



T.C.
MALTEPE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**DERS ÇİZELGESİ PROBLEMİ İÇİN BİR BİLGİSAYAR YAZILIM
SİSTEMİ**

Ahmet USTA

Yüksek Lisans Tezi

Tez Danışmanı

Prof. Dr. Kemal KÖYMEN

İSTANBUL – 2012

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

**DERS ÇİZELGESİ PROBLEMİ İÇİN BİR BİLGİSAYAR YAZILIM
SİSTEMİ**

YÜKSEK LİSANS TEZİ

Ahmet USTA

**Tez Danışmanı
Prof. Dr. Kemal KÖYMEN**

İSTANBUL – 2012

ÖZET

Eđitim-Öđretim kurumlarında haftalık ders programlarının oluřturulması; öđretim elemanı görevlendirmelerin yapılması, bölüm dersliklerin programlarının hazırlanması gibi işler, büyük zaman ve emek gerektiren rutin işlemlerdir. Yapılan bu işlemler, sadece tek bir bölüm için deđil, fakülte ya da yüksek okulun tüm bölümleri için düşünöldüğünde, istenmeyen ve kaçınılamayan sonuçlarla karşılaşmaktadır: “çakışmalar”. Zira bölümlerin kendilerine göre yaptığı planlamalarda, diđer bölümlerin imkânları göz ardı edilebilmekte, bunların çözülebilmesi için ise daha büyük bir dikkat ve çaba gerekmekte ve işlemler böylece uzayıp gitmektedir. Konuya çözüm olarak geliştirilen, tüm bölüm temsilcilerinin katıldığı toplantılar yapılmasıdır ki bu da pek mümkün olamamaktadır.

Uzman sistem yaklaşımıyla gerçekleştirilen bu çalışmada yazılımı yapılan “Ders Çizelge Oluřturma Sistemi” (DCOS) ile sarf edilen tüm çabalar minimuma indirilmektedir. Zira bölüm temsilcilerinin network üzerinde hazırladıkları ortak bir platformda hiçbir “çakışmayı hesap etmeden, verilerini (ve kısıtlarını) DCOS’a girmeleri kâfi gelmektedir. Program, çakışma olmayacak şekilde, tüm haftalık ders programını hazırlayarak, gerekli dokümanları vermektedir.

Bu çalışmada SQL Server ve C#.NET teknolojileri kullanılarak Maltepe Üniversitesi'nin ihtiyaç duyduğu kısıtların kolayca girilebildiđi, çözümlerin üretilebildiđi, deđiřtirebildiđi ve bilgilerin saklanabildiđi, otomatik ders programı hazırlayan bir uygulama geliştirilmiştir.

Bu tez 2012 yılında tamamlanmıştır ve 94 sayfadan oluşmaktadır.

Anahtar Kelimeler: Zaman çizelgesi, ders programı, ders zaman çizelgesi, NP problemi.

ABSTRACT

In educational institutions, some routine tasks such as preparing weekly course schedule, assigning faculty and classes takes a lot of time and effort. When these tasks are considered not only all department in terms of faculty or college, inevitable or undesirable results are faced. Forasmuch as departments do plan according to their needs, they ignore other departments' needs and facilities. In order to solve those problems, it takes more attention and effort. It is not possible to have a meeting of all related administrative staff.

In this study, these types of efforts are minimized in this software solution named "Course Schedule Preparing System" (CSPS) by using expert system approach. It is enough to enter all the data by department representatives without calculating any conflict in to CSPS. The software prepares weekly course schedule without having any conflict and furnishes all necessary documents.

In this study, software which prepares automated weekly course Schedule is developed by using SQL Server and C#.NET technologies in order to meet Maltepe University's needs.

This thesis has been completed in 2012 and it consists of 94 pages.

Keywords: Timetable, course timetable, course scheduling, NP problem.

TEŐEKKÜR

Bu tez konusunu seçmemde beni yönlendiren, destek ve yardımlarını esirgemeyen danışman hocam Maltepe Üniversitesi Rektörü sayın Prof. Dr. Kemal KÖYMEN'e, çalışmalarım sırasında beni yönlendiren ve gerekli kaynakların sağlanmasında yardımcı olan hocam sayın Prof. Dr. İhsan YILMAZ'a, fikirlerinden beni mahrum bırakmayan hocam sayın Doç. Dr. Oruç Raif ÖNVURAL'a, yardım ve desteklerini benden esirgemeyen hocam sayın Öğr. Gör. Dr. Fatih YÜCALAR'a ve tez uygulamasını geliştirirken yardımını benden esirgemeyen değerli meslektaş arkadaşım Musa Kazım ERGÜNEY başta olmak üzere tüm çalışma arkadaşlarıma, maddi ve manevi desteğini benden hiçbir zaman esirgemeyen çok değerli aileme ve çalışmalarım sırasında emeği geçen herkese teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET	IV
ABSTRACT	V
TEŞEKKÜR.....	VI
KISALTMALAR	IX
ŞEKİLLER.....	X
TABLolar	XII
1. GİRİŞ	1
2. ZAMAN ÇİZELGELEMESİNE GİRİŞ	4
2.1 Zaman Çizelgesinin Tanımı	4
2.2 Zaman Çizelgesi Üretimi İle İlgili Sorunlar.....	5
2.3 NP Probleminin Zorluğu.....	6
2.4 Zaman Çizelgesi Problemleri İçin Genel Bir Model.....	9
2.4.1 Zaman Çizelgesi Problemlerinin Genelleştirilmesi	9
2.4.2. Kaynak Ayrımı Üzerindeki Zorunlu Kısıtlar	14
2.4.3. Kaynak Ayrımı Üzerindeki Zayıf Kısıtlar	14
2.4.4. Zaman Çizelgesi Şablonları	15
2.5 Zaman Çizelgeleri Üretimi için Araştırma Teknikleri.....	15
2.5.1. İş-İdaresi Planlama.....	15
2.6 Zaman Çizelgesi Oluşturma Yöntemleri.....	17
2.6.1 Sıralı Yöntemler	18
2.6.2 Sezgi Üstü Yöntemler	18
2.6.3 Küme Yöntemler.....	18
2.6.4 Kısıt Tabanlı Yöntemler.....	19
3. EĞİTİM-ÖĞRETİM KURUMLARINDA ÇİZELGELEME PROBLEMLERİ ...	20
3.1 Ders Çizelgeleme Problemleri	22
3.2 Çözüm Yaklaşımları	24
3.3 Eğitim Öğretim Kurumlarında Çizelgeleme Problemi Üzerine Çalışmalar	28

4. UYGULAMA	35
4.1 Uygulama Veritabanı Modeli.....	36
4.2 Uygulama Modülleri	44
4.3 Programın Use-Case Diyagramı	59
4.4 Programın İş Birliği (Collaboration) Diyagramı.....	61
4.5 Sistemin Bilgi Tabanı.....	67
4.6 Programın Genel Algoritması	69
4.7 Programın İşleyişi	71
5. SONUÇ	78
KAYNAKLAR	79
ÖZGEÇMİŞ	85

KISALTMALAR

Kısaltma

MÜBİS

DCOS

Türkçesi

Maltepe Üniversitesi Bilgi İşletim Sistemi

Ders Çizelge Oluşturma Sistemi

ŞEKİLLER

Şekil 2. 1 Zaman Çizelgesi Öğeleri Arasındaki Bire-Bir İlişkiler	12
Şekil 2. 2 Zaman Çizelgesi Öğeleri Arasındaki Çoktan-Çoğa İlişkiler (a).....	13
Şekil 2. 3 Zaman Çizelgesi Öğeleri Arasındaki Çoktan-Çoğa İlişkiler (b).....	13
Şekil 4. 1 Kullanıcı Giriş Ekranı	45
Şekil 4. 2 Ders Çizelge Oluşturma Ekranı	45
Şekil 4. 3 Bilgi Ekle/Değiştir Ekranı.....	47
Şekil 4. 4 Yerleştirilemeyen Ders Listesi.....	47
Şekil 4. 5 Servis Bilgileri Ekranı	48
Şekil 4. 6 Programlanmış Ders Çizelge Ekranı.....	48
Şekil 4. 7 Hazırlanmış Ders Çizelge Ekranının Rapor Görünümü	49
Şekil 4. 8 Akademisyen Listesi Ekranı	50
Şekil 4. 9 Ders Listesi Ekranı.....	51
Şekil 4. 10 Fakülte Listesi Ekranı	52
Şekil 4. 11 Bölüm Listesi Ekranı	53
Şekil 4. 12 Derslik Listesi Ekranı	54
Şekil 4. 13 Bölüm Ders Akademisyen Listesi Ekranı.....	55
Şekil 4. 14 Sabit Dersler Listesi Ekranı	56
Şekil 4. 15 Çakışma Olmayacak Dersler Listesi Ekranı	57
Şekil 4. 16 Akademisyenler İçin Uygun Gün ve Saatler Listesi Ekranı	58
Şekil 4. 17 Haftalık Ders Çizelgeleme Problemi Use Case Diyagramı	60
Şekil 4. 18 Akademisyen Ekleme/Güncelleme Use-Case'i	62
Şekil 4. 19 Fakülte Ekle/Güncelleme Use-Case'i	62
Şekil 4. 20 Bölüm Ekleme/Güncelleme Use-Case'i	63
Şekil 4. 21 Ders Ekleme/Güncelleme Use-Case'i	63
Şekil 4. 22 Derslik Ekleme/Güncelleme Use-Case'i	64
Şekil 4. 23 Akademisyen Silme Use-Case'i	64
Şekil 4. 24 Fakülte Silme Use-Case'i	65
Şekil 4. 25 Bölüm Silme Use-Case'i	65
Şekil 4. 26 Ders Silme Use-Case'i.....	66

Şekil 4. 27 Derslik Silme Use-Case'i.....	66
Şekil 4. 28 Programın Yönetici Kullanıcıları İçin Genel Algoritması.....	69
Şekil 4. 29 Programın Fakülte Kullanıcıları İçin Genel Algoritması.....	70
Şekil 4. 30 Programın Yönetildiği Ana Ekran	71
Şekil 4. 31 Elle Ders Yerleştirme İşleminin Yapıldığı Ekran.....	74
Şekil 4. 32 Elle Ders Yerleştirme İşlemi Yapılırken Doluluk ve Boşluk Alanlarını Gösteren Ekran.....	75
Şekil 4. 33 Oluşturulmuş Ders Çizelge Ekranı	76
Şekil 4. 34 Oluşturulan Ders Çizelge Rapor Görünümü.....	77

TABLolar

Çizelge 2. 1 Bir Zaman Çizelgesi Örneği	4
Çizelge 2. 2 Karmaşıklık Fonksiyonlarının Karşılaştırılması	8
Çizelge 2. 3 Zaman Çizelgesi Problem Listesi	11
Çizelge 2. 4 Zaman Çizelgesi Problem Türleri Arasındaki Benzerlikler.....	11
Çizelge 3. 1 Literatür İncelemesinin Genel Gösterimi.....	34
Çizelge 4. 1 Fakülte Tablo Yapısı.....	36
Çizelge 4. 2 Bölüm Tablo Yapısı.....	37
Çizelge 4. 3 Dersler Tablo Yapısı	37
Çizelge 4. 4 Derslik Tablo Yapısı.....	38
Çizelge 4. 5 Akademisyen Tablo Yapısı.....	39
Çizelge 4. 6 Akademisyen Gün ve Saat Tablo Yapısı	39
Çizelge 4. 7 Bölüm Ders Akademisyen Tablo Yapısı	40
Çizelge 4. 8 Çakışma Olmayacak Dersler Tablo Yapısı.....	41
Çizelge 4. 9 Sabit Dersler Tablo Yapısı.....	41
Çizelge 4. 10 Ders Program Tablo Yapısı	42
Çizelge 4. 11 Servis Bilgi Tablo Yapısı.....	43
Çizelge 4. 12 Admin Data Tablo Yapısı	43
Çizelge 4. 13 Genel Veritabanı Mimarisi	44

1. GİRİŞ

Günümüzde bireysel ve örgütsel verimliliğin artırılması açısından tüketimi kolay olan zamanın etkin olarak kullanılabilmesi gereklidir. Özellikle çalışan insanların hayat kalitesi, günlük ve haftalık aktiviteleri, aile düzenleri doğrudan hazırlanan zaman çizelgeleri ile ilgilidir. Bu amaçla faaliyet ve kaynakların zaman temelinde bütünleştirilmesini sağlayan çizelgeler büyük önem kazanmaktadır.

Çizelgeleme problemleri kırk yıldan fazla bir süredir yöneylem ve yapay zekâ araştırmacılarının dikkatini çekmektedir. Hastanelerde çalışan hemşirelerin vardiyalarının düzenlenmesinde, spor müsabakaları için müsabaka programlarının hazırlanmasında, otobüslerin, trenlerin ve uçakların seferlerinin planlanmasında, eğitim kurumlarında ders ve sınav programlarının hazırlanmasında ve diğer birçok alanda yaygın olarak kullanılmaktadırlar.

Çeşitli alanlarda kullanılan çizelgelerin farklı kısıtlara, amaçlara, bireysel ve organizasyonel önceliklere sahip olması nedeniyle, farklı durumlara kolaylıkla uyarlanabilecek algoritmaların geliştirilememesi, birçok çizelgeleme modelinin ortaya çıkmasına neden olmuştur.

Çizelgeleme problemlerinde kısıtlı kaynakları kullanan görevler için uygun zaman dilimlerinin planlanması amaçlanmaktadır. Problemin yapısı gereği kısıtlar çok değişik olabilmekte ve farklı amaçlar elde edilmeye çalışılmaktadır. Örnek olarak, amaç bütün görevlerin çizelgelendiği toplam zamanın en aza indirgenmesi, belli bir zaman dönemi içinde tüm kısıtları sağlayan bir modelin oluşturulması veya kısıtların en az ihlal edildiği çözümün bulunması şeklinde belirlenebilmektedir.

Problemin ana noktasını kısıtlar oluşturmaktadır. Çözüm uzayını belirleyen ve ihlal edilmemesi gereken kısıtlar zorunlu; belli bir ceza ile ihlal edilebilen ve amaç fonksiyonuna katkı sağlayan kısıtlar ise zayıf kısıtlar olarak tanımlanmaktadır.

Eđitim-öđretim kurumları da zorunlu ve zayıf kısıtların söz konusu olduđu gerek sınav, gerekse ders çizelgeleri açısından son yıllarda arařtırmacıların yoğun ilgisini çekmektedir.

Son yıllarda bilgisayar kullanımı oldukça artmasına rağmen, birçok eğitim-öđretim kurumunda sınav ve ders çizelgeleri halen elle oluşturulmaktadır. Bu şekilde oluşturulan çizelgeler hem bu iş ile ilgilenen görevlilerin önemli ölçüde vaktini almakta, hem de oluşturulan çizelgeler personelin ihtiyaçlarını/isteklerini tam olarak karşılayamamaktadır.

