bütün genler belirlenir (Goldberg, 1989) [21].

**e. Doğrusal sıralı çaprazlama:** Falkenauer ve Bouffouix tarafından geliştirilmiştir. Dairesel çaprazlamanın bir varyantıdır. İşlem adımları (Cheng, 1999) aşağıda verilmektedir: [10]

1. Mevcut popülasyon içerisinde rastlantısal olarak iki ebeveyn seç,
2. Seçilen bu iki dizi (kromozom) üzerinde rastlantısal olarak iki alt dizi seç,
3. P1 dizisinden seçilen alt diziyi kromozomdan kopar ve boş kalan yerleri belirle, benzer şekilde P2 dizisinde de aynı işlemleri gerçekleştir,
4. Birinci alt diziyi P1. ye ve ikinci alt diziyi P2. ye yerleştir.

**f. Sıralı çaprazlama:** Bu yöntem de, Davis, Goldberg ve Lingle tarafından [18] geliştirilmiştir (Goldberg, 1989) - [21]. Bu yöntemde, gen havuzundan rastlantısal olarak iki kromozom seçilir. Bu kromozomlar üzerinde yine rastlantısal olarak iki ayrı kesim noktası belirlenir. Bu kesim noktaları arasındaki kromozom sayısının her iki kromozomda da aynı olmasına dikkat edilir. Kesim noktaları arasındaki kromozomlar karşılıklı olarak yer değiştirilir. Kesim bölgesi dışında yer alan genler içerisinde tekrarlı genler oluşursa bunlar yerine sıra ile soldan sağa doğru kromozomda bulunmayan genler yazılır.

### 3.4 Popülasyon büyüklüğü

Popülasyon büyüklüğü, problemin çözüm süresini etkilemektedir. Problemin özelliğine göre seçilecek olan popülasyondaki birey sayısı genetik algoritmayı hazırlayan tarafından iyi belirlenmelidir. Bu parametre popülasyon içinde (yalnızca bir kuşakta) kaç adet kromozom yani birey olduğunu gösterir. Eğer kromozom sayısı az olursa GA çözüm aranan uzayın ancak bir kısmını gezebilir ve çaprazlama için fazla bir seçeneği yoktur. Kromozom sayısı çok fazla olursa GA çok yavaş çalışır. Araştırmalar belli bir noktadan sonra popülasyon sayısını artırmanın bir yararı olmadığını göstermiştir.

Yeni bireyler uyumluluğa göre veya rastgele olarak seçilebilir. Yeni bireylerin tamamen rastgele seçilme durumunda yakınsama zorlaşır. Tüm bireyler uyumluluğa göre seçildiğinde ise yeni kuşak içinde bölgesel yakınsamalar olabilir.

Bu sorunların üstesinden gelmek için belli bir oranda uyumluluk seçimi belli bir oranda da rastgele seçim yapılabilir. Bu oran Kuşak Farkı (Generation Gap) ile ifade edilir.

### 3.5 GA'nın uygulama alanları

Genetik algoritmaların en uygun olduğu problemler geleneksel yöntemler ile çözümü mümkün olmayan ya da çözüm süresi problemin büyüklüğü ile üstel orantılı olarak artanlardır. Bugüne kadar GA ile çözümüne çalışılan konulardan bazıları şunlardır :

**Optimizasyon:** GA, sayısal optimizasyon ve kombinatoral optimizasyon problemleri olan devre tasarımı, doğrusal olmayan denklem sistemlerinin çözümünde ve fabrika üretim planlamasında kullanılır.

**Otomatik programlama (automatic programming):** GA, bilgisayar programları yardımıyla network sıralamasında (sorting),ders programı hazırlanmasında kullanılır. Makine öğrenmesi (machine learning): GA, robot sensorlerinde, neural networklerde, VLSI yonga tasarımı ve protein yapısal analizinde kullanılır.

**Ekonomi (economics):** GA, ekonomik modellerin geliştirilmesinde ve işleminde kullanılır.

**Bağışık sistemler (Immune systems):** GA, çok-gen'li ailelerin evrimi esnasında ve doğal immün sistem modellerinde kullanılır.

**Popülasyon genetiği (population genetics):** GA, evrim ile ilgili sorulara cevap bulmada kullanılır.

**Evrim ve öğrenme (evolution and learning):** GA, fertlerin öğrenmesini ve türlerin evrilmesinde kullanılır.

**Sosyal sistemler (social systems):** GA, sosyal sistemlerin analizinde kullanılır.

Bu bölümdeki kaynakçalar : [3], [4], [18], [34]

## 4 GELİŞTİRDİĞİMİZ ALGORİTMALAR VE YAZILIM

### 4.1 RSINA Rastgele Sınav Atama Algoritması

Adım	İşlem
1	Dönem seç
2	Döneme ait dersleri seç
3	Dönemdeki dersleri alan öğrencileri seç
4	Herbir öğrenci için DersID dizisi oluştur.
5	Aynı DersID'yi alan öğrencilerin sayısını bir dizide topla
6	Sınav tarihi ve saatlerini seç
7	Rastgele dersleri oturumlara dağıt
8	Yeni Sınav Çizelgeleme oluştur.
9	Database'i güncelle

### 4.2 SINA1 Sınav Atama Algoritması

Adım	İşlem
1	Dönem seç
2	Dönemdeki dersleri seç
3	Dönemdeki bu dersleri alan öğrencileri seç
4	Herbir öğrenci için DersID dizisi oluştur.
5	Bu iki dizinin kesişimlerini al
6	Bu kesişimleri bir diziye al
7	Herbir DersID için diğer derslerle çakışma matrisi oluştur.
8	Herbir DersID'nin diğer derslerle çakışmaların toplamlarını al
9	Bu toplamları büyükten küçüğe doğru sırala
10	Derslerin sınav saatlerini en büyük çakışmadan başlayarak sırasıyla ata
11	Yeni Sınav Çizelgeleme oluştur
12	Database'i güncelle

### 4.3 SINA2 Sınav Atama Algoritması

Adım	İşlem
1	Dönem seç
2	Dönemdeki dersleri seç
3	Dönemdeki bu dersleri alan öğrencileri seç
4	Herbir öğrenci için DersID dizisi oluştur
5	Bu iki dizinin kesişimlerini al
6	Bu kesişimleri bir diziye al
7	Herbir DersID için diğer derslerle çakışma matrisi oluştur
8	Herbir DersID'nin diğer derslerle çakışmaların toplamlarını al
9	Bu toplamları küçükten büyüğe doğru sırala
10	Derslerin sınav saatlerini en küçük çakışmadan başlayarak sırasıyla ata
11	Yeni Sınav Çizelgeleme oluştur
12	Database'i güncelle

#### 4.4 SINA1 ve SINA2 AKIŞ DİYAGRAMI



**Şekil 4.1** Algoritmalarımız için genel akış diyagramı

Sınavlardaki öğrencilerin çakışmalarını optimize etmeye çalıştığımız algoritma sonucunda sınavı olacak dersler, sınav saatleri ve oturumlarına atanmış olmaktadır. Bu oturumlara atama işlemine, eğer küçükten büyüğe ( SINA2 ) çakışmaları aldıysak, en az çakışması olan dersten başlayarak her zaman ilk oturumdan başlıyoruz. Çakışma matrisinden aldığımız en küçük çakışması olan dersi,

2. takvim gününün ilk oturumuna atıyoruz. Böylece en büyük çakışan derslere kadar tüm matris için sınav oturumu ataması gerçekleştiriyoruz.

Çakışmaları büyükten küçüğe ( SINA1 ) olarak atamalarımızı yaparsak aynı işleme en fazla çakışan dersten başlayarak yaparız. Bu algoritmaları, tasarladığımız yazılımın sistem parametrelerinden seçebiliriz ve atama modellerini de kodda yapacağımız değişikliklerle farklılaştırma olanağımız vardır.

#### **4.5 KISITLAR**

- Sınavların yapılacağı tarih aralığı belirlenir.
- Sınavların yapılacağı saat aralıkları belirlenir.
- Sınavın yapıldığı yıl, dönem ve ara sınav mı yıl sonu sınavı mı olduğu girilir.
- Özel durumu olan derslerin sınavlarının belirli bir oturum için rezerve edilmesi.

#### **4.6 SİSTEM MİMARİSİ VE GELİŞTİRİLMESİ**

Sistem tasarlanırken benzer yazılımlar araştırılmış olup sistem en iyi biçimde tasarlanmaya çalışılmıştır. Burada tanıtılan yazılımların objektifleri farklıdır. Ticari amaçlı araçlar olup oldukça pahalıdır ve Türkçe dil desteği yoktur. Yazılımın oluşturulma kriterleri aşağıdaki şekilde özetlenebilir:

- .NET FRAMEWORK 4.0, MS-SQL SERVER 2008 (Kullanılan Veritabanı), VISUAL C# (Kullanılan Programlama Dili) Developer Express (Windows Form Uygulaması için Kullanıcı Arayüz Kontrolleri)
- 3 farklı dökümantasyonun olması ve sistem parametresinden seçilerek 3 farklı algoritma ile çalışması.
- Yeni algoritmalar eklenmesine açık olması.

##### **4.6.1 Sistem tasarımı**

Sistem tasarımı için öncelikle bir veritabanı oluşturulmuştur.

#### 4.6.1.1 Veri Girişİ

SSIS kullanılarak okulun veritabanından dersler , öğrenciler ve öğretim üyeleri ile ilgili bilgileri içeren Scheduling\_Exams veri tabanı oluşturulur.

_BOLUM_DERS	Bir bölümdeki dersler tablosu. Bolum_Id, Ders_Id,Sınıf
_BOLUMLER	Bölmülerin bilgisi tablosu. Bolum_Id, Bolum_Ad
_DERS_ALT_SUBE	Derslerin alt şubeleri tablosu. Alt_sube_Id, Sube_Id, Ders_Id
_DERS_SUBE	Derslerin şubeleri tablosu. Sube_Id, Ders_Id
_DERSLER	Dersler tablosu. Ders_Id, Ders_Adi_Tr, Ders_Adi_En, K, U, L,AKTS
_DERSLİKLER	Derslik_Id, Derslik_Adı, Kapasite
_OG_BOLUM	Öğretim görevlisi bölüm ilişkisi tablosu. Ogretim_Grv_Id, Bolum_Id
_OGRENCI_DERSLERI	Öğrencilerin ilgili sınav dönemi için almış oldukları dersler tablosu. Ogrenci_No, Bolum_Id, Ders_Idi Ogretim_Grv_Id, Notu
_OGRENCILER	Öğrenci bilgileri tablosu. Ogrenci_No, Ad, Soyad, Yönetmelik, Gano, Sınıf
_OGRETIM_GOREVLILERI	Öğretim görevlileri detay bilgileri. Ogretim_Grv_Id, Ad, Soyad, Unvan, Unvan_KT, Kademe
_UNVANLAR	Öğretim görevlileri ünvanları. Unvan_Id, Unvan, Unvan_KT

Geliştirilen kullanıcı arayüzleri ile veri girişi yapılan tablolar

_SINAV_ATAMA	Database 'de sınav atamalarının son hali tutulur. Ders_Id, Sınav_Tarihi, Sınav_Saat, Donem
_SINAV_REZERVE	Öğretim görevlileri detay bilgileri.

	Ogretim_Grv_Id, Ad, Soyad, Unvan, Unvan_KT, Kademe
_SINAV_TARİHLERİ	Öğretim görevlileri ünvanları. Unvan_Id, Unvan, Unvan_KT

Bu sistemde, üniversitede  $n$  öğrenci ve  $m$  ders verildiğini varsayalım. Öğrencilerin bir dönemde aldıkları dersler \_DERSLER tablosunda verildiğini varsayarsak bu tablonun elemanı şu şekilde tanımlanır.

Öğrenci  $i$ , ders  $j$ 'i alıyorsa =1, almıyorsa =0,  $i=1, \dots, n$ ;  $j=1, \dots, m$ .

(Bir dersin şubeleri ayrı bir ders olarak tanımlanır).

	1.Ders	2.Ders	3.Ders	4.Ders	5.Ders	6.Ders	7.Ders	j.Ders	n.Ders
Öğrenciler	0	1	0	1	0	1	1	1	1
1.Öğrenci	1	0	0	0	0	0	0	0	1
2.Öğrenci	0	1	0	0	1	1	1	1	0
3.Öğrenci	0	1	1	1	0	1	0	1	1
4.Öğrenci	0	0	1	1	1	1	0	1	0
i.Öğrenci	1	0	0	0	0	0	1	0	0
n.Öğrenci	1	1	1	1	1	0	1	0	1

Şekil 4.2 Dersler tablosu

Öğrenci  $i$ , ders  $j$ 'yi alıyorsa = 1, almıyorsa = 0 şeklinde ifade edilir.  $i=1, \dots, n$ ;  $j=1, \dots, m$ .

Bu mantıkla, algoritmamız için derslerin çakışma matrisini de oluşturmak mümkündür.

#### 4.6.2 SINA1 ve SINA2 algoritmaları için Çakışma Matrisi Bulma

Herbir dersi alan öğrenciler class'ı oluşturulur. Herbir öğrencinin aldığı dersler class'ı oluşturulur. Her iki kümenin kesişimleri bulunarak, dersleri alan öğrencilerin çakışmaları bu çakışma matrisinde bulunur.

```
List<Lesson> lessons = new List<Lesson>();
List<QryDersOgrenci> Ogrenciler = (List<QryDersOgrenci>)DersOgrenci.Get(donem);
Ogrenciler.Sort("DersId asc");
Lesson lesson = null;
Student student = null;
foreach (var item in Ogrenciler)
{
```



```

if (lesson == null || lesson.LessonId != item.DersId)
{
    if (lesson != null)
        lessons.Add(lesson);
    lesson = new Lesson();
    lesson.LessonId = item.DersId;
}

student = new Student();
student.StudentId = item.OgrenciNo;
student.LessonList = GetLessonList(Ogrenciler, student.StudentId);
lesson.StudentId.Add(student);
}

//Count of intersection
foreach (var mainLesson in lessons)
{
    foreach (var innerLesson in lessons)
    {
        mainLesson.GetIntersectCount(innerLesson);
    }
}

```

DERSLER	BİL 101	BİL 102	BİL 133	BİL 134	BİL 201	BİL 202	BİL 203	BİL 204	BİL 206	BİL 221	BİL 235
BİL 101	58	58	58	58	37	37	37	37	37	37	37
BİL 102	58	58	58	58	37	37	37	37	37	37	37
BİL 133	58	58	58	58	37	37	37	37	37	37	37
BİL 134	58	58	58	58	37	37	37	37	37	37	37
BİL 201	37	37	37	37	38	38	38	38	38	38	38
BİL 202	37	37	37	37	38	38	38	38	38	38	38
BİL 203	37	37	37	37	38	38	38	38	38	38	38
BİL 204	37	37	37	37	38	38	38	38	38	38	38
BİL 206	37	37	37	37	38	38	38	38	38	38	38
BİL 221	37	37	37	37	38	38	38	38	38	38	38
BİL 235	37	37	37	37	38	38	38	38	38	38	38

**Şekil 4.3** Derslerin birbirleriyle çakışma matrisi

Herbir dersin, diğer derslerle kesişim kümelerinin toplamları alınır.

```

for (int i = 0; i < lessons.Count; i++)
{
    DataRow dr = dt.NewRow();
    for (int j = -1; j < lessons[i].IntersectLesson.Count + 1; j++)
    {
        if (j == -1)
        {
            dr["DERSLER"] = lessons[i].LessonId;
        }
        else if (j == lessons.Count)
        {
            dr["TOPLAM"] = 0;
        }
        else
        {

```

```

        dr[lessons[j].LessonId] = lessons[i].IntersectLesson[lessons[j].LessonId].ToString();
    }
}
dt.Rows.Add(dr);
}

```

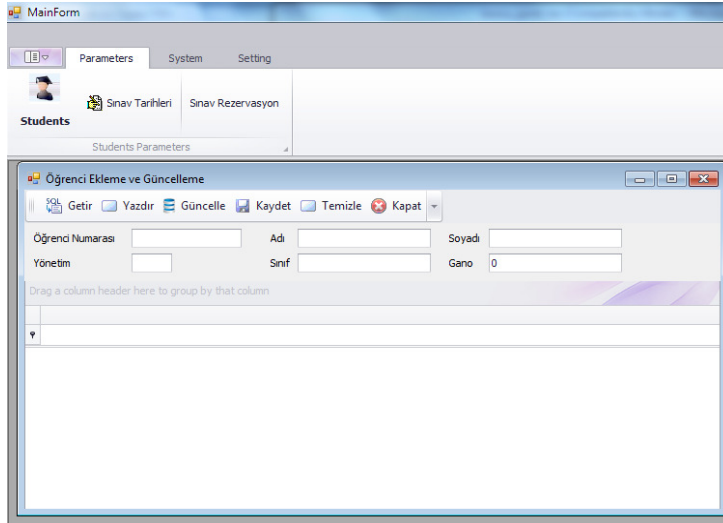
### 4.6.3 Kullanıcı Arayüz ekranları

Sınav çizelgelmelerinin kolaylıkla yapılabilmesi için, kullanıcı dostu arayüz uygulamaları geliştirilmiştir. Bu ekranların başlıca uygulamaları şöyledir ;

- Öğrenci bilgileri
- Sınav tarih ve saat bilgileri
- Sınav atama
- Sınav takvimi oluşturma
- Fonksiyonlar ;

Giriş, Görüntüleme, Silme, Güncelleme, Sürükle bırak.

#### 4.6.3.1 Öğrenci Bilgileri



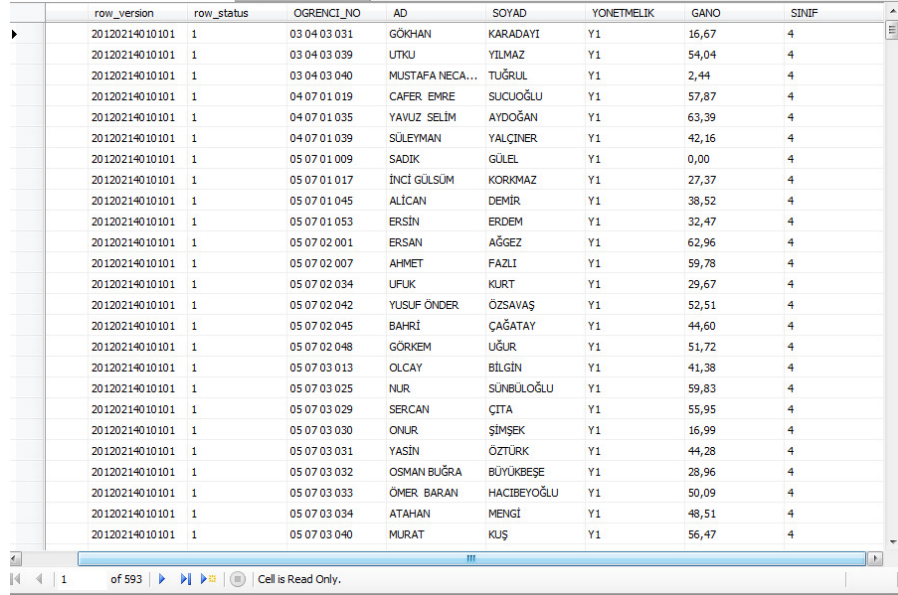
Şekil 4.4 Öğrenci görüntüleme, ekleme, güncelleme ekranı

**Getir** fonksiyonu ile, Maltepe Üniversitesi sisteminde kayıtlı tablolardan insert edilen öğrenci-ders tablolarındaki datalar görüntülenir.

Yeni öğrenci girişi yapılarak **Güncelle** fonksiyonu ile database'e insert edilebilir.

Veya mevcut bir öğrenci ile ilgili güncelleme de yapılabilir.