Bu çalışmada, ders çizelgeleme problemi ele alınmıştır. Maltepe Üniversitesi'ne ait Fakülteler ele alınarak bir ders çizelgeleme problemi analiz edilmiş ve problemin çözümüne yönelik başarılı sonuçlar elde edilmiştir. İncelenen eğitim öđretim kurumunda ders hocalarının ders salonlarına atanması mevcut durumda elle yapılmakta ve planlayıcıların problem üzerinde yaklaşık iki hafta yoğun bir şekilde çalışmasını gerektirmektedir. Geliştirilen model ile problemin farklı boyutlarda ve farklı koşullarda çözümü yapılarak, makul süreler içerisinde arzu edilen çözümlere ulaşılabilirdiği gösterilmiştir. Ayrıca modele gerekli verilerin girişini kolaylařtırmak amacıyla farklı kullanıcılar tarafından rahatlıkla anlaşılabilir basit arayüzler hazırlanmıştır.

Bu tez çalışmasının amacı optimum çözümle derslerin haftalık ders programında dağılımının yapılması ve ders saat çakışmalarının mümkün olduğunca en aza indirilmesini sağlamaktır.

Bu amaca yönelik olarak çalışmanın ilk bölümünde zaman çizelgeleme probleminin günlük hayattaki yerine değinilmiş ve kullanıldığı alanlar hakkında bilgiler verilmiştir. Ayrıca eğitim-öđretim kurumları üzerindeki yeri ve önemi hakkında bilgiler verilmiştir.

İkinci bölümde ise zaman çizelgelemesi ele alınmıştır. Zaman çizelgesinin tanımına değinilmiş, zaman çizelgesi üretimi ile ilgili sorunlar sırasıyla belirtilip açıklanmış, NP probleminin zorluğu üzerinde durulmuş, zaman çizelgesi problemleri için genel bir model oluşturulup evreleri sırasıyla açıklanmış, zaman çizelgeleri üretimi için araştırma teknikleri üzerinde durulmuş ve son olarak zaman çizelgesi oluşturma yöntemleri konusunda bilgiler verilmiştir.

Üçüncü bölümde ise çalışmanın ana konularından birisi olan eğitim öğretim kurumlarında çizelgeleme problemleri incelenmiştir. Daha sonra ders çizelgeleme problemleri hakkında bilgi verilip, çözüm yaklaşımlarından bahsedilmiştir. Son olarak ise eğitim öğretim kurumlarında çizelgeleme problemi üzerine yapılan çalışmalar hakkında bilgiler verilmiştir.

Dördüncü ve son bölüm olan projenin uygulaması üzerinde durulmuştur. Bu bölümde, uygulama ayrıntılı olarak incelenmiştir. İlk olarak uygulamanın veritabanı modeli üzerinde durulmuş ve uygulamaya ait modüller hakkında bilgiler verilmiştir. Daha sonra uygulamanın genel bir use case diyagramı çizilmiş ve gerekli açıklaması yapılmıştır. Daha sonra uygulamayı oluşturan modüllere ait iş birliği (collaboration) diyagramı çizilmiştir. Ayrıca sistemi oluşturan bilgi tabanı üzerinde durulmuş ve programın işleyişi hakkında bilgiler verilmiştir. Son olarak uygulamanın genel bir algoritması çizilip açıklanmıştır.

2. ZAMAN ÇİZELGELEMESİNE GİRİŞ

Günümüzde zaman çizelgeleme, problemlerin ve kısıtların ortaya çıkması için bir zaman periyodu üzerinde düzenlenmiş bir taslak zaman çizelgesini kapsayan ve daha çok elle yapılan bir süreçtir. Elde edilen bu zaman çizelgesi, çoğunlukla bu çizelgeyi takip etmek zorunda olan insanlar için uygundur. Ancak, bu zaman çizelgesinin elde edilmesi için daha çok zaman ve iş gücü harcanır. Ayrıca bu zaman çizelgesi üzerinde bazı iyileştirmelerin yapılması mümkün olabilir [1].

2.1 Zaman Çizelgesinin Tanımı

Bir zaman çizelgesi, genel hatlarıyla belirli bir zaman aralığında insanların ve kaynakların nerede ve ne zaman tanımlanacağını göstergesi olarak tanımlanabilir. Pek çok insan hafta, gün ve zaman dilimlerinin bir tabloda sunulduğu okullardaki ders çizelgelerine aşinadır. Bir ders çizelgesi örneği Çizelge 2.1’de sunulmuştur.

Çizelge 2. 1 Bir Zaman Çizelgesi Örneği

Günler	09.00 – 09.50	10.00 – 10.50	11.00 – 11.50
Pazartesi	Fizik I Prof. Dr. İdris Gümüş MA5	Fizik I Prof. Dr. İdris Gümüş MA5	
Salı	Matematik II Yrd. Doç. Dr. Sibel Ersan MZ2	Türkçe Öğr. Gör. Hafize Kordel M11	Türkçe Öğr. Gör. Hafize Kordel M11
Çarşamba	Kimya Yrd. Doç. Dr. İlkün Orbak MZ3	Kimya (Lab.) Yrd. Doç. Dr. İlkün Orbak Kim Lab. I	

Zaman çizelgesi, farklı şekillerde temsil edilebilir. Her öğrenci ve öğretmen, çalışma konularına bağlı olarak kendi zaman çizelgelerine sahip olabilir. Bunun yanında dersler için tahsis edilecek sınıfların durumunu da göz önünde bulundurmak gerekir. Bu gibi kıstaslar dikkate alındığında aynı zaman çizelgesi üzerinde farklı bakış açıları olacaktır.

Zaman çizelgeleri için gerekli olan diğer durumlar:

1. Üretim – üretim hatları, proje planlama,
2. Seyahat – trenler, otobüsler, ...vb.
3. Üniversite/Okul sınavları,
4. Üniversite Dersleri,
5. Okul zaman çizelgeleri,
6. TV/Radyo/Medya Planları,
7. Konferans/Toplantılar.

Bu durumlar, yer ve zaman dilimlerinin sayısı zaman çizelgesi üzerinde denirken kaynakların sayısına bağlı olarak değişmesinden dolayı karmaşık zaman çizelgelerinin oluşmasına neden olur.

2.2 Zaman Çizelgesi Üretimi İle İlgili Sorunlar

İlk bakışta zaman çizelgesi problemleri, basit gibi görülebilir. Örnek olarak bir zaman çizelgesinin bir okul içerisindeki kullanımında, ortak yönü olmayan öğrenciler ile grup konuları için basit bir algoritma kullanılabilir ve uygun olan herhangi bir sırada odalar sınıflara atanabilir. Daha sonra, sahip oldukları niteliklere göre sınıflara öğretmenler atanabilir.

Bazı sorunlar belirgin hale gelir. Eğer bir öğrenciye daha fazla konu çalışabilmesi için konu seçme imkânı verilirse, dersler arasında çakışmalar meydana gelecek/gelecektir [2]. Çakışma, bir öğrenci ya da bir hocanın bir anda birden fazla yerde planlanması olarak tanımlanır.

Bu problemin net bir açıklaması, bir öğrencinin sadece haftada bir kez okutulan bir grup konuyu okuyacağı durum olacaktır. Öğrenci dersi kaçırsa, bu dersin tekrarı yoktur. Aynı anda iki veya daha fazla konu olursa, öğrenci bir dersi kaçırmak zorunda olacaktır. Bu nokta da planda çakışma vardır. Her bir öğrenci tarafından alınmış konuların sayısı, konuların veya öğrencilerin sayısı arttıkça bu problem daha kötü bir hale gelecektir.

Meydana gelebilecek diğer problemler:

- İki veya daha fazla ders verecek bir hocayı aynı zaman dilimine yerleştiremeyiz.
- Bazı sınıflar, laboratuvar donanımları gibi sınırlı özel kaynaklar gerektirebilir veya bir sınıf oldukça büyük olabilir.

Bu problemler, bir öğretim ortamı için tanımlanmıştır. Benzer problemler diğer zaman çizelgesi problemleri içinde meydana gelebilir ve ayrıca etki eden belirli kısıtlamalar ilave olabilir. Bu kısıtlamaları birleştirdiğimizde sıradan bir zaman çizelgesi elde ederiz.

2.3 NP Probleminin Zorluğu

Her problem için, problemin büyüklüğü ve problemi çözmek için gereken zaman miktarı arasındaki ilişki iyi tanımlanmış bir algoritma ile çözülebilir [3, 4].

Açıkçası bilgisayar üzerinde bir algoritmayı uygulasak, çözümümüz ne kadar hızlı olsa bile bilgisayarın hızı algoritmanın çalışmasını direkt olarak etkiler. Yeni bilgisayar donanımları, bir algoritmayı öncekinden daha hızlı olarak çalıştırabilir ve daha kısa sürede çözümler bulabilir. Ancak, problem aynı kalmıştır.

Daha hızlı bir bilgisayarın belirli bir zamanda daha fazla döngü çalıştırabildiğini varsayarak, bir çözüm bulmak için gereksinim duyulan bilgisayar işlemcisi döngülerinin sayısını hesaplayabiliriz. Ancak daha yavaş bir makinedeki gibi aynı döngü sayısını tamamlamanız gerekecektir. Ne yazık ki, bu tip ölçüm de bilgisayar donanımına bağımlı bir eğilimdir.

Bu çözüm, bir algoritmanın zaman karmaşıklığı kavramını kullanmak içindir. Bir zaman karmaşıklığı fonksiyonu, herhangi bir algoritma için mevcuttur. Fonksiyon, algoritma çözümünün problemin büyüklüğüne göre, algoritma zaman gereksinimlerinin nasıl değiştiğini tanımlar. Ancak, bir çözüm bulmak için ne kadar zaman gerektiği açık değildir [3,4].

Bir örnek, zaman karmaşıklık fonksiyonu $t = v$ olsun. Burada t zaman karmaşıklığını, v ise problemin büyüklüğünü ifade etmektedir. Basit olarak bu, problemin boyutu iki katına çıkacağından, çözümü içinde iki katı zamana ihtiyaç olduğu anlamına gelmektedir.

Sıralama algoritmaları gibi basit algoritmalar genel olarak, $t=v$, $t=2^v$, $t=v^2$ gibi zaman karmaşıklığı fonksiyonlarına sahiptir.

Problemin büyüklüğü yani v arttıkça, t çarpıcı biçimde artmaz, bu algoritmaların “polinomial zaman karmaşıklığına” sahip olduğunu söyleyebiliriz. Çünkü bu algoritmalar, verilen bir $p(v)$ polinomu gibi aynı sraya sahiptir. $P(v)$ polinomu için $v+v^2+v^3$ örnek olarak verilebilir [3,4].

Gezgin satıcı problemi, grafik boyama ve zaman çizelgesi problemleri gibi problemlerde, problem büyüklüğü artarken, bu problemi çözmek için gerekli zamanda üssel olarak artmaktadır. Bu problemlerin polinomial olmayan zaman karmaşıklığı olduğu söylenir ve en az her bir NP problem kadar zor olan problemlerin bulunduğu sınıf (NP-hard) olarak tanımlanır [3,4].

Zaman karmaşıklığı fonksiyonları, bir problemin çözümünün ne kadar süre alacağını söylememesine rağmen, verilen herhangi bir problemin büyüklüğü için sadece gerçek zamanlı ölçeklendirilebilir bir değere sahip olacaktır.

Aşağıdaki tabloda, zaman karmaşıklığı fonksiyonu bir çözüm bulmak için gerekli bilgisayar işlemcisi döngülerinin sayısını döndürür. Bir döngünün 1 mikro saniye (0,000001 saniye) aldığı varsayarsak, belirli bir zaman karmaşıklığı algoritması ile bir problemi çözmek için ne kadar süreceğini hesaplayabiliriz [5]. Bu durum Çizelge 2.2’de sunulmaktadır.

Çizelge 2. 2 Karmaşıklık Fonksiyonlarının Karşılaştırılması

Zaman Karmaşıklık Fonksiyonu	Problem Büyüklüğü					
	10	20	30	40	50	60
V	0.00001 saniye	0.00002 saniye	0.00003 saniye	0.00004 saniye	0.00005 saniye	0.00006 saniye
V ²	0.0001 saniye	0.0004 saniye	0.0009 saniye	0.0016 saniye	0.0025 saniye	0.0036 saniye
V ⁵	0.1 saniye	3.1 saniye	24.3 saniye	1.7 dakika	5.2 dakika	13 dakika
V ¹⁰	2.7 saat	118.5 gün	18.7 yıl	3.3 yüzyıl	30.1 yüzyıl	192 yüzyıl
2 ^v	0.001 saniye	1 saniye	17.9 dakika	12.7 gün	35.7 yıl	366 yüzyıl
3 ^v	0.59 saniye	58 dakika	6.5 yıl	3855 yıl	2.28*10 ⁸ yüzyıl	1.3*10 ¹³ yüzyıl
V!	3.6 saniye	771 yüzyıl	8.4*10 ¹⁶ yüzyıl	2.6*10 ³² yüzyıl	9.6*10 ⁴⁸ yüzyıl	2.6*10 ⁶⁶ yüzyıl

Tablonun sol tarafındaki sütunda yer alan zaman karmaşıklığı fonksiyonlarının her biri, çeşitli büyüklüklerdeki problemleri çözmek için geçen süreyi gösterir. Sorunun büyüklüğü sadece bir rakam olabilir. Aynı zamanda bir giriş değeri aralığı veya bir algoritma için parametre sayısını temsil edebilir. Bu değerlerin neyi temsil ettiği önemli değildir. Her bir problemi çözmek için gerekli zaman miktarı, NP-hard problemleri ile NP-hard problemi olmayanlar arasındaki farkı gördüğümüzde artar.

Zaman karmaşıklık fonksiyonlarının ilk dördü için gerekli zaman artışı, sonraki üç fonksiyondan daha düşüktür. Dört fonksiyonda sınırda görünmesine rağmen, problemin büyüklüğü artmış olsa da NP-hard problemleri çözmek için gereken zamanın daha yüksek bir oranda arttığını görebiliriz [3].

2.4 Zaman Çizelgesi Problemleri İçin Genel Bir Model

Bu bölümde, zaman çizelgesi problemlerinin temsili için genel bir model üreterek ve benzerliklerini tanımlayarak, zaman çizelgesinin farklı türlerine bakacağız. Bu modeli, bir çözüm bulmaya kalkışmadan önce, algoritmada kullanmak üzere hangi verilerin gerektiğini tanımlayacağız.

2.4.1 Zaman Çizelgesi Problemlerinin Genelleştirilmesi

Belirlenmiş olan zaman çizelgesi problemleri aşağıda listelenmiştir:

1. İmalat - üretim hatları, proje planlama.
2. Seyahat - trenler, otobüsler, vb.
3. Üniversite / okul sınavları.
4. Üniversite haftalık ders programları.
5. Okullardaki haftalık ders programları.
6. T.V. / Radyo / Medya yayın akış planları.
7. Konferanslar / toplantılar.

Hemen tespit edilebilir bazı anahtar topluluklar vardır. Yukarıda verilenlerin her biri, bir zaman ve/veya mekâna atanabilmek için gerekli bir kaynak veya kaynak numarasına sahiptir.