**Yazdır** fonksiyonu ile öğrencilerin raporlaması yapılabilir.

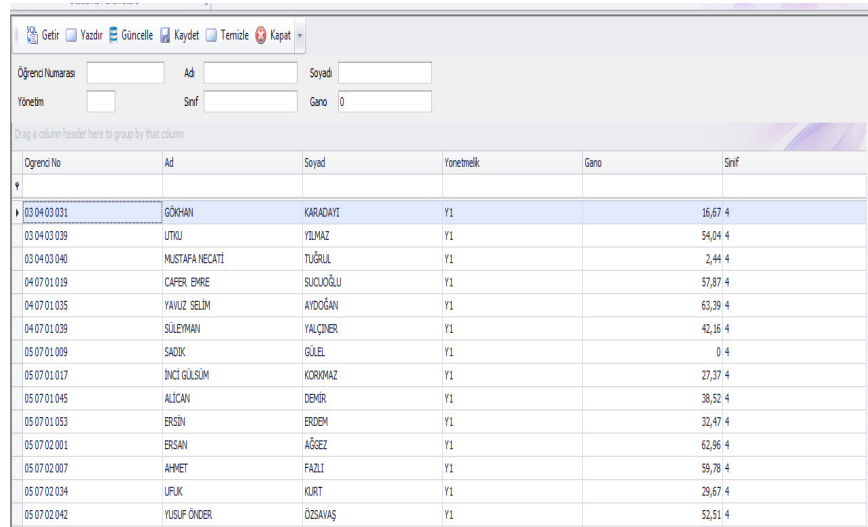


The screenshot shows a database table with the following columns: row\_version, row\_status, OGRENCI\_NO, AD, SOYAD, YONETMELIK, GANO, and SINIF. The table contains 593 rows of student data. The first few rows are:

row_version	row_status	OGRENCI_NO	AD	SOYAD	YONETMELIK	GANO	SINIF
20120214010101	1	03 04 03 031	GÖKHAN	KARADAYI	Y1	16,67	4
20120214010101	1	03 04 03 039	UTKU	YILMAZ	Y1	54,04	4
20120214010101	1	03 04 03 040	MUSTAFA NECA...	TUĞRUL	Y1	2,44	4

Şekil 4.5 Database \_OGRENCILER tablosu

\_OGRENCILER tablosunda toplam 593 öğrenci kayıtlıdır



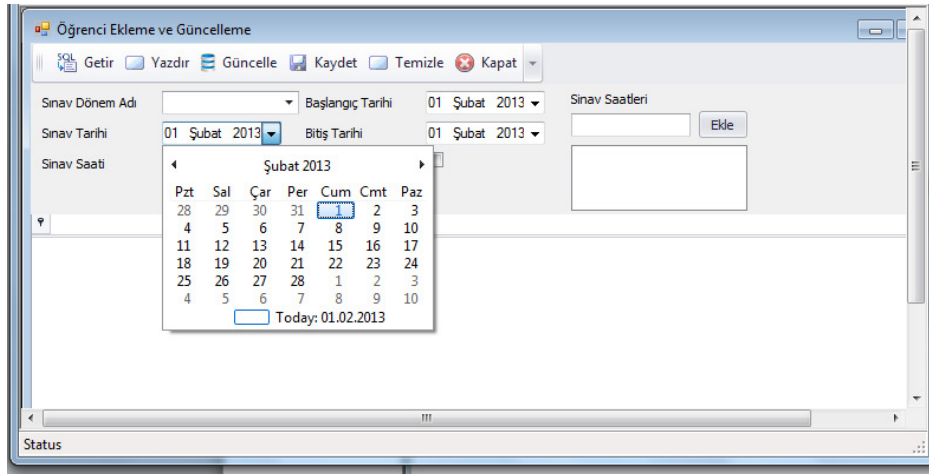
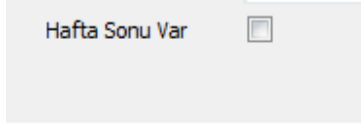
The screenshot shows a web application interface for managing students. It includes a menu bar with options like 'Getir', 'Yazdır', 'Güncelle', 'Kaydet', 'Temizle', and 'Kapat'. Below the menu, there are input fields for 'Öğrenci Numarası', 'Ad', 'Soyadı', 'Yönetim', 'Sınıf', and 'Gano'. The main area displays a table of student records with columns: Öğrenci No, Ad, Soyad, Yönetmelik, Gano, and Sınıf. The first few rows are:

Öğrenci No	Ad	Soyad	Yönetmelik	Gano	Sınıf
03 04 03 031	GÖKHAN	KARADAYI	Y1	16,67	4
03 04 03 039	UTKU	YILMAZ	Y1	54,04	4
03 04 03 040	MUSTAFA NECATİ	TUĞRUL	Y1	2,44	4

Şekil 4.6 Öğrenciler listesi

#### 4.6.3.2 Sınav Tarihleri Belirleme ve Listeleme

- Sınav Dönemi Adı girişi yapılır.
- Sınav tarihleri takvimden seçilir.
- Sınav tarihleri başlangıç ve bitiş tarihleri de takvimden seçilebilir.
- Sınav Saatleri girilir.
- **Güncelle** butonu ile veritabanına kaydedilir.
- Haftasonu da sınav yapılacaksa, Hafta Sonu Var butonu seçilerek sınav saatleri oluşturulur.



Şekil 4.7 Sınav saati giriş ve güncelleme ekranı

row_status	SINAV_TARIH	SINAV_SAAAT	DONEM_AD
1	20111031	09:00	2011-EXAM1
1	20111031	11:00	2011-EXAM1
1	20111031	14:00	2011-EXAM1
1	20111101	10:00	2011-EXAM1
1	20111101	14:00	2011-EXAM1
1	20111102	09:00	2011-EXAM1
1	20111102	13:00	2011-EXAM1
1	20111102	16:00	2011-EXAM1
1	20111103	10:00	2011-EXAM1
1	20111103	15:00	2011-EXAM1
1	20111104	09:00	2011-EXAM1
1	20111104	14:00	2011-EXAM1
1	20111104	16:00	2011-EXAM1

Şekil 4.8 Database \_SINAV\_TARİHLERİ tablosu

Daha önce girişi yapılmış sınav tarih ve saatleri listelenebilmektedir. Güncelleme yapılabilir.

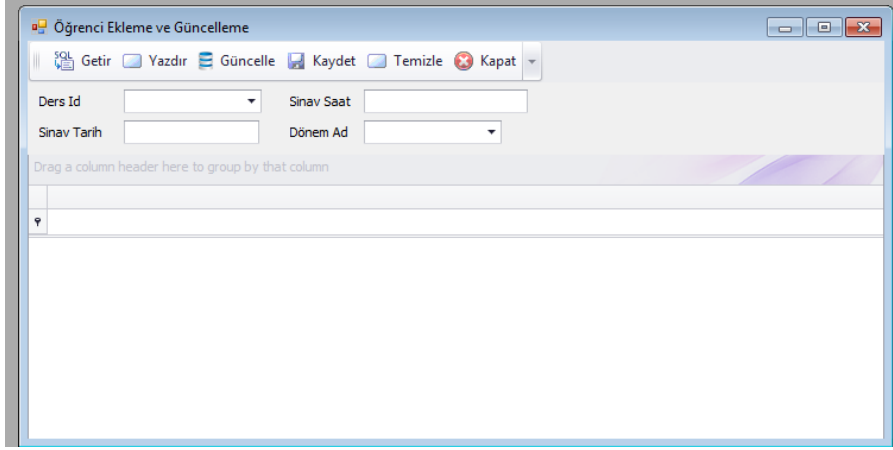
Sınav Tarihi	Sınav Saati	Dönem Adı
31.10.2011	09:00	2011-EXAM1
31.10.2011	11:00	2011-EXAM1
31.10.2011	14:00	2011-EXAM1
01.11.2011	10:00	2011-EXAM1
01.11.2011	14:00	2011-EXAM1
02.11.2011	09:00	2011-EXAM1
02.11.2011	13:00	2011-EXAM1
02.11.2011	16:00	2011-EXAM1
03.11.2011	10:00	2011-EXAM1
03.11.2011	15:00	2011-EXAM1
04.11.2011	09:00	2011-EXAM1
04.11.2011	14:00	2011-EXAM1
04.11.2011	16:00	2011-EXAM1
21.11.2011	09:00	2011-EXAM2
21.11.2011	11:00	2011-EXAM2
21.11.2011	14:00	2011-EXAM2
22.11.2011	10:00	2011-EXAM2

Şekil 4.9 Sınav saatleri listesi

#### 4.6.3.3 Sınav Rezervasyon

Bazı fakültelerde, belirli bir sınav tarihi ve saatinin belirli bir ders için rezerve edilmesi ihtiyacı duyulmaktadır. Bu nedenle bu ekran geliştirilmiştir. DersId'si

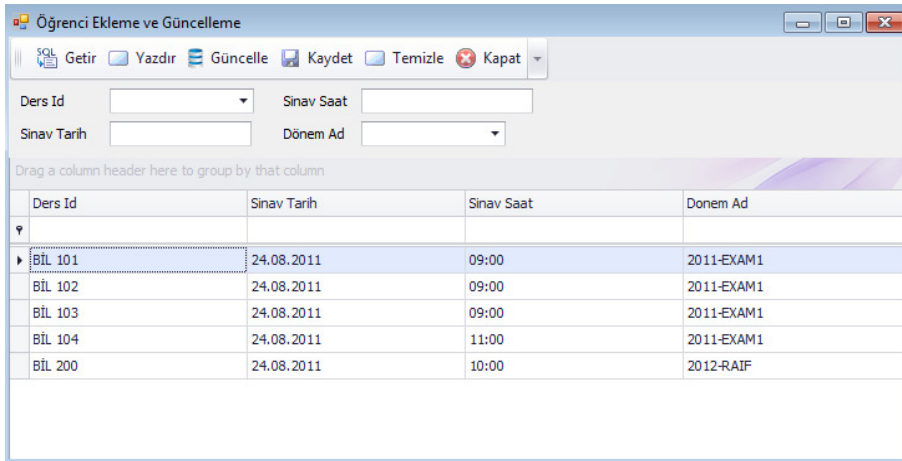
seçilen ders için sınav saati ve tarihi girilerek, dönem adı seçildiğinde bu oturuma bu ders ataması sınav takviminde otomatik olarak yapılmaktadır.



Şekil 4.10 Sınav oturumu rezervasyon ekranı

	row_guid	row_version	row_status	DERS_ID	SINAV_TARİH	SINAV_SAAAT	DONEM_AD
▶	1	20120214010101	1	BİL 101	24.08.2011	09:00	2011-EXAM1
	2	20120214010101	1	BİL 102	24.08.2011	09:00	2011-EXAM1
	3	20120214010101	1	BİL 103	24.08.2011	09:00	2011-EXAM1
	4	20120214010101	1	BİL 104	24.08.2011	10:00	2011-EXAM1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Şekil 4.11 Database SINAV\_REZERVE tablosu



Şekil 4.12 Sınav rezervasyonu listesi

#### 4.6.3.4 Sınav Atama

- Seçtiğimiz algoritmaya göre sınav ataması yapılır.
- Herhangi bir çakışmaya bakılmaksızın ders\_id sırasında ve rastgele oturumlara sınavlar dağıtılır.
- Aynı seanstaki diğer sınavlarla çakışma varsa çizelgelemede ilgili ders kırmızıya boyanır
- Aynı gün içindeki diğer sınavlarla çakışma varsa sarıya boyanır
- Gün içinde veya aynı saatte çakışma yoksa siyah renkle gösterilir
- Derse girecek toplam öğrenci sayısı ve çakışan öğrenci sayısını da çizelgelemeden görebiliriz.
- Sürükle bırak fonksiyonu ile istediğimiz dersi istediğimiz seanstan çıkarıp başka bir seansa dahil edebiliriz. Sürükle bırak yapar yapmaz aynı saat ve gün içinde çakışmalar yeniden hesaplanır.
- Silinmek istenen sınavlar silinerek güncellenir.
- Çift klik ile detay öğrenci listesi görülebilir.
- Detay listede çakışmalar yine ders bazında ve kırmızı/sarıya boyalı olarak görüntülenir.
- İstenilen bir ders\_Id seçilerek filtreleme yapılabilir.
- İstenilen bir öğrenci, adı soyadı veya numarası ile sorgulanabilir. Gireceği sınavlar, varsa çakışmaları da görünecek şekilde listelenebilir.

Bu sınav dönemi için veritabanında 593 öğrencinin, 76 adet ders aldığını görüyoruz. 2011-Exam2 dönemine ait sınav ataması takvimi RSINA algoritması ile çalıştırıldığında 48 dersin çakıştığını gözlemliyoruz.

2011-EXAM2		Getir	Güncelle	Ders Ata	
	21 Pazartesi	22 Salı	23 Çarşamba	24 Perşembe	25 Cuma
08					
09 <sup>00</sup>	BİL 101, BİL 102, BİL 133, BİL 134, BİL 201		BİL 202, BİL 203, BİL 204, BİL 206, BİL 221		BİL 235, BİL 237, CEN 101, CEN 102, CEN 133
10 <sup>00</sup>		CEN 134, CEN 201, CEN 202, CEN 203, CEN 204			
11 <sup>00</sup>	CEN 221, CEN 235, CEN 237, EE 204, EE 206		ELK 101, ELK 102, ELK 201, ELK 202, ELK 203	ELK 204, ELK 205, ELK 206, ELK 208, ELK 211	
12					
13 <sup>00</sup>					
14 <sup>00</sup>	ELK 212, ELK 223, ELK 232, ENİ 101, ENİ 102	ENİ 103, ENİ 201, ENİ 202, ENİ 203, ENİ 204	ENİ 206, ENİ 208, ENİ 210, ENİ 212, ENİ 214	ENİ 216, ENİ 218, ENİ 220, IE 102, İNS 102	İNS 112, İNS 102, İNS 103, İNS 104, YZM 101
15 <sup>00</sup>					
16 <sup>00</sup>		YZM 102, YZM 103, YZM 104, YZM 105, YZM 106			

Şekil 4.13 Sınav atama ekranı 1

Manuel düzeltmelerle sürükleyip bırak işlemi ile 2012-Raif dönemine ait aşağıdaki sınav takvimini oluşturabiliriz.

4 tane derste çakışma olduğunu gözlemledik.

2012-RAIF		Getir	Güncelle	
	1 Pazartesi	2 Salı	3 Çarşamba	4 Perşembe
09	BİL 101, CEN 235, ELK 204, END 210, İNS 102		ELK 205, CEN 237, BİL 102, END 212, İNS 112, YZM 204	YZM 205, İNS 102, END 214, BİL 133, ELK 202
10 <sup>00</sup>		BİL 201, İNS 104, ELK 206, YZM 207, END 218		
11 <sup>00</sup>	YZM 103, YZM 203, END 101, ELK 208, CEN 101		YZM 104, IE 102, END 102, BİL 203, CEN 102, ELK 211, YZM 211	ELK 212, BİL 204, CEN 133, END 103, YZM 101, YZM 251
12				
13 <sup>00</sup>	BİL 202 37/0, ELK 232 72/0, END 220 74/0	YZM 210 37/0		
14 <sup>00</sup>	END 206, YZM 105, BİL 221, CEN 202, EE 204 108/0		EE 206 72/0, CEN 203, BİL 235, END 202, YZM 106	YZM 108, END 203, BİL 237, CEN 204, ELK 101
15 <sup>00</sup>		YZM 202, END 208 37/0, CEN 221 0/0, ELK 201 36/0		
16 <sup>00</sup>				

Şekil 4.14 Sınav atama ekranı 2

Manuel değişikliklere devam edelim ve aşağıdaki takvimi oluşturalım. 9 tane çakışmamızı gözlemleyebiliriz.



2012-RAIF		25	26	27	28	29	30	1
		Cuma	Cumartesi	Pazar	Pazartesi	Salı	Çarşamba	Perşembe
08								
09	BİL 203	YZM 130	YZM 101	END 204	BİL 201	ELK 102	CEN 102	ELK 232
10		YZM 207	END 203	END 202	YZM 105	CEN 133	ELK 101	YZM 102
11		123/8	111/7	111/3	111/3	72/36	206/36	205/36
12	ELK 223	CEN 237	ELK 212	END 103	ELK 211	END 101	CEN 202	CEN 201
13	0/0	0/0	0/0	0/0	0/0	0/0	17/0	37/0
14	INS 104	ELK 202	INS 103	ELK 201	INS 102	EE 204	INS 102	YZM 202
15	0/0	36/0	0/0	36/0	0/0	72/0	0/0	74/0
16	YZM 251	END 218	BİL 102	END 214	BİL 237	END 212	BİL 235	END 210
17	0/0	37/0	74/0	74/0	38/0	74/0	0/0	37/0
18			BİL 101					
19			37/0					
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								
37								
38								
39								
40								
41								
42								
43								
44								
45								
46								
47								
48								
49								
50								
51								
52								
53								
54								
55								
56								
57								
58								
59								
60								
61								
62								
63								
64								
65								
66								
67								
68								
69								
70								
71								
72								
73								
74								
75								
76								
77								
78								
79								
80								
81								
82								
83								
84								
85								
86								
87								
88								
89								
90								
91								
92								
93								
94								
95								
96								
97								
98								
99								
100								

Şekil 4.15 Sınav atama ekranı 3

Sınav atama takviminde yer alan herhangi bir dersin detayına girebiliriz. Öğrenci bazında sorguladığımızda, bu öğrencinin aldığı dersler, sınav takvimindeki sınav tarih ve saatleri ile aynı seans ve gün içi çakışmalarını da gözlemlememiz mümkündür.

Kırmızı ile boyanan derslerin sınav tarih ve saatleri çakışmaktadır. Sarı ile boyalı derslerin aynı gün içerisinde sınavı vardır.

Yine, öğrencinin aldığı derslerin herbirinin ders\_Id'si ile birlikte sınav takvimindeki tarih ve saat bilgisini de görebiliriz.

Ders Adı			
END 102 - Teknik Çözüm			
Gün içinde eşleşen	134	Gün	20110803
Aynı saatte içinde eşleşen	60	Saat	11:00
Yazdır			
Ders Id			
Sınav Zamanı	Öğrenci No	Öğrenci Adı	Öğrenci Soyadı
▼			
Ders Id: END 101	08 07 02 041	TAYLAN	ÇOBAN
01.08.2011 11:00			
1			
Ders Id: END 102	08 07 02 041	TAYLAN	ÇOBAN
03.08.2011 11:00			
1			
Ders Id: END 103	08 07 02 041	TAYLAN	ÇOBAN
04.08.2011 11:00			
1			
Ders Id: END 201	08 07 02 041	TAYLAN	ÇOBAN
05.08.2011 11:00			
1			
Ders Id: END 202	08 07 02 041	TAYLAN	ÇOBAN
03.08.2011 14:00			
1			
Ders Id: END 203	08 07 02 041	TAYLAN	ÇOBAN
04.08.2011 14:00			
1			
Ders Id: END 204	08 07 02 041	TAYLAN	ÇOBAN
05.08.2011 14:00			
16			
x [Öğrenci No] Like '08 07 02 041%'			

Şekil 4.16 Sınav takvimi öğrenci sorgulama

Sadece ders\_ID bazında da sorgulama yaparak yine çakışması olan derlerin altında öğrenci bilgilerine de ulaşmak mümkündür.