Genel bir model oluřturma giriřiminde bulunmamıza rađmen, yukarıdaki çizelgeleme problemi türlerinin bazılarını atlayarak, daha dođru bir şekilde modelinin kapsamını tanımlayabiliriz. İlk bařta, genel bir modelde mümkün olandan çok daha fazla detayı inceleyen üretim/proje programlarını oluřturmada kullanılan birçok özel teknik vardır [4, 6, 7]. Bir çözüm oluřturmadan önce büyük miktarlarda veriye gereksinim duyan seyahat çizelgelerini genel bir model içinde sabitlemek zor problemlerdendir. Büyük miktarlardaki veri gereksinimlerine örnek olarak, gün boyunca yolcu sayısının deđiřimi, seyahat süreleri gibi yerlere iliřkin veriler, pozisyonlar, rota bilgileri verilebilir.

TV / Radyo programları otomasyon için gerçek bir problem sayılmaz. Bir program atanırken, genellikle yayınlanacak programın kanalına iliřkin yapılacak hiçbir seçim yoktur. Yayın zamanı, sosyal düzeni sabitlemek için seçilebilir.

Bir modeli temsil etmek üzere oluřabilecek zaman çizelgesi problemlerinin geri kalanı ařađıdaki gibidir:

1. Üniversite / okul sınavları.
2. Üniversite haftalık ders programları.
3. Okullardaki haftalık ders programları.
4. Konferansların / büyük toplantıların zaman çizelgeleri.

Daha genel olarak, farklı varlık grupları herhangi bir zaman çizelgesi probleminin bir veya birkaç kez üstesinden gelebilir ve her zaman deđiřen mevcut kaynaklara sahip olabilir. Aynı zamanda, zaman çizelgesi için mevcut zaman ve lokasyonların sayısı sınırlıdır. Őimdi, yukarıda verilen zaman çizelgesi problemleri listesine bakacađız. Bu problemlerin her biri için, temel öğeleri tanımlarsak:

Çizelge 2. 3 Zaman Çizelgesi Problem Listesi

Problem Türü	Öğeler
Sınav çizelgeleri	Danışman, öğrenci, ders, sınıf, zaman aralığı
Üniversite dersleri	Öğretim üyesi, öğrenci, ders, sınıf, zaman aralığı
Okul ders programı	Öğretmen, öğrenci, ders, sınıf, zaman aralığı
Konferans/toplantılar	Konuşması, katılımcı, başlık, mekân, zaman aralığı

Tamamlanmış bir zaman çizelgesinde doğrudan atanması gereken öğeler vardır. Diğer öğeler, bu öğelerin nitelikleri ve bunlar arasındaki ilişkiler, üstteki öğelerin atanmasını etkileyecektir. Ancak tamamlanmış zaman çizelgesi görünmez. Diğer öğeleri tanımlayacağız, öncelik olarak bu temel öğeleri inceleyeceğiz.

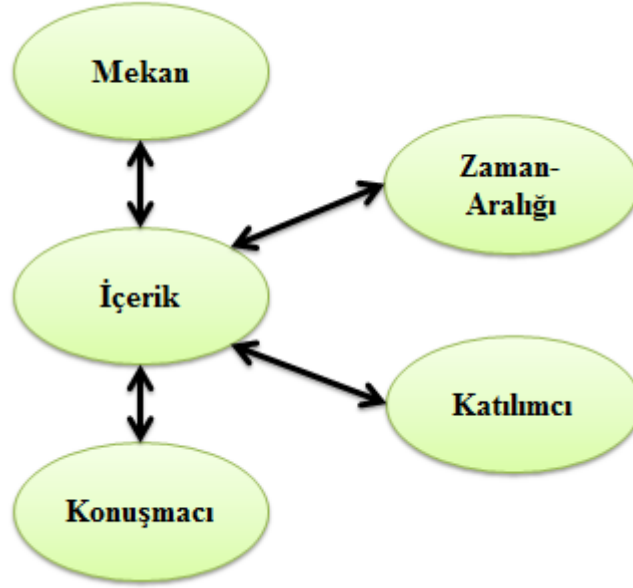
Zaman çizelgesi problemlerinin dört türü arasındaki benzerlikleri görmek kolaydır.

Aşağıdaki tabloda görüldüğü üzere, öğeler gruplanabilir ve genel bir ad verilebilir:

Çizelge 2. 4 Zaman Çizelgesi Problem Türleri Arasındaki Benzerlikler

Problem Türü				
Genel Öğe Adı	Sınav	Ders	Okul	Konferans
Konuşmacı	Danışman	Okutman	Öğretmen	Konuşmacı
Katılımcı	Öğrenci	Öğrenci	Öğrenci	Katılımcı
İçerik	Ders	Ders	Ders	Başlık
Mekân	Mekân	Mekân	Mekân	Mekân
Zaman-aralığı	Zaman-aralığı	Zaman-aralığı	Zaman-aralığı	Zaman-aralığı

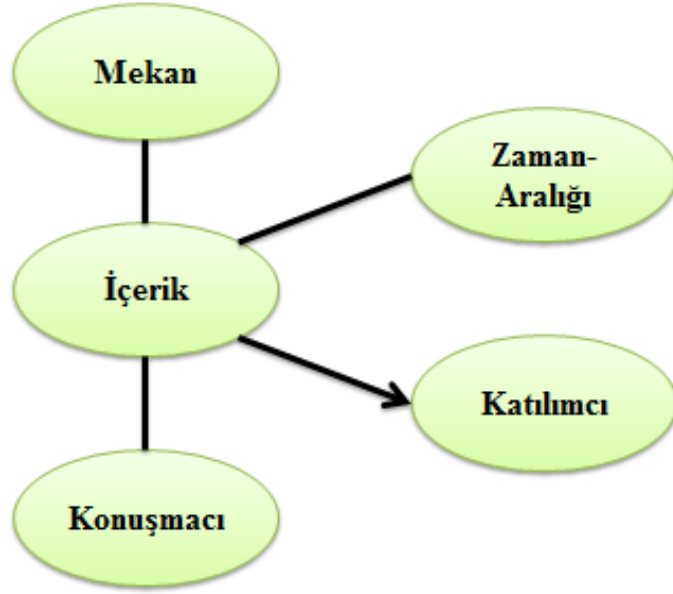
Öğeler için bazı basit ilişkileri uygulayabiliriz:



Şekil 2. 1 Zaman Çizelgesi Öğeleri Arasındaki Bire-Bir İlişkiler

Pek çok insan, belirli bir konu veya içeriğe göre konuşmacı olmak için yetenekli olabilir. Aynı zamanda bir konuşmacı, pek çok farklı konu ve içerik ile ilgili olarak konuşabilir. Benzer olarak, pek çok katılımcı belirli bir içeriğe sahip bir toplantıya katılmak isteyebilirler ve aynı zamanda çok farklı içeriğe sahip bir toplantıya da katılmak isteyebilirler.

Bu ilişki modellerinin tamamı için, çoktan çoğa ilişki söz konusudur. İşte bu bir zaman çizelgesi ataması için modeldir.



Şekil 2. 2 Zaman Çizelgesi Öğeleri Arasındaki Çoktan-Çoğa İlişkiler (a)

Bu daha kullanışlı bir modeldir. Her zaman çizelgesi ataması için uygulanması gereken sınırları gösterir. Zaman çizelgesinde bir atama, doğrudan ya da dolaylı olarak yukarıda belirtilen varlıkların tamamını kapsayacak bir şekilde kullanılmalıdır. Konuşmacı ve katılımcılar, tek bir mekânda ve zaman aralığında belirli bir içeriğe göre bir araya gelirler. Bu kısıtlamaların yerine getirilmesi gerektiği için bu kısıtlamalar zor kısıtlamalar olarak adlandırılır.



Şekil 2. 3 Zaman Çizelgesi Öğeleri Arasındaki Çoktan-Çoğa İlişkiler (b)

Her bir mekân, bu mekânla ilişkili bir kaynak listesine sahiptir. Bunlar, koltuklar, bilgisayar veya projeksiyon gibi mekândaki mevcut kaynaklardır. Genel bir model oluşturmak için, bu basit ilişkilerin yeterli olduğu öngörülmektedir.

2.4.2. Kaynak Ayrımı Üzerindeki Zorunlu Kısıtlar

Genel bir modeli temel alan zaman çizelgesi problem türlerinin her biri, konuşmacı ve katılımcılara bir mekân ve zaman-aralığının nasıl ayrılacağını sınırlayan ortak kısıtlara sahiptir. Bunlardan en barizi, bir kimse herhangi bir zamanda birden fazla yerde olabilir. Bu gibi kısıtlar, “Zorunlu Kısıtlar” olarak bilinir. Zorunlu kısıtları ihlal eden bir zaman çizelgesi, gerçek dünyada mümkün değildir [2]. Zorunlu kısıtlar:

- Bir konuşmacı, herhangi bir zamanda sadece tek bir yere atanabilir.
- Bir katılımcı, herhangi bir zamanda sadece tek bir içeriğe atanabilir.
- Bir yer, herhangi bir zamanda sadece tek bir içeriğe atanabilir.
- Mekânlar maksimum kapasitesini aştığında, katılımcılara tahsis edilemez.

2.4.3. Kaynak Ayrımı Üzerindeki Zayıf Kısıtlar

Zayıf kısıtlar, uygun bir zaman çizelgesinin meydana gelebilmesi için kaynak ayrımı üzerindeki sınırlar olarak tanımlanabilir [2]. Örneğin; öğrenciler arka arkaya sınav olmasını istemediklerinden, sınav dönemi boyunca tüm sınavları mümkün olduğunca yaymak isteriz. Herhangi bir kişi ile ilgili bir zaman çizelgesi elde etmek istediğimizde, göz önünde bulundurmamız gereken şeyler şunlardır:

1. Birbirini takip eden zaman çizelgesi atamalarının oluşumunu en aza indirin.
2. Birbirini takip eden atamalar oluşursa, bir satırdaki atamaların sayısını en aza indirin.
3. Zaman çizelgesi atamaları arasındaki boşlukları en üst düzeye çıkarın. (Özellikle bu, sınavlar için uygulanabilir)
4. Zaman çizelgesi atamaları arasındaki boşlukları en aza indirin.
5. Seyahat zamanlarını/mesafelerini en aza indirin.

Bir zaman çizelgesi içinde istenilen özellikler, ne tür bir zaman çizelgesi oluşturulacağına bağlı olarak 3. ve 4. maddelerle çelişmektedir. 3. madde sınavlar için, 4. madde ise zaman çizelgesi içinde yer alan günler için istenilen bir durumdur.

2.4.4. Zaman Çizelgesi Şablonları

Şimdiye kadar bütün konuşmacılar, katılımcılar ve içerikler için herhangi bir tahsisin yapılmadığını varsayarsak, herhangi bir zaman dilimi kullanılabilir ve herhangi bir yere atanabilir. Aslında, diğer yükümlülüklerle ve kaynaklara sahip insanlar, herhangi bir nedenden dolayı kullanılmıyor olabilir. İnsanlar ve kaynaklar kullanılabilir olduğunda, bir yöntem tanımına ihtiyaç duyarız. Bu ihtiyaç duyulan hizmet için, insanların nerede ve ne zaman olacağını dağıtabilen bir şablon tanımlamalıyız.

Şablonlar, doğrudan kullanılamazlar ve aynı zamanda günlük / haftalık zamanları işe yaramaz bir hale getirirler. Üstü kapalı olarak açıklarsak, zaman tahsisi nihai uygulamayı aşırı karmaşık hale getireceği gerekçesiyle kullanılmaz. Ayrıca, her kaynak ve katılımcı için bir şablon tanımlamak mümkün olmayabilir. Bu yüzden, sadece konuşmacılar için şablon sağlanmış olacaktır.

2.5 Zaman Çizelgeleri Üretimi için Araştırma Teknikleri

2.5.1. İş-İdaresi Planlama

Planlama çalışmalarının çoğu üretim süreçleri ile ilgilidir. İş-idaresi planlama problemi, bir iş yerindeki makinelere uygun işin atanmasını ifade eder; ayrıca model planlama problemlerinin birçoğuna uygulanabilir. Örneğin; iş, hastanedeki bir hasta olabilirken, makine ise bir doktor olabilir. Bir iş-idaresi problemi için temel şablon, işleme (işler) gereksinim duyan ve bu işlemi gerçekleştirecek (makine) bir dizi varlıktır [3].

Genel iş-idaresi olarak adlandırılan kendi iş dizisine sahip n tane işin makine üzerinde işletilmesidir. Bu model, bütün iş-idaresi problemlerini temsil etmek için kullanılabilir. Ayrıca belirli bir biçimde, dört parametre kullanımını temsil edebilen problemler de tanımlanabilir [3].

- n: İşlerin sayısı
- m: Makinelerin sayısı
- A: Makine üzerinden işlerin akışını tanımlar. $m=1$ (sadece bir makine) olduğu sürece bu parametre ihmal edilir. Aşağıdakilerden herhangi biri olabilir; F-akış idaresi. Her iş makineleri aynı sırayla geçer.
- P: İş-idaresi permütasyonu. İş-idaresi gibi, ama burada her makine için iş emri aynıdır.
- G: Genel iş-idaresi.
- B: Değerlendirilen bir plan ile performans ölçümünün tanımlanmasıdır.

Son parametre, bir planı oluştururken en az veya en üst performans ölçütlerini belirtmek için kullanılır. Örneğin; toplam plan zamanı, mümkün olduğu kadar çalışan pahalı makinenin veya iş yerindeki periyotların ortadan kaldırılmasını temel alır. Bu haliyle iş-idaresi, tam olarak bir zaman çizelgesini temsil edemez. Örneğin; bir okul için hazırlanmak istenen zaman çizelgesi, sadece bir grup öğrenci için yapılmamalıdır, bu öğrenciler için dersi verecek öğretmen ve anlatılacak dersin de göz önünden bulundurulması gereklidir.

Tüm sınıf büyüklüklerinin ve derslerin aynı olduğunu varsayarsak, tüm sınıflarda bu dersler öğretilir. O zaman problem, sınıfları “işler” ve zaman aralıklarını ise “makinelere” olarak temsil eder. Bir öğretmenin bir sınıfa atanması, ayrı bir iş-idaresi problemi olarak ele alınabilir. Burada sınıflar “işleri”, öğretmenler ise “makinelere” temsil etmektedir. Son duruma göre iki iş-idaresi problemini, zaman çizelgesi problemi içinde bölmüş olduk. Ancak, son durumda sınıflarda hala sabit bir öğrenci grubu olduğunu ve tüm derslerin bu sınıflarda öğretilebileceğini varsaymaktadır. Bu ekstra değişkenler, işleri makinelere dağıtabilecek bir şekilde sınırlar ve aynı zamanda bir zaman çizelgesi probleminin her bir örneğini çeşitlendirerek bu sınırlama işlemini gerçekleştirirler.

Standart iş-idaresi planlama teknikleri içinde, etki alanına bağımlı bu kısıtlamalara tamamen izin verilmez. Ayrıca, zaman çizelgesi problemini iki ayrı iş-idaresi problemine bölerek, zaman çizelgelerine karar vermek mümkün olmayabilir.