Ders Adı			
INS 102 - Teknik Çözüm			
Gün içinde eşleşen	0	Gün	20110801
Aynı saatte içinde eşleşen	0	Saat	09:00
Yazdır			
Ders Id			
Sınav Zamanı	Öğrenci No	Öğrenci Adı	Öğrenci Soyadı
▼			
03.08.2011 11:00	10 07 02 005	GÖRKEM	GÜNDÜZ
03.08.2011 11:00	10 07 02 006	DUYGU	YILMAZ
03.08.2011 11:00	10 07 02 008	FATMA GÖKSU	DEMİRÇEVİREN
03.08.2011 11:00	10 07 02 010	EMİR PEKCAN	PEKCAN
60			
Ders Id: INS 102	10 07 08 002	YILMAZ	AZIZOĞLU
01.08.2011 09:00	10 07 08 005	UĞURCAN	BULAK
01.08.2011 09:00	10 07 08 024	OĞUZHAN	ÇELİK
01.08.2011 09:00	10 07 08 031	SELİN	GÜNDOĞDU
01.08.2011 09:00	10 07 08 032	BARIŞ	GEZEN
5			
Ders Id: INS 112	10 07 08 002	YILMAZ	AZIZOĞLU
03.08.2011 09:00	10 07 08 005	UĞURCAN	BULAK
03.08.2011 09:00	10 07 08 024	OĞUZHAN	ÇELİK
03.08.2011 09:00	10 07 08 031	SELİN	GÜNDOĞDU
03.08.2011 09:00	10 07 08 032	BARIŞ	GEZEN
5			
Ders Id: İNŞ 102	10 07 08 002	YILMAZ	AZIZOĞLU
04.08.2011 09:00	10 07 08 005	UĞURCAN	BULAK
04.08.2011 09:00	10 07 08 024	OĞUZHAN	ÇELİK
2807			

Şekil 4.17 Derslerin sınav takviminde ders\_id ve öğrenci detayları

Ders Id	Oğrendi No	Oğrendi Adı	Oğrendi Soyadı
01.06.BİL 102	08 07 01 023	EMİN	KAYA
01.06.BİL 133	08 07 01 024	DENİZ	DEMİREL
01.06.BİL 134	08 07 01 025	SERAP	AKCAN
01.06.BİL 201	08 07 01 033	GÖRKEM KAAN	DURUK
01.06.BİL 202	08 07 01 035	BİLAL	KAPLAN
01.06.BİL 203	08 07 01 037	GÖZDE	MUCUK
01.06.BİL 204	08 07 01 043	SİNAN	DÖNER
01.06.BİL 206	08 07 01 045	EKREM CAN	YARDAR
01.06.BİL 221	08 07 01 046	BAHAR	BÜLBÜLÖĞÜ
01.06.BİL 235	08 07 01 052	BURAK	OKULDAŞ
01.06.BİL 237	08 07 01 054	MESUT	ELYAK
01.06.CEN 101	08 07 01 055	MUSTAFA	ÇİFCİ
01.06.CEN 102	08 07 01 056	ORHUN	BALOĞLU
01.06.CEN 103	08 07 01 060	EMRE	ARAS
01.06.CEN 104	08 07 01 061	HAMİDE	ÇELEN
01.06.CEN 105	08 07 01 062	OKAN	OZMAN
01.06.CEN 106	08 07 01 063	BÜŞRA	TANMAZ
01.06.CEN 107	08 07 01 065	HACI ALI	YURTTAŞ
01.06.CEN 108	09 07 01 002	ÖZGE HİLAL	TEKDAL
01.06.CEN 109	09 07 01 004	HAKAN	ÇORLUH
01.06.CEN 110	09 07 01 005	CAN	UZUNKAYA
01.06.CEN 111	09 07 01 006	SILA	KIR
01.06.CEN 112	09 07 01 008	FAHRE EVREN	KAYA
01.06.CEN 113	09 07 01 009	SEMA	ARSLAN

Şekil 4.18 Derslerin sınav takviminde ders\_id ve öğrenci detayları

#### 4.6.3.5 Ders Atama fonksiyonu, SINA1 ve SINA2 algoritmalarının çalışması

Bu sınav dönemi için veritabanında 593 öğrencinin, 76 adet ders aldığı, 2011-Exam2 dönemine ait sınav takviminde RSINA ile 48 dersin çıktığını gözlemlemiştik. Bu takvimdeki derslerin çakışma matrislerini bulalım.

DERSLER	BİL 101	BİL 102	BİL 133	BİL 134	BİL 201	BİL 202	BİL 203	BİL 204
BİL 101	58	58	58	58	37	37	37	37
BİL 102	58	58	58	58	37	37	37	37
BİL 133	58	58	58	58	37	37	37	37
BİL 134	58	58	58	58	37	37	37	37
BİL 201	37	37	37	37	38	38	38	38
BİL 202	37	37	37	37	38	38	38	38
BİL 203	37	37	37	37	38	38	38	38
BİL 204	37	37	37	37	38	38	38	38
BİL 206	37	37	37	37	38	38	38	38
BİL 221	37	37	37	37	38	38	38	38
BİL 235	37	37	37	37	38	38	38	38

Şekil 4.19 Derslerin çakışma matrisi



2011-EXAM2		Getir			Güncelle			Ders Ata							
	21 Pazartesi	22 Salı	23 Çarşamba	24 Perşembe	25 Cuma										
09	YZM 102	ELK 204	IE 102 111/0	END 103	ELK 203	YZM 101	CEN 237	END 102	ELK 202	CEN 235	END 101	ELK 201	CEN 221	EE 206 0/0	END 220
10 <sup>00</sup>	EE 204 0/0	CEN 204	END 218	BİL 134	CEN 203	END 216	BİL 133	CEN 202	END 214	BİL 102	CEN 201	END 212	BİL 101	END 210	İNŞ 104
11 <sup>00</sup>	END 208	BİL 237	İNŞ 103	END 206	BİL 235	İNŞ 102	END 204	BİL 221	İNŞ 112	END 203	BİL 206	İNŞ 102	END 202	BİL 104	37/0
12															
13 <sup>00</sup>															
14 <sup>00</sup>	END 201 0/0	BİL 203 0/0	BİL 202 37/0	ELK 102 0/0	BİL 201 37/0	ELK 101 0/0	ELK 232 0/0	CEN 134 17/0	ELK 223 36/0	CEN 133 17/0					
15 <sup>00</sup>															
16 <sup>00</sup>	ELK 212 36/0	CEN 102 17/0	YZM 106	ELK 211	CEN 101	YZM 105 0/0	ELK 208 72/0	YZM 104 0/0	ELK 206 72/0	YZM 103 0/0	ELK 205 72/0				
17 <sup>00</sup>															

Şekil 4.22 SINA1 ile yeni sınav takvimi

2012 Raif dönemine ait sınav takvimini, sürükle-bırak yordamıyla değiştirerek veritabanında \_SINAV\_ATAMA tablosunda update ettik.

2012-RAIF		Getir			Güncelle			Ders Ata																		
	25 Cuma	26	27 Pazar	28 Pazartesi	29 Salı	30 Çarşamba	1 Perşembe																			
09	BİL 203	END 204	YZM 130	BİL 202	EN 202	YZ 106	YZ 206	BİL 201	EN 202	YZ 106	EN 206	CE 134	YZ 102	EN 201	YZ 204	ELK 102	CEN 133	YZM 104	CE 102	ELI 101	YZ 102	ELI 202	ELK 232	YZM 102	CEN 101	
10 <sup>00</sup>																										
11 <sup>00</sup>	ELI 222	YZ 103	CE 237	YZ 207	ELK 212	IE 102	CEN 235	END 103	ELK 211	CEN 221	ELK 208	END 102	CEN 204	END 101	ELK 206	CEN 203	CEN 202	YZM 211	CEN 201	ELK 204	YZM 210					
12	ELK 203	İNŞ 104	ELK 202	İNŞ 103	ELK 201	İNŞ 102	YZM 205	EE 206 36/0	INS 112																	
13 <sup>00</sup>																										
14 <sup>00</sup>	YZM 251	END 218	BİL 102	END 216 74/0	END 214	BİL 237	END 212	BİL 235																		
15 <sup>00</sup>																										
16 <sup>00</sup>																										
17 <sup>00</sup>																										

Şekil 4.23 2012-Raif manuel düzenlenmiş takvim

```

SELECT row_guid, row_version, row_status, DERS_ID, SINAV_TARİH, SINAV_SAAAT, DONEM, SINAV_SAAAT_BITIS
FROM SE_CMN.SINAV_ATAMA
WHERE (DONEM = '2012-RAIF')

```

row_guid	row_version	row_status	DERS_ID	SINAV_TARİH	SINAV_SAAAT	DONEM	SINAV_SAAAT_BITIS
2	2012030218...	1	BİL 101	20111201	09:00	2012-RAIF	09:59
4	2012030218...	1	BİL 102	20111201	09:00	2012-RAIF	09:59
7	2012030218...	1	BİL 133	20111201	09:00	2012-RAIF	09:59
10	2012030218...	1	BİL 134	20111201	09:00	2012-RAIF	09:59
13	2012030218...	1	BİL 201	20111201	09:00	2012-RAIF	09:59
18	2012030218...	1	BİL 202	20111225	09:00	2012-RAIF	09:59
21	2012030218...	1	BİL 203	20111225	09:00	2012-RAIF	09:59
24	2012030218...	1	BİL 204	20111225	09:00	2012-RAIF	09:59
27	2012030218...	1	BİL 206	20111226	09:00	2012-RAIF	09:59
30	2012030218...	1	BİL 221	20111225	09:00	2012-RAIF	09:59
33	2012030218...	1	BİL 235	20111226	09:00	2012-RAIF	09:59
36	2012030218...	1	BİL 237	20111226	09:00	2012-RAIF	09:59
39	2012030218...	1	CEN 101	20111226	09:00	2012-RAIF	09:59
42	2012030218...	1	CEN 102	20111226	09:00	2012-RAIF	09:59
45	2012030218...	1	CEN 133	20111226	09:00	2012-RAIF	09:59
48	2012030218...	1	CEN 134	20111227	09:00	2012-RAIF	09:59
51	2012030218...	1	CEN 201	20111227	09:00	2012-RAIF	09:59
54	2012030218...	1	CEN 202	20111227	09:00	2012-RAIF	09:59
57	2012030218...	1	CEN 203	20111227	09:00	2012-RAIF	09:59
60	2012030218...	1	CEN 204	20111227	09:00	2012-RAIF	09:59
63	2012030218...	1	CEN 221	20111228	09:00	2012-RAIF	09:59
66	2012030218...	1	CEN 235	20111228	09:00	2012-RAIF	09:59
69	2012030218...	1	CEN 237	20111228	09:00	2012-RAIF	09:59

Şekil 4.24 SINAV\_ATAMA tablosu son hali

Ders Ata butonu ile çakışma matrisini bulup SINA2 algoritmasını çalıştırarak çakışmaları küçükten büyüğe sort ederek sınav tarihlerini atadık. Aşağıdaki yeni sınav takvimini oluşturduk. Görüldüğü üzere, 5 derste çakışma varken bu çakışma 1'e düşmüştür. Güncelle butonu ile veritabanına son sınav çizelgemizi update edebiliriz.