Okul zaman çizelgesi örneğini ele alalım. İlk problem, derslerin zaman çizelgesi aralıklarına tahsis edilmesidir. Bu problemi başarılı bir şekilde çözdüğümüzü varsayarsak, bir öğretmenin derse atanması ise ikinci problem olarak karşımıza çıkar. Bu probleme çözüm bulamayabiliriz, çünkü bazı derslere özel öğretmenler gerekebilir ve aynı zamanda bu öğretmenler farklı bir sınıfa tahsis edilmiş olabilir. Böyle bir durum oluştuğunda, ilk iş-idaresi problemine geri döneriz ve zaman çizelgesi aralıklarında bu derslerin dağıtımını düzenleriz. O yüzden, bu iki sorun birbirinden tamamen bağımsız değildir. Bundan dolayı tüm değişkenler ile başa çıkabilmek için gerçekten tek bir tekniğe ihtiyaç duyulur.

2.6 Zaman Çizelgesi Oluşturma Yöntemleri

Zaman çizelgesi oluşturmada saat boşluklarına atanacak faaliyet sayısının çok olması ve çok kısıt olması olası çözüm kümesinin çok büyümesine neden olur. Zaman çizelgesini oluşturmak çok zor olabilir ve elle yapmaya kalkışıldığında çok fazla zaman ve iş gücü gerektirebilir [8].

Zaman çizelgesi problemleri 40 küsur senedir bazı bilim dallarının araştırma konusu olmuştur (Yapay Zekâ, Yöneylem Araştırması). Literatürde tanımlanmış birçok çözüm yöntemi vardır. Bu yöntemler genel olarak dört ana başlık altında toplanabilir.

Bunlar;

1. Sıralı (Sequential) Yöntemler
2. Sezgi Üstü (Metaheuristic) Yöntemler
3. Küme (Cluster) Yöntemler
4. Kısıt Tabanlı (Constraint Based) Yöntemler

2.6.1 Sıralı Yöntemler

Bu yöntemler faaliyetleri ortam sezgiselleri kullanarak sıralar ve sonra geçerli saat boşluklarına atarlar. Böylece hiç bir faaliyet birbiriyle çakışmaz. Sıralı yöntemlerde zaman çizelgesi problemleri genellikle graf olarak gösterilir. Faaliyetler düğüm, faaliyetler arası çakışma da ayrıt olarak gösterilir. Örneğin bazı öğrenciler iki faaliyete birden katılmak zorundadır. Düğümler arasında çakışmayı gösteren bir ayrıt vardır. Çakışmasız bir zaman çizelgesi oluşturulması bir graf renklendirme problemi gibi modellenabilir. Zaman çizelgesindeki her bir saat boşluğu graf renklendirme problemindeki bir renkte karşılık gelir ve grafın düğümleri de iki komşu düğümün aynı renkle renklenmeyeceği şekilde renklendirilir [8].

2.6.2 Sezgi Üstü Yöntemler

Sezgi üstü yöntemler bir veya daha fazla önceden üretilmiş çözümle başlar ve verilen parametreler doğrultusunda yeni bir sonuç üretir. Bu sonuç başlangıçtakinden daha iyi değilse, istenilene yakın bir sonuç bulununcaya kadar algoritma rekürsif bir şekilde çalışmaya devam eder. Bunun için arama stratejileri kullanılır (Yerel arama, Tabu arama vb.). Bu arama algoritmalarının hepsi çok iyi sonuçlar üretebilir fakat maliyetleri yüksektir [8].

2.6.3 Küme Yöntemler

Bu yöntemlerde durum kümesi, zorunlu kısıtları karşılayacak gruplara ayrılır ve sonra zayıf kısıtları karşılamak için gruplar zaman aralıklarına atanırlar. Durum gruplarını zaman aralıklarını atama problemini çözmek için farklı optimizasyon teknikleri kullanılmıştır. Bu yaklaşımların temel dezavantajı; durum kümeleri algoritmanın başında oluşturuluyor ve bu da zayıf bir zaman çizelgesi oluşmasına neden olmasındır.

2.6.4 Kısıt Tabanlı Yöntemler

Bu yöntemlerde zaman çizelgesi problemi bir deęişkenler kümesi olarak modellenir (örneğin faaliyetler). Bu deęişkenlere, belirli sayıdaki kısıtları karşılamak için derslik sayısı, saat boşlukları gibi deęerler atanmalıdır. Genellikle kaynakları faaliyetlere atamak için belli sayıda kural tanımlanır. Eğer hiç bir kural çözüme uygun deęilse geri dönüş yapılır ve uygun bir çözüm bulunana kadar işlem devam eder.

Kısıt tabanlı yöntemler için Kısıt Mantıklı Programlama (Constraint Logic Programming) dilleri geliştirilmiştir. Araştırmaların çoęu farklı problemlere adapte edilebilecek teknikler geliştirmek üzere yapılmıştır. Ders programı zaman çizelgesi oluşturmada yaygınca kullanılan iki deklaratif dil geliştirilmiştir; CHIP ve ECLIPS. Başka zaman çizelgesi problemleri için de farklı diller geliştirilmiştir; WPROLOG, COASTOOL, Oz ve EaCL [8].

3. EĞİTİM-ÖĞRETİM KURUMLARINDA ÇİZELGELEME PROBLEMLERİ

Çizelgeleme problemi “zaman uzayında kaynakların, belirli kısıtlar altında, belirlenen hedeflere mümkün olduğu kadar ulaşılarak tahsis edilmesi” şeklinde tanımlanmaktadır [9].

Çizelgeleme problemleri birçok alanda kullanılmakla birlikte eğitim öğretim kurumlarındaki uygulamaları oldukça fazladır. Özellikle çizelgeler oluşturulurken yönetimin, eğiticilerin ve öğrencilerin isteklerinin ve önceliklerinin sağlanabilmesi problemin çözümünü zorlaştırmaktadır. Her eğitim kurumu kendine özel kısıtlara ve amaçlara sahip olduğu için bütün durumlar için kullanılacak genel bir model oluşturulabilmesi mümkün değildir [10].

Akademik bir kurumda mevcut operasyonel kuralları ve ihtiyaçları, tüm personelin ve öğrencilerin istek ve ihtiyaçlarını da aynı anda göz önünde bulundurarak sağlayabilecek bir çizelge oluşturulması, bu işlemlerle uğrasan personel için oldukça zor ve zaman alıcı bir faaliyettir. Birçok kurumda bu işlem yönetim kademesine verilmiştir ve genellikle yapılmakta olan daha önceki yıllarda yapılmış olan çizelgeler üzerinde yeni gelişen durumlara göre küçük değişiklikler yaparak yeni çizelgeyi oluşturmaktır. Fakat son yıllarda değişimlerin çok fazla ve çok hızlı olması nedeniyle, eski çizelgeler üzerinde değişiklikler yapılarak yeni çizelgelerin oluşturulması her zaman en iyi sonucu vermemektedir. Yazılım ve donanım teknolojisinde meydana gelen gelişmeler, etkin ve istenilen çizelgelerin oluşturulabilmesi için bilim dünyasına yeni fırsatlar yaratmıştır [11].

Bunun yanında akademik kurumlarda ders ve sınav çizelgeleri üzerine bilimsel ve ticari amaçlarla birçok çalışma yapılmış olmasına rağmen hala birçok kurumda çizelgeler elle oluşturulmaktadır. Çeşitli üniversitelerde ve okullarda mevcut farklı problem formülasyonları, okulların özel ihtiyaçlarına adapte edilebilecek

programların ve farklı kullanıcılar tarafından kolaylıkla işletilebilecek kullanıcı arayüzlerinin yetersizlikleri bu duruma neden olmaktadır [12].

İngiliz üniversitelerinde sınav çizelgeleri oluşturulurken bilgisayar kullanımının belirlenmesi amacıyla yapılan bir araştırmaya katılanların %21'i çizelgeleri bilgisayarda oluşturduklarını, %42'si çizelgeyi oluştururken bilgisayardan yararlandıklarını, %37'si ise bu amaçla bilgisayar kullanmadıklarını belirtmişlerdir [13].

Eğitim kurumlarında hazırlanan çizelgeler genelde üç ana başlık altında toplanmaktadır: Üniversite ve okul sınav programları, üniversite ders programları ve okul ders programları [14]. Bunun yanında bazı çalışmalarda okul nöbetçi listelerinin hazırlanması da çizelgeleme problemi şeklinde ele alınmıştır [15]. Sınav programlarının ve ders programlarının genellikle benzer kaynaklar kullanıldığı için birçok yönden birbirine benzer özelliklere sahip olsa da aralarında bazı hususlar farklılıklar göstermektedir [16]. Örneğin sınav programları yapılırken öğrencilerin sınavlara hazırlanabilmesi için iki farklı dersten ardışık zaman dilimlerinde sınav olmaması arzu edilirken, ders programlarının yapılmasında öğrencilerin dersler arasında zaman kaybetmesinin önlenmesi için ardı sıra farklı dersler planlanabilmektedir [17].

Öğretim elemanlarının derslere atandırılması, daha sonra derslerin uygun zaman dilimlerine planlanması üniversitelerde yönetim kademesi tarafından her yarıyılın başında yerine getirilmesi gereken önemli görevlerdendir. Akademik ortam içinde bu planlamaları etkileyen örgütsel ve kişisel öncelikler planlamayı zorlaştırmaktadır. Ders seçimi esnasında öğrencilerin derslere olan taleplerinin değişkenlik göstermesi ve dersleri vermesi planlanan öğretim görevlilerinin isteklilik seviyesi yönetim kademesinin amaçlarını etkilemektedir. Açılması talep edilen derslerin fakültenin toplam ders yükünü geçmemesi, çizelgeleme için uygun sayıda zaman diliminin bulunması, derslik sayısı, full-time ve part-time öğretim elemanlarının sayısı problemin çözümünü etkileyen faktörler arasında yer almaktadır.

Gece ders planlamasının olup olmaması gibi genel yönetim politikaları da büyük öneme sahiptir. Zaman çizelgelerin hazırlanması, farklı seviyelerde farklı roller üstlenen birçok insanı ilgilendiren zor ve zaman alıcı işlemlerdir. İlköğretim okullarından üniversitelere farklı seviyelerde okullar için zaman çizelgeleri yapılmaktadır. Eğitim sistemlerinin ve okul kurallarının ülkeden ülkeye farklılıklar göstermesi de problemin çeşitliliğini artırmaktadır [18].

3.1 Ders Çizelgeleme Problemleri

Ders programı hazırlanması çizelgelemenin özel bir türü olup yöneylem araştırmasında; eğiticiler, öğrenciler, dersler ve dersanelerden oluşan kaynakların haftalık çizelge içindeki uygun zaman dilimlerine kaynaklar arasındaki kısıtlara bağımlı amaç fonksiyonunun iyileştirilmesi için çözülmeye çalışılan bir kaynak tahsis problemi olarak düşünülmektedir [9].

Ders çizelgeleri sadece birkaç genel prensip ile çözülemeyecek, karmaşık bir problem türüdür. Probleme dâhil olan tüm aktörlerin (yöneticiler, öğretim elemanları, öğrenciler) kendilerine özel amaçları vardır ve bu amaçlar genellikle birbiri ile çelişmektedir. Dersler, derslikler, zaman dilimleri, eğiticiler ve öğrenciler arasındaki karmaşık ilişkiler uygun bir çözüm bulabilmeyi zorlaştırmaktadır. Son otuz yıl içinde çeşitli çizelgeleme problemlerinin çözümü için birçok model ve yaklaşım önerilmiştir. Yöneylem araştırması, insan-makine etkileşimi, kısıt programlama, uzman sistemler veya sinir ağları problemin çözümü için kullanılmıştır. Fakat özel ihtiyaçlar ve kısıtlar sebebi ile problemin kurumdan kuruma değişmesi, problemin genel bir yapısının olmamasının çizelgeleme sisteminin gelişimini yavaşlatması, sistemle ilgili tüm bilgilere ve isteklere ulaşım bunları probleme dâhil etmenin zorluğu hala üzerinden durulması gereken konulardır [19].

Akademik yıl boyunca üniversitelerde idareciler, insan kaynakları yöneticileri ve ders programlarını planlayan personel öğrencilerin belirli kurallar altında gruplar haline getirilmeleri, öğretim elemanlarının ve öğrenci gruplarının derslere tahsis edilmesi, uygun zaman dilimlerinin ve dersliklerin de derslere tahsis edilmesi için

yoğun bir şekilde çaba göstermektedirler. Bu kararlar alınırken göz önünde bulundurulmuş kurallar ve kabuller; okulun yapısına, bölümlerin müfredatına, kaynakların kullanılabilirliğine ve çeşitli sosyal etkenlere bağımlı olarak kurumdan kuruma değişmektedir [20].

Ders programları da birçok kurumda hala elle yapılmaktadır. Bu amaçla tecrübeli bir personel orta dereceli bir kurumda yaklaşık bir hafta boyunca programı oluşturmak için çalışmaktadır. Genellikle hazırlanan program birçok personel tarafından özel istekleri ve pedagojik ihtiyaçları tam olarak karşılamadığı için tatmin edici bulunmamaktadır [21].

Bunun yanında problemin başlangıç şartlarında meydana gelebilecek değişimler problemi çözümsüzlüğe götürebilmektedir. Problem boyutları açısından NP-Zor olduğu için elle yapılan uygulamalarda sadece önemli görülen kısıtlar sağlanmaya çalışılmakta, fakat yine de problemin boyutları açısından uygun çözümlere ulaşmak çok uzun süreler almaktadır [22].

Ders çizelgeleme probleminde de her probleme özel kısıtlar olabileceği gibi, her problem için ortak olan ve sıkı kısıtlar olarak adlandırılan kısıtlarda mevcuttur. Aynı grupta bulunan öğrencilerin aynı zamanda iki farklı ders için planlanmaması, bir öğreticinin aynı zaman diliminde iki farklı ders için görevlendirilmemesi, bir zaman dilimi için planlanan toplam ders miktarının okulun mevcut derslik sayısından fazla olmaması ve bütün derslerin uygun herhangi bir zaman dilimine planlanması bu tür kısıtlara örnek olarak verilebilir [10].

Sıkı kısıtların ihlal edilmesi problemi çözümsüzlüğe götürmektedir. Bu sebeple uygun bir çözüm için bu kısıtların sağlanması gerekmektedir. Bunun yanında ihlal edilmesi istenmeyen fakat ihlal edilse dahi uygun çözümlere ulaşılmasına engel olmayan zayıf kısıtlar vardır. Bu kısıtların sağlanması çizelgenin kalitesini ve etkinliğini artırmaktadır. Bir öğreticinin bir gün içinde dörtten fazla ders istememesi, öğlen saatlerine ders planlanması, öğrencilerin günün erken ve geç saatlerinde ders istememesi ve iki parça halinde verilen bir ders için en az bir gün ara verilmesi gibi

kısıtlar, zayıf kısıtlara örnek olarak verilebilir [10]. Fakat gerçek hayat problemlerinin karmaşıklığı nedeniyle çoğu zaman zayıf kısıtları ihlal etmeden uygun bir çözüme ulaşmak mümkün olmamaktadır. Bu tür kısıtlar gevşetilerek uygun çözümler bulunabilir. Böylece bu kısıtlar amaç fonksiyonu içinde çözümün kalitesini belirlemek için kullanılabilir [23].

Ders programları hazırlanırken problemin aşırı kısıtlandırılmış olması ve problemin eniyileme kriterlerini belirlemenin zorluğu çözümü zorlaştırmaktadır. Ayrıca problemin dinamik olması; faaliyetlerin, kaynakların ve kısıtların tam olarak bilinmemesi veya son anda aniden değişmesi problemin karmaşıklığını artırmaktadır [24].

3.2 Çözüm Yaklaşımları

Eğitim öğretim kurumlarında oluşturulan çizelgelerin çözümü için farklı birçok yaklaşım geliştirilmiştir. Kurumdan kuruma farklı durumlar söz konusu olduğu için bir durum için uygun çözümler veren bir yaklaşımın başka şartlar altında iyi performans göstermemesi, problemin başlangıç safhasında hangi çözüm yaklaşımının daha etkin olduğu konusunda karar vermeyi zorlaştırmaktadır [10].

Eğitim öğretim kurumlarında yapılan ilk çizelgeleme çalışmalarında geleneksel elle yapılan çözüm uygulamaları bilgisayar ortamına aktarılmıştır. Çözüm metotlarında gelişimin olmadığı bu durumlarda sadece çizelgeleme çözüm süreci iyileştirilmiş, modern veri yapıları kullanılmamış ve bilgisayarlardan etkin bir şekilde yararlanılamamıştır [25]. Aşağıda çizelgeleme problemlerinin modellenmesi ve çözümü için geliştirilen yaklaşımların bir özeti verilmiştir.

Grafik Renklendirme Sezgiselleri çizelgeleme problemlerinin çözümünde kullanılan ilk yaklaşımlardandır. Önce ders çizelgeleme problemi grafik renklendirme problemine dönüştürülmektedir. Herhangi iki düğümü birbirine bağlayan bir bağlantı çizgisi olduğunda, iki düğüm farklı renklerde olacak şekilde, grafiğin düğümlerini renklendirme işlemi grafik renklendirme problemi olarak tanımlanmaktadır. Ders

çizelgelerinin hazırlanmasında dersler düğümlerle gösterilmektedir. Eğer iki dersin aynı anda atanmaması için bir kısıtlayıcı mevcutsa, bu iki dersi temsil eden düğümler arasında bir bağlantı bulunmaktadır. Bu tür bir kısıtlayıcı iki ders aynı öğretim elemanı tarafından verileceğinde veya aynı öğrenci grubu tarafından alınacağında görülmektedir. Grafik renklendirme, NP-Zor tipi bir problemdir ve literatürde ders çizelgeleme problemleri için birçok sezgisel yaklaşım önerilmiştir [25].

Tavlama Benzetim ve Tabu Arama çizelgeleme problemlerinin çözümü için geliştirilmiş, grafik renklendirme sezgisellerine ve matematiksel programlama yaklaşımlarına göre daha yeni tekniklerdir. Yerel bir en iyi noktaya yakalanmadan en iyi çözümün araştırılması için geliştirilmiş iteratif iyileştirme algoritmalarıdır [10].

Tavlama Benzetim tekniği, sıcak bir ortamda hareket halindeki atomların soğumasını taklit eden bir tekniktir. Yüksek sıcaklıkta atomlar serbest halde hareket ederken, sıcaklığın düşmesi sonucu atomlar arasındaki bağlar, atomları bir araya gelmeye zorlamaktadır. Belli bir seviyeden sonra atomlar hareket edememektedir. Kütle hızla soğutulduğunda elde edilen çözümün maliyeti, yavaş yavaş soğutulduğunda elde edilen maliyetten daha büyüktür.

Tavlama benzetim tekniğinin ders çizelgeleme problemlerine uyarlanması şu şekilde olmaktadır. Başlangıçta elemanlar zaman periyotlarına gelişigüzel atanmaktadır. Başlangıç ataması için ilk sıcaklık ve maliyet hesaplanmaktadır. Maliyet, çizelgenin kalitesini ölçmek için; sıcaklık ise maliyetteki artışı kontrol etmek için kullanılmaktadır. Her bir iterasyonda iki farklı zaman periyodu rastgele seçilip, bu iki zaman periyoduna önceden atanan elemanlar yer değiştirilerek değişen maliyet hesaplanmaktadır. Maliyet bir önceki maliyetten veya belirli bir ihtimalden daha küçükse kabul edilmektedir. Bu tekrarlı işlemler tüm periyotlara farklı elemanların yerleştirilmesini sağlamaktadır. İlk sıcaklıklarda yüksek maliyetli çözümlere izin verilmesi, yerel en iyiye yakalanmayı önlemektedir. Tavlama benzetim algoritmaları gelişigüzel permütasyonlarla çizelgeleme yaptıkları için her zaman gerçek en düşük maliyetli çözümü vermeyebilir ve farklı iki simülasyon aynı çözümü bulmayı garanti edemeyebilir [25].

Tabu Arama tekniğinin, ders çizelgeleme probleminin grafik renklendirme problemine dönüştürüldüğü durumlarda uygulamaları oldukça yaygındır. Ders çizelgeleme probleminin tabu arama tekniği ile çözümünde dersler öncelikle bir başlangıç tablosuna yerleştirilmektedir. Tabu arama tekniği başlangıç tablosunun geliştirilmesi için kullanılmaktadır. Çözüm uzayı kısıtları, çözüm değerleri; amaç fonksiyonu, kullanıcı tarafından belirlenen amaçların ağırlıklı toplamı olmak üzere çeşitli komşuluk yapıları tanımlanmaktadır. Basit komşuluk bir dersin bir periyottan diğer bir periyoda atanması ile elde edilmektedir. Değiştirme komşuluğu, iki dersin periyotlarının değiştirilmesi ile çoklu değişim komşuluğu ise ikiden fazla dersin yerlerinin değiştirilmesi ile elde edilmektedir. Çoklu değişim komşuluğu çözüm yapısında daha büyük değişikliklere izin vererek arama işleminin çözüm uzayının çok çeşitli bölgelerine ulaşmasına imkân tanımaktadır. Bu komşuluklar sonucu elde edilen yeni çözümün değeri bir önceki çözümün değerinden daha iyi ise kabul edilmektedir. Tabu listesi ise her bir iterasyon için değiştirilen dersi ve periyodu saklamaktadır. Eğer incelenen hareket, amaç fonksiyonunun mevcut en iyi çözümünden daha iyi bir çözüm ise hareket tabu listesinde olsa bile gerçekleştirilebilmektedir. Tabu aramanın bu tekrarlı prosedürü amaç fonksiyonunda bir gelişme olmayıncaya kadar uygulanmaktadır.

Genetik Algoritmalar da eğitim kurumlarında çizelgeleme problemlerinin çözümü için kullanılan algoritmalarındandır. Genetik algoritmalar doğadaki evrim mekanizmasını örnek olarak alan ve en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üreten yaklaşımlardır. Bu amaçla iyinin ne olduğunu belirleyen bir uygunluk fonksiyonu, yeni çözümler üretmek için yeniden üretim ve değiştirme operatörleri kullanılmaktadır. Genetik algoritmalar çözümü bulmak için taranması gereken parametre uzayının çok büyük olduğu durumlarda etkin çözümler vermektedir. Evrimin her sürecinde elde edilen bilgi sonraki nesillere aktarılarak taramanın daha uygun bölgelerde yapılması sağlanmakta ve değişim işlemleri yardımı ile yerel çözüm noktalarına sıkışılması önlenmektedir [25].

Kısıtlayıcı Programlama büyük boyutlu çizelgeleme problemlerinin çözümünde kullanılan yeni yaklaşımlardandır. Bu yaklaşımda çizelgeleme problemi kısıt tatmin problemi olarak modellenmekte ve üç ana elemandan oluşmaktadır:

- Değişkenler
- Her bir değişkenin alabileceği değerler seti
- Değişkenlerin aynı zamanda alabileceği değerleri sınırlayan kısıtlar

Okul çizelgeleme problemlerinde, kısıt programla tekniği ile çözümlerde genellikle dersler ve sınavlar değişkenler olarak tanımlanmaktadır. Tipik çizelgeleme kısıtları da kısıt setini oluşturmaktadır. Problem, değişkenlere bütün kısıtları sağlayan değerler atandırılmasıdır. Problemin eniyileme modelinde kısıtların sağlanmasının yanında amaç fonksiyonunun en küçüklenmesi veya en büyüklenmesi amaçlanmaktadır.

Çizelgelerin oluşturulmasında kısıt programlama tekniğinin tercih edilmesinin birçok sebebi vardır. Örneğin bu tür problem çözümlerinde iki değişkenin aynı anda alabileceği değerleri sınırlayan birçok kısıt mevcuttur. Kısıt programlama, kısıtların üretilmesinde bu olayı kullanmaktadır.

Kısıt üretilmesi, çözüm sürecinde mevcut kısıt setine bağımlı olarak değişkenlerin alabileceği değerler için yeni kısıtlar üreten bir mekanizmadır. Kısıt programlamanın bu tür özellikleri değişkenlere değerler atarken arama uzayının küçültülmesine olanak tanımaktadır. Ders ve sınav sayılarının çok fazla olduğu, dolayısıyla çözüm uzayının çok geniş olduğu problemlerin çözümünün çok uzun zaman almasından dolayı bu tür yaklaşım oldukça önemlidir. Çözüm uzayının küçültülmesinin yanında arama stratejisinin de belirlenebilme olasılığı kısıt programlamayı daha çekici hale getirmektedir [10].

Şebeke Akış Yaklaşımları ise genellikle küçük boyutlu problemlerin çözümünde kullanılan bir yaklaşımdır. Şebeke akış yaklaşımında ders çizelgeleme problemi derslerin öğretim elemanlarına atandırılması ile başlamaktadır. Öğretim elemanlarının istekleri, dersane durumlarına bakılarak karşılanmaya çalışılmaktadır. Şebeke modelinde beş çeşit düğüm olduğu kabul edilmektedir. Birinci ve besinci düğümler kaynak ve tüketim düğümleri olarak kabul edilmekte ve şebekenin akısını sağlamaktadır. Bu iki düğüm arasında bölüm seviyesi, konu seviyesi ve dersane boyutu/zaman seviyesi olmak üzere üç seviyede düğüm olduğu kabul edilmektedir. Son yıllarda yapılan çalışmalarda birkaç yaklaşımın birleştirildiği melez çözüm yöntemleri ile etkin çözümlere ulaşılmaktadır [25].

3.3 Eğitim Öğretim Kurumlarında Çizelgeleme Problemi Üzerine Çalışmalar

Çizelgeleme problemleri çok farklı alanlarda, farklı kısıtlara ve önceliklere sahip problemlerin çözümünde kullanıldıkları için, literatürde birçok yaklaşımın kullanıldığı görülmektedir. Tek bir yaklaşımla problemin çözümüne ulaşılmaya çalışıldığı gibi, çözümün geliştirilmesi ve daha etkin çözümlerin bulunabilmesi amacıyla iki ve ikiden fazla yaklaşımların birleştirilerek melez yaklaşımlarla da problem çözülmeye çalışılmıştır.

Adamidis ve Arapakis, genetik algoritmalar yardımıyla gerçek bir çizelgeleme problemi üzerine çalışmışlardır. İki farklı genetik algoritmayı karşılaştırmışlardır. Ayrıca mutasyona izin veren, mutasyona izin vermeyen ve değişik kombinasyon ve mutasyon operatörlerine sahip üç farklı grubun performanslarını karşılaştırmışlardır. Sonuç olarak genetik algoritmaların her zaman sıkı kısıtları sağlayarak uygun çözümler üretebildiklerini göstermişlerdir [26].

Schaerf, okullarda ders çizelgeleme problemini haftalık olarak derslerin uygun zaman dilimlerine atandırılması olarak tanımlamış, bir öğretmenin aynı zaman diliminde başka bir derse atanmaması ve bunun yanında diğer kısıtların da sağlanması olarak problemin amacını belirtmiştir. Yaptığı çalışmada yerel arama tekniğine dayalı bir çözüm algoritması geliştirmiştir. Algoritma sıkı kısıtların uygun

bir şekilde gevşemesine izin veren yapısı ile önceden geliştirilen çözüm yöntemlerine alternatif olmaktadır [14].

Abbas ve Tsang, okul ve üniversite çizelgeleme problemi için kısıt tatmin tekniklerinin bir uygulamasını örnek bir durum çalışması üzerinde göstermiştir. Yazılım mühendisliğinin kısıt tatmin teknikleri üzerine kullanımı üzerinde durmuşlardır. Geliştirdikleri yazılım ile problemin yazılım mühendisleri tarafından daha rahat anlaşılmasını sağlamışlardır [27].

Blum ve ark., bir çizelge oluşturucuya çözüme ulaşabilmesi için yollar gösteren genetik bir algoritma geliştirmişler ve önerdikleri yolları daha önce kullanılmış sezgiseller ile karşılaştırmışlardır [28].

Tam ve Ting, genel üniversite çizelgeleme problemlerinin etkili çözümü için yerel arama metotlarında kullanılan en az çakışma sezgiselleri ile ileri bakış sezgisellerini birleştirmişlerdir [29].

Sigl ve ark., ders çizelgeleme problemi için genetik bir algoritma geliştirmişler, küçük ve büyük ölçekli problemlerin çözümü için geliştirdikleri algoritmayı kullanmışlardır. Ana genetik operatörün geliştirilmesi ile algoritmanın performansı iyileştirilmiştir. Akıllı operatörler yardımıyla yeni çakışmaların oluşması önlenmiş ve algoritmanın tüm davranışları geliştirilmiştir [30].

Mahdi ve ark., üniversite çizelgeleme probleminin çözümü için tasarlanan uyumlu parametre kontrolü ile genetik bir algoritma sunmuşlardır. Genetik algoritmanın parametre setlerinin kontrol edilmesi ile çizelgenin kalitesinin artırılabilceğini göstermeyi amaçlamışlardır [31].

Daskalaki ve ark., 0-1 tamsayı programlama formülasyonu ile üniversite çizelgeleme problemi üzerine çalışmışlardır. Kurdukları modelin birçok akademik kurum ve kuruluş için geçerli olan ihtiyaçları ve operasyonel ihtiyaçları sağlayan kısıtları sağladığını göstermişlerdir. Modelin amacı, doğrusal amaç fonksiyonunu en

küçükleme olmuştur. Personel tarafından belirtilen belli günlerin belli ders periyotlarına planlanması, bazı dersler için özel sınıfların tahsis edilmesi gibi isteklerin karşılanması ile amaç fonksiyonu iyileştirilmektedir. Ayrıca amaç fonksiyonundaki maliyet katsayılarının uygun bir şekilde tanımlanması ile çözüm uzayının daraltılabileceğini ve çözüme daha rahat ulaşılabileceğini göstermişlerdir [11].