	25 Cuma	26	27 Pazar	28 Pazartesi	29 Salı	30 Çarşamba	1 Perşembe
08							
09 00	Yzm 104, Elk 206, Ins 102	Yzm 105, Elk 208, Ins 112	Yzm 106, Elk 211, Ins 102	Yzm 108, Elk 212, Ins 103	Yzm 130, Elk 223, Ins 104	Elk 232, Cen 201	Elk 101, Cen 202
10 00							
11 00	Elk 102, Cen 203	End 201, Cen 204	End 202, Cen 221	End 203, Cen 235	End 204, Cen 237	End 206, Cen 101	Cen 102, End 208
12 00	Cen 133, End 210	End 134, Cen 212	Bil 201, End 214	Bil 202, End 216	Bil 203, End 218	End 220, Bil 204	End 202, Bil 206
13 00							
14 00	Yzm 251, Yzm 203, Bil 221	Yzm 204, Bil 235	Yzm 205, Bil 237	Yzm 206, Bil 101	Yzm 207, Bil 102	Yzm 210, Bil 133	Yzm 211, Bil 134
15 00							
16 00	End 101, Ee 204, 72/0	End 102, Ee 206, 36/0	End 103, Elk 201	Ie 102, Elk 202, 37/0	Yzm 101, Elk 203	Yzm 102, Elk 204	Elk 205, Yzm 103
17 00							

Şekil 4.25 SINA2 algoritmasından sonraki sınav çizelgesi

2011-Exam2 dönemi için RSINA ile oluşturduğumuz sınav çizelgelemesinde 48 adet çakışma gözlemlenmiştir. Şimdi bu takvimi SINA2 ile tekrar oluşturalım.

Şekil 4.26 2011-Exam2 sınav tablosu

Ders Ata butonu yardımıyla çakışma matrisini hesaplıyoruz.

DERSLER	BİL 101	BİL 102	BİL 133	BİL 134	BİL 201	BİL 202	BİL 203
BİL 101	58	58	58	58	37	37	37
BİL 102	58	58	58	58	37	37	37
BİL 133	58	58	58	58	37	37	37
BİL 134	58	58	58	58	37	37	37
BİL 201	37	37	37	37	38	38	38
BİL 202	37	37	37	37	38	38	38
BİL 203	37	37	37	37	38	38	38
BİL 204	37	37	37	37	38	38	38
BİL 206	37	37	37	37	38	38	38
BİL 221	37	37	37	37	38	38	38

Şekil 4.27 2011-Exam2 için çakışma matrisi

Derslerin akışma matrisinde her satırdaki toplam akışmaları küçükten büyüğe sıralayarak SINA2 algoritması ile ders saatlerini atayarak yeni sınav izelgemizi oluşturalım.

Takvimi güncellediğimizde yeni Sınav izelgeleme takvimimiz oluştu. Ve akışmaların sadece gün içindeki diğer oturumlarla olduğunu gördük. Güncelle butonunu kullanarak veritabanına kaydedebiliriz.

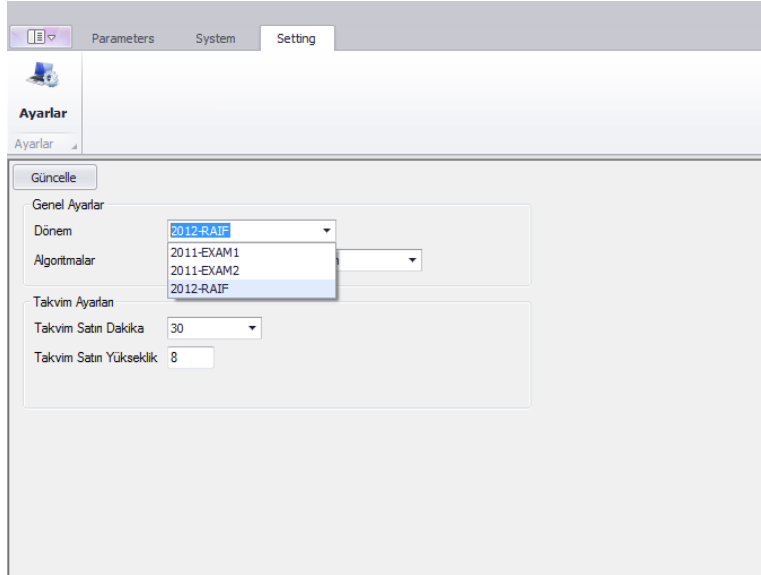
	21 Pazartesi			22 Salı			23 Çarşamba			24 Perşembe			25 Cuma		
09	END 203	BİL 206	İNŞ 102	END 204	BİL 221	İNŞ 112	END 206	BİL 235	İNŞ 102	END 208	BİL 237	İNŞ 103	BİL 101	END 210	İNŞ 104
10 <sup>00</sup>	BİL 102	CEN 201	END 212	BİL 133	CEN 202	END 214	BİL 134	CEN 203	END 216	EE 204	CEN 204	END 218	CEN 221	EE 206	END 220
11 <sup>00</sup>	CEN 235	END 101	ELK 201	CEN 237	END 102	ELK 202	END 103	ELK 203	YZM 101	YZM 102	ELK 204	IE 102	YZM 103	ELK 205	72/0
12															
13 <sup>00</sup>															
14 <sup>00</sup>	YZM 104	ELK 206	72/0	YZM 105	ELK 208	72/0	ELK 211	CEN 101	72/0	ELK 212	CEN 102	36/0	ELK 223	CEN 133	17/0
15 <sup>00</sup>															
16 <sup>00</sup>	ELK 232	CEN 134	17/0	BİL 201	YZM 106	ELK 101	BİL 202	ELK 102	37/0	END 201	BİL 203	0/0	END 202	BİL 204	37/0
17 <sup>00</sup>															
18 <sup>00</sup>															

Şekil 4.28 2011-Exam2 için SINA2 algoritması ile sınav izelgesi



#### 4.6.4 System - Ayarlar

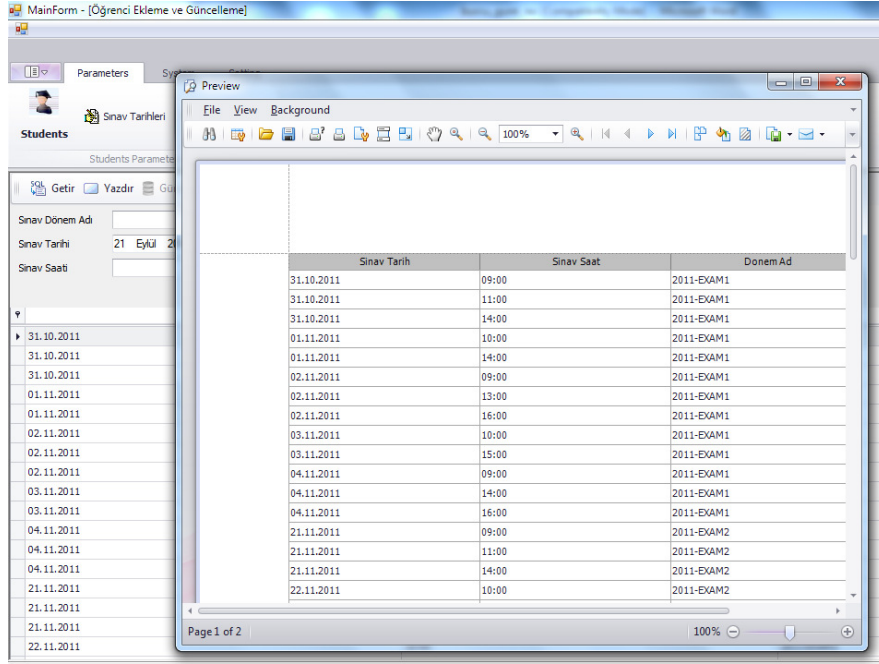
- Ayarlar ekran arayüzü ile istediğimiz Dönem için sistem ayarları belirleyebiliriz.
- Yeni tanımladığımız algoritmaları buradan görüntüleyebilir ve sınav takvimi oluşturulması için buradan algoritma belirleyebiliriz.
- Sınav takvimindeki hücre genişlik ayarlarını değiştirebiliriz.
- Takvim görüntüsündeki saat aralıklarını dakika bazında değiştirebiliriz.



Şekil 4.29 Sistem ayarları ekranı

#### 4.6.5 Raporla / Yazdır

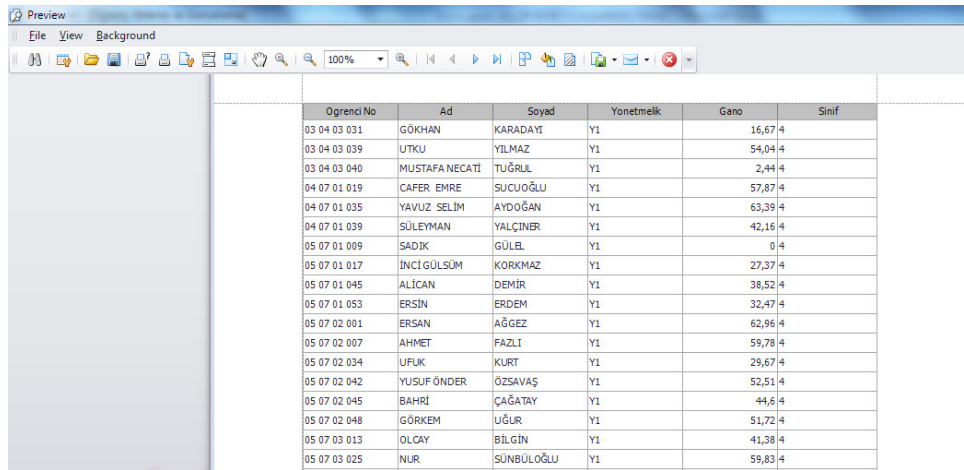
- Tüm ekranlarda raporlama yapabiliyoruz.
- Detay veya özet sorgulama yaparak bunları print edebiliyoruz.



The screenshot shows a software application window titled 'MainForm - [Öğrenci Ekleme ve Güncelleme]'. A 'Preview' window is open, displaying a table of exam dates and times. The table has four columns: 'Sinav Tarihi', 'Sinav Saat', and 'Danem Ad'. The data is as follows:

Sinav Tarihi	Sinav Saat	Danem Ad
31.10.2011	09:00	2011-EXAM1
31.10.2011	11:00	2011-EXAM1
31.10.2011	14:00	2011-EXAM1
01.11.2011	10:00	2011-EXAM1
01.11.2011	14:00	2011-EXAM1
02.11.2011	09:00	2011-EXAM1
02.11.2011	13:00	2011-EXAM1
02.11.2011	16:00	2011-EXAM1
03.11.2011	10:00	2011-EXAM1
03.11.2011	15:00	2011-EXAM1
04.11.2011	09:00	2011-EXAM1
04.11.2011	14:00	2011-EXAM1
04.11.2011	16:00	2011-EXAM1
21.11.2011	09:00	2011-EXAM2
21.11.2011	11:00	2011-EXAM2
21.11.2011	14:00	2011-EXAM2
22.11.2011	10:00	2011-EXAM2

Şekil 4.30 Örnek sınav saatleri detay listesi



The screenshot shows a software application window titled 'Preview'. A table of student information is displayed. The table has seven columns: 'Oğrenci No', 'Ad', 'Soyad', 'Yönetmelik', 'Gano', and 'Sınıf'. The data is as follows:

Oğrenci No	Ad	Soyad	Yönetmelik	Gano	Sınıf
03 04 03 031	GÖKHAN	KARADAYI	Y1	16,67	4
03 04 03 039	UTKU	YILMAZ	Y1	54,04	4
03 04 03 040	MUSTAFA NECATI	TUĞRUL	Y1	2,44	4
04 07 01 019	CAFER EMRE	SUCUOĞLU	Y1	57,87	4
04 07 01 035	YAVUZ SELİM	AYDOĞAN	Y1	63,39	4
04 07 01 039	SÜLEYMAN	YALÇINER	Y1	42,16	4
05 07 01 009	SADIK	GÜLEL	Y1	0	4
05 07 01 017	İNCI GÜLSÜM	KORKMAZ	Y1	27,37	4
05 07 01 045	ALİCAN	DEMİR	Y1	38,52	4
05 07 01 053	ERSİN	ERDEM	Y1	32,47	4
05 07 02 001	ERSAN	AĞGEZ	Y1	62,96	4
05 07 02 007	AHMET	FAZLI	Y1	59,78	4
05 07 02 034	UFUK	KURT	Y1	29,67	4
05 07 02 042	YUSUF ÖNDER	ÖZSAVAŞ	Y1	52,51	4
05 07 02 045	BAHRİ	ÇAĞATAY	Y1	44,6	4
05 07 02 048	GÖRKEM	UĞUR	Y1	51,72	4
05 07 03 013	OLCAY	BİLGİN	Y1	41,38	4
05 07 03 025	NUR	SUNBÜLOĞLU	Y1	59,83	4

Şekil 4.31 Örnek öğrenci bilgileri listesi

## 5 TARTIŞMA VE SONUÇ

Bu çalışmada, sınav çizelgeleme işleminin geliştirilen 3 farklı algoritma ile gerçekleştirilmesi amaçlanmıştır.

Örnek uygulama alanı olarak MÜBİS'deki öğrenci, ders ve öğretim üyeleri bilgileri kullanılmıştır.

Farklı tarihler için sınav dönemleri girişleri yapılarak sonuçları gözlemlenmiştir.

Gerekli kısıtlar sisteme girildikten sonra sunucuda bu bilgiler çakışmaları büyükten küçüğe, küçükten büyüğe hesaplayıp kullanıcı yordamıyla değiştirilecek şekilde hesaplamaktadır. Bu iki algoritma(SINA1, SINA2) sonucunda oluşan sınav çizelgelemesi ile ders sırasında gelişi güzel (RSINA) oluşturduğumuz sınav çizelgelemesindeki çakışmaları karşılaştırılmıştır.

Çakışma matrisi oluşturarak sıralama algoritmalarını kullandığımız sınav çizelgelemelerindeki çakışmaların optimize edilebildiği gözlemlenmiştir.

Yazılım, yeni algoritma tanımlanmasına elverişli olup, parametrik olarak da algoritma seçimine izin verdiği için kullanıcı dostu arayüz ekranlarına sahiptir diyebiliriz.

Bu çalışma, üniversitenin tüm fakültelerinde, MÜBİS veritabanına entegre olacak niteliktedir. Derslik ve öğretim görevlileri atamaları için yazılım geliştirilebilecek veritabanı özelliklerine sahiptir.

Problemin doğrusal modelindeki inputları şu şekilde ifade edebiliriz:

- Sınav tarih ve saatleri

$$O \left( \begin{array}{c} \#days * \# hours / day \end{array} \right)$$

- Öğrencilerin aldığı dersler  
i : ders\_id, i = ( 1, n )

n: toplam ders sayısı

$m_i$  : i dersini alan öğrenci sayısı

Tüm derslerdeki toplam öğrenci sayısı :

$$\sum_{i=1}^n m_i$$

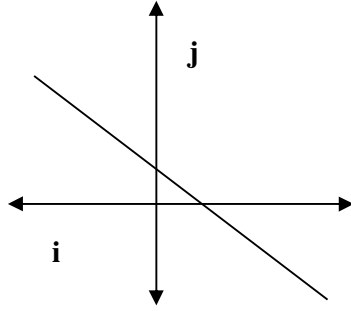
$$O \left( n * m_{max} \right)$$

- Çakışma matrisi

Çakışma matrisinin maximum büyüklüğü n x n kadardır.

i : herbir satırdaki dersi ifade eder. i = 1, n

j : herbir kolondaki dersi ifade eder. j = 1, n



Çakışma matrisi, simetriktir.  
Bu nedenle, formülasyonda  
çıkan sonucun yarısını  
almamız gerekir.

$$O \left( \frac{\sum_{i=1}^n \sum_{j=1}^n m_i m_j}{2} \right)$$

Bir sınav oturumu için en fazla 5 derse ait sınav ataması yapılır  
( parametrik olarak bu sayıyı azaltıp çoğaltmamız mümkündür).

Bir sınav tarihi ve saati için her yeni ders ataması yapıldığında diğer  
aynı oturuma atadığımız derslerle çakışmalarına bakarız.

Bu döngü, 1 oturuma 5 ders sınavı atadığımızda 10 kez yapılmış olur.  
Maximumu 10 x sınav oturumu kadardır.

Bu performansı şöyle ifade edebiliriz.

a: Bir sınav oturumunda verilebilecek max. ders sayısı

$$\sum_{i=1}^a a_i = a(a + 1) / 2$$

## KAYNAKLAR

- [1] Abdullah, S., 2006, Heuristic Approaches for University Timetabling Problems, Doctor of Philosophy Thesis, The School of Computer Science and Information Technology, The University of Nottingham, Nottingham,UK.
- [2] Abramson, D., January 1991. Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms, *Management Science*, 37(1):98-113.
- [3] Alaykiran, K., Engin, O., 2005. Karınca Kolonileri Metasezgiseli ve Gezgin Satıcı Problemleri Üzerinde Bir Uygulaması. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, Sayı: 20, 69-76.
- [4] Altıntaş, C. Sezgisel Algoritmalarla Sınav Çizelgeleme Problemi Çözümü, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi
- [5] April, J.; Glover, F.; Kelly, J.; Laguna, M.: *Simulation/Optimization Using Real-World Applications* , *Proceedings of the 2001 Winter Simulation Conference*, B.A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds., (2001) 134.
- [6] Birogul, S. , 2005, Genetik Algoritma Yaklaşımıyla Atölye Çizelgeleme, Yüksek Lisans, Gazi Üniversitesi , Fen Bilimleri Enstitüsü, Ankara.
- [7] Brailsford, S.C., Potts, C.N. ve Smith, B.M. , 1999, “Constraint satisfaction problems: Algorithms and applications”, *European Journal of Operational Research*, Vol.119, o.3,pp.557-581.
- [8] Burke E., Hart E., Kendall G., Newall J., Ross P. and Schulenburg S., 2003b, “Hyper-Heuristics: An Emerging Direction in Modern Search Technology”, *Handbook of Meta-Heuristics* (Glover, F., ed), pp 457 – 474, Kluwer.
- [9] Burke, E.K., Silva, D.L. , 2004, “The Design of Memetic Algorithms for Scheduling and Timetabling Problems”, *Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing*, Krasnogor, N., Hart, W. ve Smith, J., Springer, pp.289-312.
- [10] Cheng, R., Gen, M., Tsujimura, Y., 1999. ”A Tutorial Survey of Job Shop Scheduling Problems Using Genetic Algorithms: Part II. Hybrid Genetic Search Strategies”, *Computers and Industrial Engineering*, 37, 51-55.
- [11] Cladeira, JP and Rosa, AC, 1997. School Timetabling using Genetic Search, *PATAT 97*, pp. 115-122.
- [12] Colomi, A. and Dorigo, M. and Maniezzo, V., 1992. Genetic algorithm to solve the timetable problem. Tech. rep. 90-060 revised, Politecnico di Milano, Italy
- [13] David, P. , 1998, “A constraint-based approach for examination timetabling using local repair techniques”, In: Burke, E.K. ve Carter, M.W. (eds). (1998). *Practice and Theory of Automated Timetabling: Selected Papers from the 2nd International Conference*. Springer Lecture Notes in Computer Science, vol.408,pp.169-186.
- [14] D.Corne, P. Ross, H.L. Fang, *Evolutionary Timetabling: Practice, Prospects and Work in Progress*, Proc. of the UK Planning and Scheduling SIG Workshop, 1994.
- [15] Di Gaspero, L. ve Schaerf, A., 2001, “Tabu search techniques for examination timetabling”, In: Burke, E.K. ve Erben, W. (eds). (2001). *Practice and Theory of Automated Timetabling: Selected Papers from the 3rd International Conference*. Springer Lecture Notes in Computer Science, vol.2079, pp.104-117.