Dimopoulou ve Miliotis, üniversitelerde ders çizelgeleme oluşturulmasına yardımcı olabilecek bilgisayar ağlarına dayalı bir sistemin tasarımı ve uygulaması üzerine çalışmışlardır. Önerilen sistem bütün uygun verileri bulunduran merkezi bir veri tabanını kullanarak hesaplama platformu üzerinde çalışmaktadır. Üniversitenin farklı bölümleri veritabanına girerek gerekli bilgileri güncelleyebilmekte, çizelgeleme ile uğraşan personel güncellenen veriler ile her bölüm için oluşturulan yeni çizelgeleri bölümlere gönderebilmektedir. Tüm bölümler için oluşturulan çizelgelemelere ulaşmak mümkündür. Sistem her bölüm için dersleri uygun zaman aralıklarına ve uygun dersliklere atamak için tamsayı programlama modeli kullanmaktadır. Aynı zamanda her bir bölümün verileri ile tamsayı programlama modelini birbirine bağlayan ve işlemler esnasında meydana gelen çakışmaları çözen otomatik bir prosedür de geliştirmişlerdir [32].

Piechowiak ve Kolski, yaptıkları çalışmada çizelgeleme yönetimi için interaktif bir karar destek sistemi tasarlamışlardır. Hiyerarşik veri organizasyonunu dikkate alarak bu verilere dayalı uyumlu kısıtlar geliştirmişlerdir. Yaptıkları çalışmada üç amaca odaklanmışlardır. Birinci amaçları açık, genetik bir araç geliştirmek olmuştur. Bu amaçla hedef sınıflar üzerine odaklanan amaca yönelik bir yaklaşım takip etmişlerdir. İkinci amaçları problem tanımlarında meydana gelebilecek değişimleri göz önünde bulundurarak farklı yapılarda kullanılacak genetik bir organizasyon kurmak, üçüncü amaçları ise kişisel bilgisayarlarda değişik kullanıcılar tarafından kolaylıkla kullanılacak bir sistem oluşturmak olmuştur [33].

Parthiban ve ark. Fakültelerde ders çizelgelemesinin tesisler, uygun zaman dilimleri, derslikler, öğretmenler ve öğrencilerin birleştirilmesi açısından genel okul çizelgeleme problemlerinden daha karmaşık olduğunu belirtmişler ve özellikle personel tarafından belirtilen isteklerin problem çözümünü daha da zorlaştırdığını vurgulamışlardır. Bu tür bir problemin çözümü için personel önceliklerini ve isteklerini dikkate alan karar destek yaklaşımına dayalı bir analitik hiyerarşik proses tekniği geliştirmişlerdir [34].

Bhatt ve Sahajpal, hesaplama açısından NP-Zor kategorisine giren çizelgeleme problemlerinin yıllardan beri elle çözülmeye çalışıldığını, zaman içinde bilgisayar destekli deterministik metotların geliştirildiğini, daha iyi sonuçlar veren genetik algoritmalar gibi yaklaşımların kullanıldığını belirtmişler ve yaptıkları çalışmada araştırmanın kalitesini yükselten yeni ve etkin melez bir genetik bir algoritma sunmuşlardır [22].

Daskalaki ve Birbas, üniversite çizelgeleme problemleri gibi büyük boyutlu ve karmaşık problemlerin çözümü için tamsayı programlama tekniğinin her zaman bir alternatif olduğunu, fakat karmaşık operasyonel kuralları modellemek için gerekli olan yoğun çaba ve gerçek problemlerin boyutlarının büyüklüğünden kaynaklanan hesaplama zorluklarının araştırmacıların sevklerini kırdığını ve onların isteklerini diğer yaklaşımlara yönelttiğini belirtmiştir. Yaptıkları çalışmada üniversite çizelgeleme problemini için geliştirdikleri etkin bir tamsayı programlama formülasyonunu iki basamaklı gevşeme yaklaşımı ile çözmüşlerdir. Aynı problemi tek aşamada çözen diğer yaklaşımlar ile yapılan karşılaştırmalarda çözümün kalitesinde herhangi bir azalma olmadan çözüm süresinin oldukça kısaldığını göstermişlerdir [20].

Zhang ve Lau, üniversite çizelgeleme problemi için kısıt programlama modeli geliştirmişlerdir. Kısıt programlama modeli çözümü bütün kısıtları sağlayacak şekilde bütün değişkenlerin belli değerlere atandırılmasıdır. Bu amaçla örnek bir durum üzerinde ILOG çizelgeleyici ve ILOG çözücü kullanmışlardır.

Geliştirdikleri kısıt programlama modelinin performansını farklı amaçlar için ILOG çözücü üzerinde test etmişlerdir [9].

Cambazard ve ark., aşırı kısıtlandırılmış, dinamik çizelgeleme problemi üzerine çalışmışlardır. Problemin dinamik yapısını dikkate alan ve otomatik olarak kısıtların gevşemesine izin veren bir kısıt programlama modeli geliştirmişlerdir [24].

Santiago-Mozos ve ark., bir İspanyol üniversitesinde bir durum çalışması için kişiselleştirilmiş çizelgeler elde etmek amacıyla iki aşamalı genetik bir sezgisel algoritma geliştirmişlerdir. Birinci aşamada kullanılan sezgisel yaklaşım ile öğrencilerinin isteklerini birinci önem derecesinde göz önünde bulundurarak öğrenciler gruplara ayrılmış ve ilk atamalar yapılmış, ikinci aşamada kullanılan genetik sezgisel yaklaşım ile birinci aşamada elde edilen çizelge geliştirilmiştir [35].

Lai ve ark., ders çizelgeleme problemlerinin birkaç genel prensip ile çözülemeyecek, karmaşık problemler olduğunu; yönetim, eğitici ve öğrencilerin birbirlerinden farklı ve çoğunlukla birbirleri ile çelişen amaçları olduğunu belirtmişlerdir. Ders zamanları, derslikler, dersler ve eğiticiler arasındaki karmaşık ilişkilerin çözümü zorlaştırdığını vurgulamışlardır. Yaptıkları çalışmada uzman sistemler ile kısıt programlamayı birleştiren bir yapay zekâ yaklaşımı ile ders çizelgeleme problemi üzerine çalışmışlardır. Uzman sistemlerdeki ayrıştırmanın değişimlere karşı esnekliği artırdığını göstermişler, zorunlu ve zayıf kısıtlar arasında bir hiyerarşi oluşturarak, kısıt tatmin seviyelerini ve rahatlama tekniklerini kullanarak zorunlu ve zayıf kısıtları ele almışlar ve amaca yönelik yazılım mühendisliğini çizelgelemenin iyileştirilmesi için kullanmışlardır [19].

Beligiannis ve ark., genetik hesaplama tekniklerine dayalı uyumlu bir algoritma tasarlamış, geliştirmiş ve bu algoritmayı eğitim öğretim kurumlarında çizelgeleme problemi için kullanmışlardır. Algoritma öncelikle Yunanistan'da bulunan yüksek okullarda çözüm elde etmek amacıyla kullanılmış, daha sonra algoritmanın etkinliğini ve üstünlüğünü kanıtlamak amacıyla dünyanın diğer bölgelerinden elde edilen veriler kullanılarak farklı yöntemlerle oluşturulan çizelgeler ile karşılaştırmalar yapılmıştır [36].

Al-Yakoob ve Sherali, Kuveyt Üniversitesinde yeni uygulamaya konulan cinsiyet farklılığı ile ilgili uygulamaları dikkate alarak, ders programlarındaki çakışmaları önlemek ve üniversitenin kaynaklarının en faydalı şekilde kullanılabilmesini sağlayabilmek amacıyla karışık bir tamsayı programlama modeli geliştirmişlerdir [37].

Head ve Shaban, akademik çevrelerde ders çizelgeleme problemi üzerinde birçok çalışma yapıldığını fakat bu çalışmalarda öğrenci çizelgelemesi ile ders çizelgelemesinin ayrı ayrı incelendiğini belirtmişlerdir. Geliştirdikleri sezgisel yaklaşım ile öğrencilerin ve derslerin aynı anda çizelgenmesini sağlamışlardır. Çizelge oluşturulduktan sonra diğer bir sezgisel yaklaşımla çözümü iyileştirmeye çalışmışlardır [38].

Yukarıda verilen literatür incelemesi bir özet halinde Çizelge 3.1’de sunulmuştur.

Çizelge 3. 1 Literatür İncelemesinin Genel Gösterimi

Yazar	Yılı	Yöntem	Çalışma Alanı
Adamidis P., Arapakis P.	1999	Genetik Algoritmalar	Ders Çizelgeleme
Schaerf, A.	1999	Yerel Arama Tekniği	
Abbas, A.M., Tsang, E.P.K.	2001	Kısıt Tatmin Teknikleri	
Blum, C., Correia, S., Paechter, B., Rossi-Doria, O., Snoek, M.	2002	Genetik Algoritmalar	
Tam V., Ting D.	2003	Sezgisel Yaklaşım	
Sigl B., Golub M., Mornar V.	2003	Genetik Algoritmalar	
Mahdi, O., Aion, R.N., Zainuddin, R.	2003	Genetik Algoritmalar	
Daskalaki S., Birbas T., Housos E.	2004	Tamsayı Programlama	
Dimopoulou M., Miliotis P.	2004	Tamsayı Programlama	
Piechowiak S., Kolski C.	2004	Karar Destek Sistemi	
Parthiban, P., Ganesh, K., Narayanan, S., Dhanalakshmi, R.	2004	Karar Destek Yaklaşımı	
Bhatt, V., Sahajpal R.	2004	Genetik Algoritmalar	
Zhang L., Lau S.K.	2005	Kısıt Tatmin Teknikleri	
Cambazard H., Demazeau F., Jussien N., David P.	2005	Kısıt Programlama	
Santiago-Mozos R., Salcedo-Sanz S., DePrado-C. M., Bousoño-Calzón C.	2005	Genetik Sezgisel Algoritmalar	
Santos H.G., Ochi L.S., Sauza M.J.F.	2005	Tabu Arama Sezgiseli	
MirHassani S.A.	2006	Tamsayı Programlama	
Lai L., Hsueh N., Huang L., Chen T.	2006	Yapay Zeka Yaklaşımı	
Ciscon L. A., Oliveira H.C.B.	2006	Genetik Algoritmalar	
Beligiannis G.N., Moschopoulos C.N., Kaperonis G.P., Likothanassis S.D.	2006	Genetik Algoritma	
Al-Yakoob S.A., Sherali H.D.	2007	Tamsayı Programlama	
Head C., Shaban S.	2007	Sezgisel Yaklaşım	

4. UYGULAMA

Bu bölümde kısıtlayıcı programlama mantığı kullanılarak geliştirilmiş olan “Ders Çizelge Oluşturma Sistemi (DCOS)” uygulama yazılımını anlatılacaktır. Kısıt olarak 2 zorunlu, 1 zayıf kısıt belirlenmiştir. Zorunlu kısıtlar, günü ve saati sabit olan dersler ve bir dersin verildiği saat içerisinde başka bir veya birden fazla dersle çakışmamasıdır. Zayıf kısıt ise dışarıdan görevlendirme ile gelen hocaların verdiği ders için gün ve saat önerisinde bulunmasıdır. Uygulamada öğrencilerin alttan veya üstten aldığı derslerle kendi sınıflarından aldığı derslerin çakışmama kısıdı göz ardı edilmiştir. Çünkü üniversitemizde ders programları ders seçimlerinden önce hazırlanmaktadır.

Biçimsel olarak zaman çizelgeleme problemleri, “V, D ve C” olmak üzere üç ana kümeyle tanımlanabilir:

$$V = \{v_1, v_2, \dots, v_M\}$$

$$D = \{d_1, d_2, \dots, d_M\}$$

$$C = \{c_1, c_2, \dots, c_K\}$$

V kümesi *değişkenler* (variables) kümesidir. D kümesi değişkenlerin *tanım kümelerini* (domains of variables) ifade eder. Zaman çizelgeleme problemlerinde D kümesi en az bir eleman içerir, o da zaman (time) kümesidir. C kümesi de V kümesindeki her bir değişkenin alabileceği değerler için tanımlanmış *kısıtlardır* (constraints). İki tür kısıt vardır: *zorunlu* ve *zayıf*. Zorunlu kısıtlar sağlanması zorunlu olan kısıtları, zayıf olanlar da tercihleri temsil eder.

4.1 Uygulama Veritabanı Modeli

Uygulamada veritabanı olarak MS SQL Server 2008 kullanılmıştır. Veritabanımızda 12 tablo bulunmaktadır. Bunlar; fakülte, bölüm, dersler, derslik, akademisyen, akademisyen gün saat, bölüm ders akademisyen, çakışma olmayacak dersler, sabit dersler, ders program, servis bilgi, admin data.

Fakülte tablosunun yapısı Çizelge 4.1’de gösterilmiştir. Oluşturmuş olduğumuz ders programını belirli bir isim altında kaydetmek için “arşiv adı”, fakülte isimlerinin tutulduğu “fakülte adı”, her bir fakülteye ait bölümlere şifre belirleyebilmek için “şifre bölümde olsun”, her bir fakülteye ait “şifre” gibi alanlar bulunmaktadır. Ayrıca “fakulte_id” ve “arsiv_adi” alanları birincil anahtardır.

Çizelge 4. 1 Fakülte Tablo Yapısı

	Column Name	Data Type	Allow Nulls
🔑	fakulte_id	bigint	<input type="checkbox"/>
🔑	arsiv_adi	nvarchar(50)	<input type="checkbox"/>
	fakulte_adi	nvarchar(50)	<input checked="" type="checkbox"/>
	sifre_bolumde_olsun	tinyint	<input checked="" type="checkbox"/>
	sira	int	<input checked="" type="checkbox"/>
	onay	tinyint	<input checked="" type="checkbox"/>
	sifre	nvarchar(50)	<input checked="" type="checkbox"/>

Bölüm tablosunun yapısı Çizelge 4.2’de gösterilmiştir. Oluşturmuş olduğumuz ders programını belirli bir isim altında kaydetmek için “arşiv adı”, bölüm isimlerinin tutulduğu “bölüm adı”, her bir fakülteye ait “fakülte id”, her bir bölüme ait “şifre”, bölümlere ait “id” gibi alanlar bulunmaktadır.

Çizelge 4. 2 Bölüm Tablo Yapısı

Column Name	Data Type	Allow Nulls
id	bigint	<input checked="" type="checkbox"/>
fakulte_id	bigint	<input checked="" type="checkbox"/>
arsiv_adi	nvarchar(50)	<input checked="" type="checkbox"/>
bolum_adi	varchar(MAX)	<input checked="" type="checkbox"/>
sira	int	<input checked="" type="checkbox"/>
onay	tinyint	<input checked="" type="checkbox"/>
sifre	nvarchar(50)	<input checked="" type="checkbox"/>

Dersler tablosunun yapısı Çizelge 4.3’de gösterilmiştir. Oluşturmuş olduğumuz ders programını belirli bir isim altında kaydetmek için “arşiv adı”, ders isimlerinin tutulduğu “ad”, her bir derse ait “ders kodu”, her bir derse ait “id”, derslerin hangi bölümler tarafından açıldığını gösteren “açan bölüm”, derslere ait “teori saati” ve “uygulama saati” gibi alanlar bulunmaktadır.

Çizelge 4. 3 Dersler Tablo Yapısı

Column Name	Data Type	Allow Nulls
id	bigint	<input checked="" type="checkbox"/>
arsiv_adi	nvarchar(50)	<input checked="" type="checkbox"/>
ders_kodu	varchar(50)	<input checked="" type="checkbox"/>
ad	varchar(MAX)	<input checked="" type="checkbox"/>
teori_saati	tinyint	<input checked="" type="checkbox"/>
uygulama_saati	tinyint	<input checked="" type="checkbox"/>
acan_bolum	int	<input checked="" type="checkbox"/>

Derslik tablosunun yapısı Çizelge 4.4’de gösterilmiştir. Oluşturmuş olduğumuz ders programını belirli bir isim altında kaydetmek için “arşiv adı”, derslik isimlerinin tutulduğu “şube adı”, dersliklere ait “kontenjan”, dersliklerin teorimi yoksa laboratuvar mı olduklarını belirten “derslik tipi”, dersliklerin hangi fakülte ya da bölümler tarafından açıldığını gösteren “öncelik fakülte id, öncelik bölüm id”, dersliklerin sahip oldukları özellikleri gösteren ”özellik” gibi alanlar bulunmaktadır.

Çizelge 4. 4 Derslik Tablo Yapısı

Column Name	Data Type	Allow Nulls
id	bigint	<input checked="" type="checkbox"/>
oncelik_fakulte_id	int	<input checked="" type="checkbox"/>
oncelik_bolum_id	int	<input checked="" type="checkbox"/>
fakulte_id	int	<input checked="" type="checkbox"/>
arsiv_adi	nvarchar(50)	<input checked="" type="checkbox"/>
sube_kodu	varchar(50)	<input checked="" type="checkbox"/>
sube_adi	varchar(50)	<input checked="" type="checkbox"/>
kontenjan	int	<input checked="" type="checkbox"/>
bagli_olunan_sube	varchar(50)	<input checked="" type="checkbox"/>
acan_birim	varchar(50)	<input checked="" type="checkbox"/>
derslik_tipi	int	<input checked="" type="checkbox"/>
onem_seviyesi	tinyint	<input checked="" type="checkbox"/>
ozellik	nvarchar(80)	<input checked="" type="checkbox"/>

Akademisyen tablosunun yapısı Çizelge 4.5’de gösterilmiştir. Oluşturmuş olduğumuz ders programını belirli bir isim altında kaydetmek için “arşiv adı”, akademisyen isimlerinin tutulduğu “ad, soyad”, akademisyenlere ait unvan bilgisinin yer aldığı “unvan”, her bir akademisyene ait “id” akademisyenlerin sahip oldukları “önem seviyesi”, akademisyenlerin boş oldukları günleri belirten “boş gün”, akademisyenlerin kadrolu mu yoksa ders saat ücretli mi olduklarını belirten ”kadrolu”, akademisyenlerin gün içerisinde verebilecekleri maksimum ders sayısını gösteren “günlük maksimum sayı” gibi alanlar bulunmaktadır.

Çizelge 4. 5 Akademisyen Tablo Yapısı

Column Name	Data Type	Allow Nulls
id	bigint	<input checked="" type="checkbox"/>
arsiv_adi	nvarchar(50)	<input checked="" type="checkbox"/>
ad	varchar(50)	<input checked="" type="checkbox"/>
soyad	varchar(50)	<input checked="" type="checkbox"/>
unvan	varchar(50)	<input checked="" type="checkbox"/>
cinsiyet	varchar(50)	<input checked="" type="checkbox"/>
onem_seviyesi	tinyint	<input checked="" type="checkbox"/>
bos_gun	tinyint	<input checked="" type="checkbox"/>
gunluk_max_sayi	tinyint	<input checked="" type="checkbox"/>
kadrolu	tinyint	<input checked="" type="checkbox"/>

Akademisyen gün ve saat tablosunun yapısı Çizelge 4.6’da gösterilmiştir. Oluşturmuş olduğumuz ders programını belirli bir isim altında kaydetmek için “arşiv adı”, akademisyenlere ait “akademisyen id”, akademisyenlerin ders vereceği gün bilgisinin yer aldığı “gün”, akademisyenlerin ders verecekleri günlere ait saat bilgisinin tutulduğu “saat baş, saat bit” gibi alanlar bulunmaktadır.

Çizelge 4. 6 Akademisyen Gün ve Saat Tablo Yapısı

Column Name	Data Type	Allow Nulls
akademisyen_id	bigint	<input checked="" type="checkbox"/>
arsiv_adi	nvarchar(50)	<input checked="" type="checkbox"/>
gun	tinyint	<input checked="" type="checkbox"/>
saat_bas	tinyint	<input checked="" type="checkbox"/>
saat_bit	tinyint	<input checked="" type="checkbox"/>






Bölüm ders akademisyen tablosunun yapısı Çizelge 4.7’de gösterilmiştir. Oluşturmuş olduğumuz ders programını belirli bir isim altında kaydetmek için “arşiv adı”, her bir şube için “şube kodu”, bölümlere ait “bölüm id”, derslere ait “ders id”, akademisyenlere ait “akademisyen id”, derslerin hangi bölümler tarafından açıldığını gösteren “açan bölüm id”, derslerin buldukları küme aralıklarını gösteren “küme” hangi dersin kaçınıcı sınıfa ait olduğunu gösteren “sınıf”, her bir şubeye ait “kontenjan”, akademisyenlerin önem seviyelerini gösteren “önem seviyesi”, derslerin verileceği saat aralıkları “ilk ders saati, son ders saati”, derslerin verileceği gün bilgisi “ders günleri”, derslerin verileceği saat aralıklarına ait 75 dakika özelliği “dakka 75”, oluşturmuş olduğumuz ders programını görüntülemek için “program_hazirla” gibi alanlar bulunmaktadır.

Çizelge 4. 7 Bölüm Ders Akademisyen Tablo Yapısı

Column Name	Data Type	Allow Nulls
potansiyel_id	int	<input checked="" type="checkbox"/>
sube_id	int	<input checked="" type="checkbox"/>
bolum_id	bigint	<input checked="" type="checkbox"/>
akademisyen_id	bigint	<input checked="" type="checkbox"/>
acan_bolum_id	int	<input checked="" type="checkbox"/>
ders_id	bigint	<input checked="" type="checkbox"/>
derslik_id	int	<input checked="" type="checkbox"/>
arsiv_adi	nvarchar(50)	<input checked="" type="checkbox"/>
sube_tipi	int	<input checked="" type="checkbox"/>
sube_kodu	nvarchar(50)	<input checked="" type="checkbox"/>
kume	nvarchar(5)	<input checked="" type="checkbox"/>
sinif	int	<input checked="" type="checkbox"/>
kontenjan	int	<input checked="" type="checkbox"/>
onem_seviyesi	tinyint	<input checked="" type="checkbox"/>
ilk_ders_saati	tinyint	<input checked="" type="checkbox"/>
son_ders_saati	tinyint	<input checked="" type="checkbox"/>
max_ders_sayi	int	<input checked="" type="checkbox"/>
min_ders_sayi	int	<input checked="" type="checkbox"/>
ders_gunleri	nvarchar(50)	<input checked="" type="checkbox"/>
dakka75	tinyint	<input checked="" type="checkbox"/>
program_hazirla	tinyint	<input checked="" type="checkbox"/>






Çakışma olmayacak dersler tablosunun yapısı Çizelge 4.8’de gösterilmiştir. Oluşturmuş olduğumuz ders programını belirli bir isim altında kaydetmek için “arşiv adı”, her bir bölüme ait tutulan “bölüm id”, çakışma olmaması istenen dersler için tutulan “ders id 1, ders id 2, sınıf” gibi alanlar bulunmaktadır. Ayrıca “arsiv_adi”, “bolum_id”, “ders_id_1”, “ders_id_2” ve “sınıf” alanları birincil anahtardır.

Çizelge 4. 8 Çakışma Olmayacak Dersler Tablo Yapısı

	Column Name	Data Type	Allow Nulls
	bolum_id	bigint	<input type="checkbox"/>
	ders_id_1	bigint	<input type="checkbox"/>
	ders_id_2	bigint	<input type="checkbox"/>
	arsiv_adi	nvarchar(50)	<input type="checkbox"/>
	sınıf	int	<input type="checkbox"/>

Sabit dersler tablosunun yapısı Çizelge 4.9’da gösterilmiştir. Oluşturmuş olduğumuz ders programını belirli bir isim altında kaydetmek için “arşiv adı”, bölümlere ait tutulan “bölüm id”, sabit olarak yerleştirilecek dersler için tutulan “bölüm id, ders id, gün, saat”, yerleştirilecek dersin tipini göstermek için tutulan “is teori” gibi alanlar bulunmaktadır. Ayrıca “arsiv_adi”, “bolum_id”, “ders_id”, “gun” ve “saat” alanları birincil anahtardır.

Çizelge 4. 9 Sabit Dersler Tablo Yapısı

	Column Name	Data Type	Allow Nulls
	bolum_id	bigint	<input type="checkbox"/>
	ders_id	bigint	<input type="checkbox"/>
	arsiv_adi	nvarchar(50)	<input type="checkbox"/>
	gun	tinyint	<input type="checkbox"/>
	saat	tinyint	<input type="checkbox"/>
	is_teorı	tinyint	<input checked="" type="checkbox"/>

Ders program tablosunun yapısı Çizelge 4.10’da gösterilmiştir. Bu tablo programlanmış ders planını göstermektedir. Bu tablo içerisinde; oluşturmuş olduğumuz ders programını belirli bir isim altında kaydetmek için “arşiv adı”, bölümlere ait tutulan “bölüm”, dersler için tutulan “ders”, akademisyenler için “akademisyen”, derslikler için “derslik” gibi alanlar bulunmaktadır.

Çizelge 4. 10 Ders Program Tablo Yapısı

Column Name	Data Type	Allow Nulls
potansiyel_id	int	<input checked="" type="checkbox"/>
derslik_hucre_id	nvarchar(80)	<input checked="" type="checkbox"/>
dakka75_hucre_id	nvarchar(80)	<input checked="" type="checkbox"/>
dakka75_hucre_id2	nvarchar(80)	<input checked="" type="checkbox"/>
arsiv_adi	nvarchar(50)	<input checked="" type="checkbox"/>
ders_tipi	tinyint	<input checked="" type="checkbox"/>
bolum	int	<input checked="" type="checkbox"/>
ders	int	<input checked="" type="checkbox"/>
akademisyen	int	<input checked="" type="checkbox"/>
derslik	int	<input checked="" type="checkbox"/>
gun	int	<input checked="" type="checkbox"/>
saat	int	<input checked="" type="checkbox"/>
sinif	int	<input checked="" type="checkbox"/>

Servis bilgi tablosunun yapısı Çizelge 4.11’de gösterilmiştir. Bu tablo sayesinde kullanıcı gerekli alanlar içerisine yazacağı SQL komutları sayesinde, ilgili veritabanına bağlanıp, gerekli bilgilerin temin edilmesini sağlayacaktır. Bu tablo içerisinde; ilgili sunucuya ait isim bilgisinin tutulduğu “server name”, ilgili sunucu içerisinde yer alan veritabanının ismi “database name”, ilgili veritabanına ait kullanıcı bilgilerinin tutulduğu “user id, pass”, ilgili veritabanı içerisinde yer alan tablolardan gerekli bilgileri elde etmek için tutulan “akademisyen, ders, fakülte, bölüm, bölüm ders akademisyen” gibi alanlar bulunmaktadır.

Çizelge 4. 11 Servis Bilgi Tablo Yapısı

Column Name	Data Type	Allow Nulls
user_id	nvarchar(50)	<input checked="" type="checkbox"/>
server_name	nvarchar(50)	<input checked="" type="checkbox"/>
database_name	nvarchar(50)	<input checked="" type="checkbox"/>
pass	nvarchar(50)	<input checked="" type="checkbox"/>
akademisyen	nvarchar(MAX)	<input checked="" type="checkbox"/>
ders	nvarchar(MAX)	<input checked="" type="checkbox"/>
fakulte	nvarchar(MAX)	<input checked="" type="checkbox"/>
bolum	nvarchar(MAX)	<input checked="" type="checkbox"/>
blm_ders_adk	nvarchar(MAX)	<input checked="" type="checkbox"/>

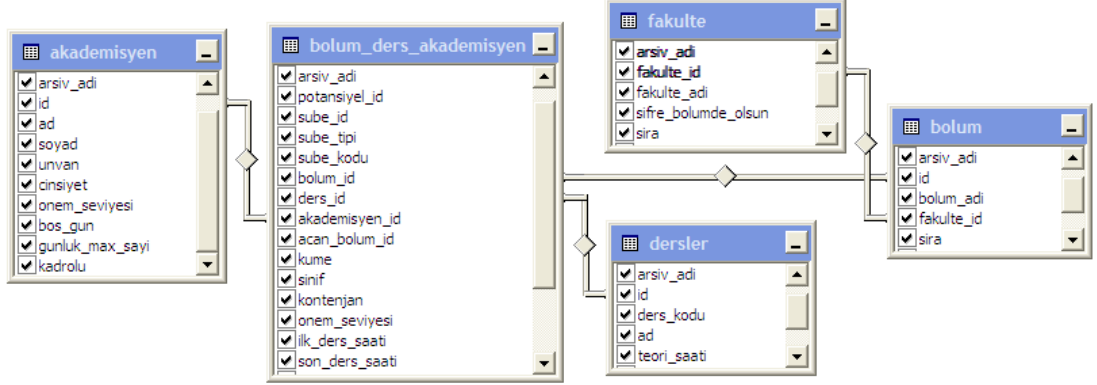
Admin data tablosunun yapısı Çizelge 4.12’de gösterilmiştir. Bu tablo içerisinde admin kullanıcısı için “şifre” bilgisi tutulmaktadır. Admin kullanıcısının diğer kullanıcılardan farkı sınırsız yetkiye sahip olmasıdır. Bu yetki sayesinde diğer kullanıcıların çalışabileceği gerekli ortamı oluşturur. Ayrıca belirlemiş olduğu kullanıcıların yetki seviyesini belirler.

Çizelge 4. 12 Admin Data Tablo Yapısı

Column Name	Data Type	Allow Nulls
sifre	nvarchar(50)	<input checked="" type="checkbox"/>

Genel veritabanı mimarisi tablosunun yapısı Çizelge 4.13’de gösterilmiştir. Bu tablo yapısında veritabanı içerisinde yer alan tabloların birbirleriyle olan ilişkisel yapısı gösterilmektedir.