- [16] Dignum, F. P. M. and Nuijten, W.P.M. and Janssen, L.M.A., 1995. Solving a Time Tabling Problem by Constraint Satisfaction, Tech. report, Eindhoven University of Technology.)
- [17] Duong, T.A., Lam, K.H. , 2004, “Combining constraint programming and simulated annealing on university exam timetabling”, In: Proceedings of the 2nd International Conference in Computer Sciences, Research, Innovation & Vision for the Future (RIVF2004), Hanoi, Vietnam, February 2-5, 205-210.
- [18] Engin, O. (2001). Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Artırılmasında Parametre Optimizasyonu,Doktora tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [19] Erben, W. and Keppler, J., 1995. A Genetic Algorithm Solving a Weekly Course-Timetabling Problem, Proc. of the First Int. Conf. on the Practice and Theory of Automated Timetabling (ICPTAT), pp. 21-32, Napier University, Edinburgh.
- [20] Even, S., Itai, A., Shamir, A., 1976. “On the Complexity of Timetable and Multicommodity Flow Problems”, SIAM Journal of Computing, 691-703.
- [21] Goldberg, D.E., 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, Boston, MA, USA ,Addison-Wesley Longman Publishing Co., Inc.
- [22] Güldalı,A., Seri İş Ak ışlı Atölye Çizelgelmesinde Sezgisel Teknikler,Yüksek Lisans Tezi, G.Ü. Fen Bilimleri Enstitüsü, Ankara, 1990.
- [23] Hertz, A., 1992. Finding a feasible course schedule using a tabu search, Discrete Applied Mathematics, 35, 255-270.
- [24] Holland, J.H., 1975, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI. 188
- [25] Kalender, M., 2007. Ders Çizelgeleme Programı. TMMOB Elektrik Mühendisleri Odası İstanbul Şubesi, 2006-2007 Öğretim Yılı Proje Yarışması, İstanbul.
- [26] Karaboga, D., 2004, Yapay Zeka Optimizasyon Algoritmaları, Atlas Yayın Dağıtım.
- [27] Kendall, G. ve Hussin, N.M., 2004, “A tabu search hyper-heuristic approach to the examination timetabling problem at the MARA university of technology”,In: Burke, E.K. ve Trick, M. (eds), (2005),Selected Papers from the 5th International Conference on the Practice and Theory of Automated Timetabling, Springer Lecture Notes in Computer Science, vol.3616, pp.199-218.
- [28] Koza, J.R., 1993, Genetic Programming: On the Programming of Computers by Means of Natural Selection, Massachusetts, MIT Press, Cambridge.,
- [29] Merlot, L., Boland, N., Hughes,B. ve Stuckey,P.J. , 2003, “A hybrid algorithm for the xamination timetabling problem”, In: Burke, E, Causmaecker, P.D.(Eds.): Practice and Theory of Automated Timetabling IV (PATAT 2002, Gent Belgium, August, selected papers). Lecture Notes in Computer Science, Vol.2740, pp.207-231, Springer-Verlag.
- [30] Monfroglio, A., 1988. Timetabling Through a Deductive Database: A Case Study, Data & Knowledge Engineering, 3:1-27.
- [31] Murata, T., Ishibuchi, H., Tanaka, H., 1996. “Genetic Algorithms for Flow Shop Scheduling Problems”, Computers and Industrial Engineering, Vol.30, No.4, 1061-1071.

- [32] Naji Azimi. (2004). Comparison of metaheuristic algorithms for examination timetabling problem. *Applied Mathematics and Computation*, 16(1-2): 337-354.
- [33] Norberciak, M. , 2007, "Universal Method for Timetable Construction based on Evolutionary Approach", *International Journal of Applied Mathematics and Computer Sciences*, Vol. 3, No.3, pp.155-160.
- [34] Özcan, E., Alpay, A., 2002. Çok Nüfuslu Kararlı Genetik Algoritması Kullanarak Otomatik Çizelgeleme, *TBD 19. Bilişim Kurultayı*, Sayfa 149-155.
- [35] Özel S, 1999, "Üretim çizelgeleme algoritmasının programlanması", *Uludağ Üniversitesi Fen bilimleri Enstitüsü, Bursa*, 5-9, 55-56.
- [36] Özkazanç, Ümit Ali 1999, Atölye tipi Üretim Ortamında İşlerin Çizelgenmesi için Yapay Sinirsel Ağ Yaklaşımı, *Y.Lisans Tezi, Osmangazi Üniv. Eskişehir*
- [37] Palamutçuoğlu, Bedrettin Türker "Üretim ve Hizmet Planlamasında Çizelgeleme Problemlerinin Yöneylem Teknikleriyle Çözümü", *Yüksek Lisans, Celal Bayar Üniversitesi, Manisa*, 39-53.
- [38] Paquete, L. and Stützle, T. , 2002, An Experimental Investigation of Iterated Local Search for Coloring Graphs. In *Proceedings of the Applications of Evolutionary Computing on Evoworkshops 2002: Evocop, Evoiasp, EvoSTIM/EvoPLAN (April 03 - 04, 2002)*. S. Cagnoni, J. Gottlieb, E. Hart, M.Middendorf, and G. R. Raidl, Eds. *Lecture Notes In Computer Science*, vol. 2279, 122-131. Springer-Verlag, London.
- [39] Qu, R., Burke, E.K., McCollum, B., Merlot, L.T. ve Lee, S.Y., 2006, A Survey of Search Methodologies and Automated Approach for Examination Timetabling (Technical Report NOTTCS-TR-2006-4), *School of Computer Science and Informatin Technology, University of Nottingham*.
- [40] Reis, L.P. and Oliveira, E., 1999, "Constraint logic programming using set variables for solving timetabling problems", *INAP'99, 12th International Conference on Applications of Prolog, Tokyo, Japan*.
- [41] Schaerf, A., 1995, A Survey of Automated Timetabling (Technical Report CSR9567), *Armsterdam, Centrum voor Wiskunde en Informatika-CWI, The Netherlands*.192
- [42] Schaerf, A., 1996. Tabu Search Techniques for Large High-School Timetabling Problems, *Proc. of the Fourteenth National Conference on AI*, pp. 363-368, August.
- [43] Schmidt, G. and Strohlein, T., 1979. Time table construction-an annotated bibliography, *The Computer Journal*, 23(4):307-316.
- [44] Sheibani, K., 2002, "An evolutionary approach for the examination timetabling problems", In: Burke, E.K. ve De Causmaecker, P. (eds). (2002). *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling, 21st-23rd August 2002*, pp.387-396. KaHo St.-Lieven, Gent, Belgium.
- [45] Thompson, J. ve Dowsland, K. , 1998, "A robust simulated annealing based examination timetabling system", *Computers & Operations Research*, Vol.25, No.7-8, pp.637-648.
- [46] White, G.M. ve Xie, B.S. , 2001, "Examination timetables and tabu search with longer-term memory", In: Burke, E.K. ve Erben, W. (eds).*Practice and Theory*



- of Automated Timetabling: Selected Papers from the 3rd International Conference. Springer Lecture Notes in Computer Science, Vol. 2079, pp.85-103.
- [47] Wong, T., Cote, P. ve Gely, P. , 2002, "Final exam timetabling: a practical approach", IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2002). Vol.2, pp.726- 731.
- [48] Wren, A. , 1996, "Scheduling, timetabling and rostering - A special relationship?", In: Burke, E.K. ve Ross, P. (eds), (1996), Practice and Theory of Automated Timetabling: Selected Papers from the 1st International Conference, LNCS 1153. Springer-Verlag, Berlin, Heidelberg. pp.46-75.194
- [49] Yang, Y. ve Petrovic, S., 2004, "A Novel similarity measure for heuristic selection in examination timetabling", In: Burke, E.K. ve Trick, M. (eds). (2005). Selected Papers from the 5th International Conference on the Practice and Theory of Automated Timetabling, Springer Lecture Notes in Computer Science, vol.3616, pp.377-396.
- [50] Youssef, H.; Sait, S. M.; Adiche, H.: "Evolutionary Algorithms, Simulated Annealing and Tabu Search: A comparative Study", Engineering Applications of Artificial Intelligence, 14 (2001) 167.

## ÖZGEÇMİŞ

Adı Soyadı : Ülkü Burcu Gürel  
Doğum Yeri ve Yılı : Ankara - 1976  
Medeni Hali : Evli ve Boğaç'ın annesidir.  
Yabancı Dili : İngilizce  
Ortaokul ve Lise : Beşiktaş Kız Lisesi  
İlkokul : Özel Tarhan Koleji  
Önlisans : İstanbul Üniversitesi Bilgisayar Programcılığı MYO  
Lisans : Anadolu Üniversitesi İşletme Fakültesi - İşletme  
İş Tecrübeleri : Anadolu Kredi Kartı Turizm ve Tic.A.Ş. , Provus Bilişim Teknolojileri, Akbank (GMG Bilgi Teknolojileri) , İş Bankası (Softtech) , Yapı ve Kredi Bankası (Mirsis), Türk Ekonomi Bankası kurumlarının, Banka ve Kredi Kartı Kartlı Ödeme Sistemleri IT birimlerinde, 15 yıllık yazılım, teknik ve business analiz, proje yönetimi çalışmaları olmuştur. Halen, HSBC Bankasında ( Lynx ) çalışmaya devam etmektedir.