Çizelge 4. 13 Genel Veritabanı Mimarisi



4.2 Uygulama Modülleri

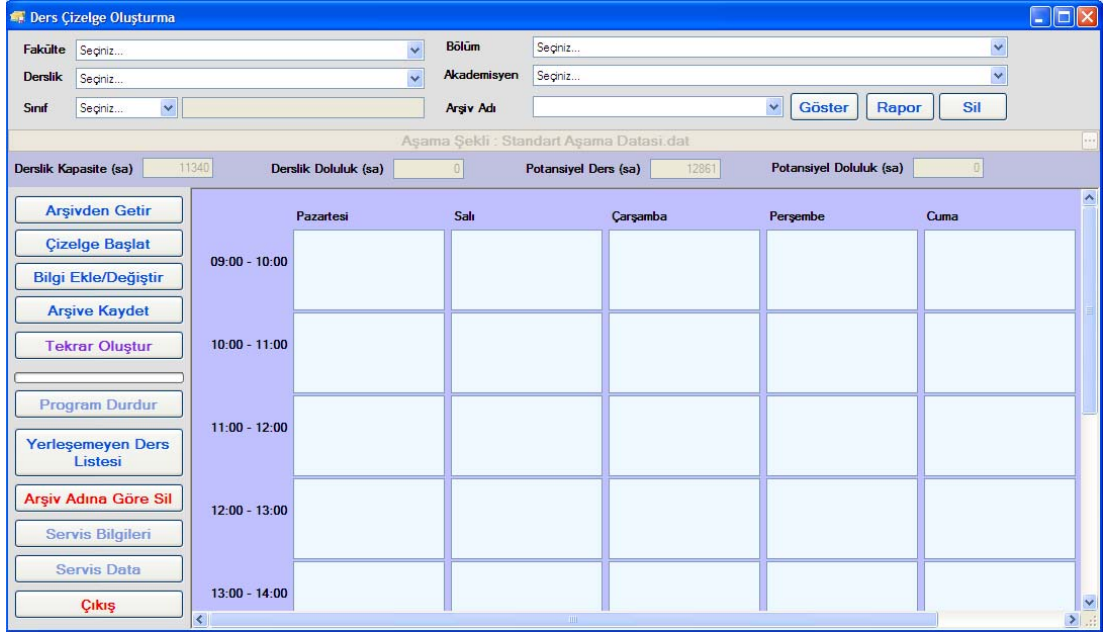
Uygulamamız 11 temel modülden oluşmaktadır. Bunlar; Kullanıcı Girişi, Akademisyen, Ders, Fakülte, Bölüm, Derslik, Bölüm Ders Akademisyen, Sabit Dersler, Çakışma Olmayacak Dersler, Program Başlat ve Akademisyenler İçin Uygun Gün ve Saatler Listesidir.

“Kullanıcı Girişi” kısmında iki tip kullanıcı yer almaktadır. Bunlar admin kullanıcı ve fakülte sekreteri kullanıcılarıdır. Admin kullanıcı kendisine ait kullanıcı adı ve şifresini girerek sisteme giriş yapar. Daha sonra “Servis Data” kontrolü yardımıyla veritabanından gerekli bilgileri çekip, belirli bir isim altında bu bilgileri kaydederek diğer kullanıcıların çalışabileceği ortamı hazırlar. Fakülte sekreteri kullanıcıları ise “Arşiv Adı” altında kaydedilmiş ilgili veritabanını seçip, kendilerine ait kullanıcı adı ve şifresini girerek sisteme giriş yaparlar. Şekil 4.1’de, “Kullanıcı Girişi” ekranı görülmektedir.



Şekil 4. 1 Kullanıcı Giriş Ekranı

Kullanıcı girişi ekranını geçtikten sonra karşımıza çıkan ekran “Ders Çizelge Oluşturma” ekranıdır. Bu ekran üzerinde; Arşivden Getir, Çizelge Başlat, Bilgi Ekle/Değiştir, Arşive Kaydet, Tekrar Oluştur, Program Durdur, Yerleşemeyen Ders Listesi, Arşiv Adına Göre Sil, Servis Bilgileri, Servis Data ve Çıkış alanları yer alır. Şekil 4.2’de, “Ders Çizelge Oluşturma” ekranı görülmektedir.



Şekil 4. 2 Ders Çizelge Oluşturma Ekranı

Arşivden Getir, veritabanına kaydetmiş olduğumuz bilgileri okumak için kullanılır.

Çizelge Başlat, ders programı oluşturmak için kullanacağımız ekrandır.

Arşive Kaydet, hazırlamış olduğumuz ders programını veritabanına kaydetmek için kullanmış olduğumuz bölümdür.

Tekrar Oluştur, programlanmış ders programından farklı bir ders programı oluşturmak için kullanmış olduğumuz bölümdür. Bu bölümü kullanıcı oluşturmuş olduğu ders programını beğenmeyip farklı bir ders programı oluşturmak için kullanır.

Program Durdur, başlatmış olduğumuz ders programını durdurmak için kullanılır.

Arşiv Adına Göre Sil, veritabanına kaydetmiş olduğumuz ders programını tamamen silmek için kullanılan bölümdür. Bu bölüme sadece admin kullanıcı tarafından erişilebilir.

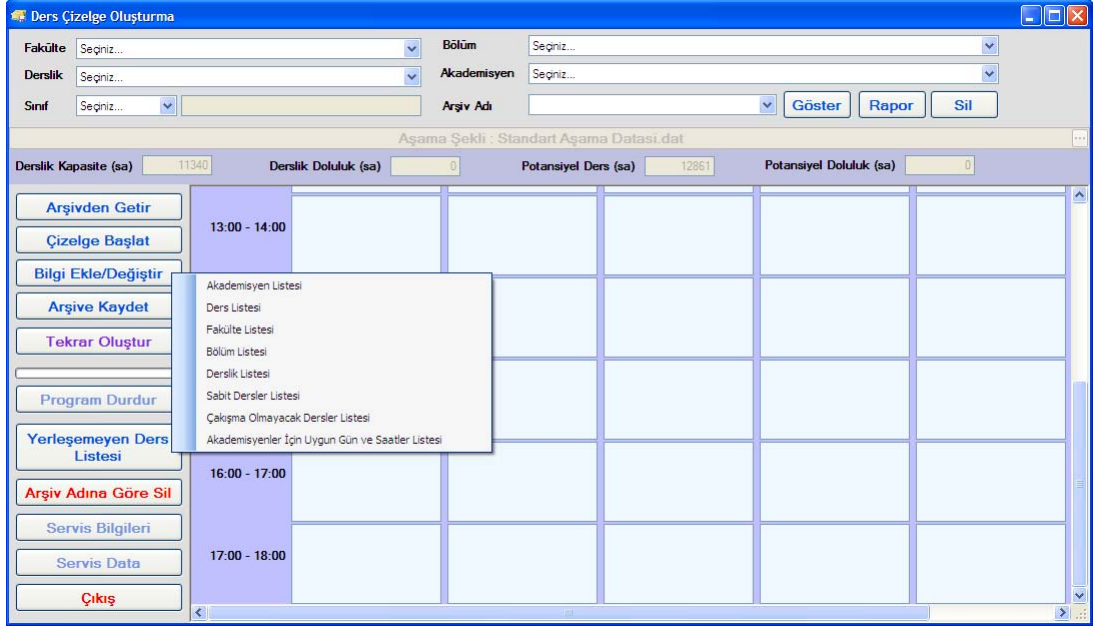
Servis Data, ders programını oluşturmak için gerekli verilerin ilgili veritabanından alınmasını sağlayan bölümdür. Bu bölüme sadece admin kullanıcı tarafından erişilebilir.

Çıkış, ders programından çıkmak için kullanılan bölümdür.

Sil, hazırlamış olduğumuz ders programını sınıf bazında silmek için kullanılan bölümdür.

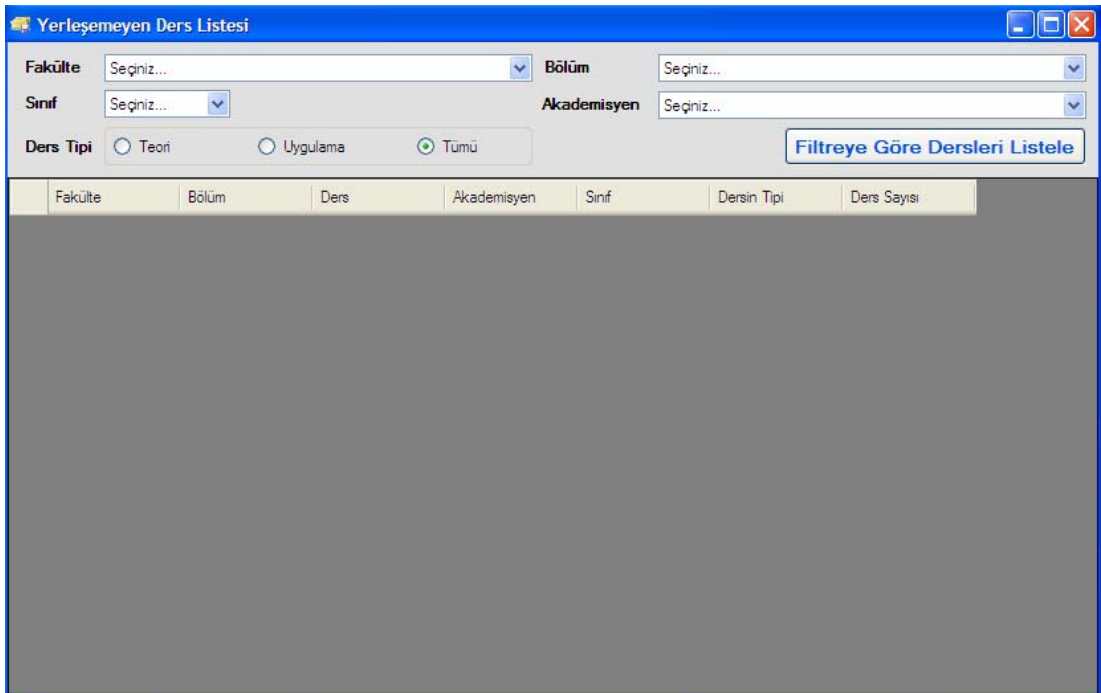
Arşiv Adı, hazırlamış olduğumuz ders programını belirli bir isim altında kaydetmek için kullanmış olduğumuz bölümdür.

Bilgi Ekle/Değiştir, ders programını oluşturmak için gerekli bilgi giriş/çıkış ekranlarının yer aldığı bölümdür. Şekil 4.3'de, "Bilgi Ekle/Değiştir" ekranı görülmektedir.



Şekil 4. 3 Bilgi Ekle/Değiştir Ekranı

Yerleşemeyen Ders Listesi, oluşturmuş olduğumuz ders programı içerisinde yerleştirilemeyen ders/dersleri listelemek için kullanılan bölümdür. Bu bölüm içerisinde yer alan ders/dersler kullanıcı tarafından ders programı içerisine elle yerleştirilir. Şekil 4.4’de, “Yerleştirilemeyen Ders Listesi” ekranı görülmektedir.



Şekil 4. 4 Yerleştirilemeyen Ders Listesi

Servis Bilgileri, bu bölüm içerisinde kullanıcının ilgili alanlara yazacağı sql sorguları aracılığıyla ilgili veritabanına bağlanılmasını sağlar. Bu bölüme sadece admin kullanıcı tarafından erişilebilir. Şekil 4.5’de, “Servis Bilgileri” ekranı görülmektedir.

Servis Bilgileri

Server: ***** User ID: *****
 Database: ***** Password: ***** **Kaydet**

Akademisyen
 select P.SK, P.Ad, P.Soyad, U.Adli, cast(isnull(PB.KadroluMu,0) as tinyint) from LIKAS.Personel P left join LIKAS.AkademikUnvanlar U ON P.AkademikUnvanSK = U.SK left join LISANS.PersonellerBirimler PB on PB.PersonelSK=P.SK left join LIKAS.IdariUnvanlar IU on IU.SK=PB.IdariUnvanSK where P.AktifMi = 1

Ders
 SELECT DISTINCT ISNULL(d.SK,0), ISNULL(d.Kodu, ''), ISNULL(d.AdTR, ''), ISNULL(d.TeorSaati, 0), ISNULL(d.UygulamaSaati, 0), isnull(d.BirimSK,0) FROM LIKAS.Dersler AS d INNER JOIN vwBolumSinfSubelemleriGetir AS bd ON d.SK = bd.DersSK

Fakülte
 select isnull(SK,0), isnull(TRAd, '') from dbo.vwUstBimler where sistemSK = 6

Bölüm
 select isnull(SK,0), isnull(TRAd, ''), isnull(UstBirimSK,0) from dbo.vwBimler where sistemSK = 6

Bölüm Ders Akademisyen
 SELECT fn.AlacakBirimSK,fn.DersSK,fn.AaglHocaSK,fn.SubeKodu,fn.SubeSK,fn.Sinf, sb.ToplamKontenjan, sb.SubeTipi,SK,Kume,AcanBirimSK FROM MUBIS2.dbo.vwBolumSinfSubelemleriGetir fn inner join GENEL.Subeler sb on fn.SubeSK = sb.SK order by fn.AlacakBirimSK,fn.DersSK,SubeKodu

Şekil 4. 5 Servis Bilgileri Ekranı

Göster, hazırlanmış olduğumuz ders programını görüntülemek için kullanılan bölümdür. Şekil 4.6’da, “Programlanmış Ders Çizelge” ekranı görülmektedir.

Ders Çizelge Oluşturma

Fakülte: Mühendislik Fakültesi Bölüm: Bilgisayar Mühendisliği
 Derslik: Seçiniz... Akademisyen: Seçiniz...
 Sınıf: 1.Sınıf Arşiv Adı: Bilgisayar Mühendisliği **Göster Rapor Sil**

Aşama Şekli : Standart Aşama Datası.dat

Derslik Kapasite (sa) 2160 Derslik Doluluk (sa) 63 Potansiyel Ders (sa) 12861 Potansiyel Doluluk (sa) 63

	Pazartesi	Salı	Çarşamba	Perşembe	Cuma
09:00 - 10:00	KİM 151 02 (T) Genel Kimya Doç. Dr. Hamit TOPUZ [M21]		BİL 133 01 (T) Algoritmalar ve Programlama I Öğr. Gör. Okan VARDARLI [M21]		FİZ 181 0101 (U) Fizik I Prof. Dr. İdris Adnan GUMUŞ [M22]
10:00 - 11:00			BİL 133 01 (T) Algoritmalar ve Programlama I Öğr. Gör. Okan VARDARLI [M21]		FİZ 181 0101 (U) Fizik I Prof. Dr. İdris Adnan GUMUŞ [M22]
11:00 - 12:00	MAT 181 0101 (U) Matematik I Yrd. Doç. Dr. EMIN ÇİFTÇİ [M26]	ING 151 32 (T) İngilizce I Okutman Hülya BOZDAĞ [M24]			
12:00 - 13:00	MAT 181 0101 (U) Matematik I Yrd. Doç. Dr. EMIN ÇİFTÇİ [M26]	ING 151 32 (T) İngilizce I Okutman Hülya BOZDAĞ [M24]			
13:00 - 14:00		KİM 151 02 (T) Genel Kimya Doç. Dr. Hamit TOPUZ [M21]			

Arşivden Getir Çizelge Başlat Bilgi Ekle/Değiştir Arşive Kaydet Tekrar Oluştur Program Durdur Yerleşemeyen Ders Listesi Arşiv Adına Göre Sil Servis Bilgileri Servis Data Çıkış

Şekil 4. 6 Programlanmış Ders Çizelge Ekranı

Rapor, hazırlanan ders programının yazıcıdan çıktı alınabilmesini sağlamak amacıyla kullanılan bölümdür. Bu bölümün kullanılabilmesi için öncelikle ders programının arşive kaydedilmesi gerekir. Şekil 4.7’de, “Hazırlanmış Ders Çizelgesine ait Rapor ” ekranı görülmektedir.

2010 - 2011 Eğitim - Öğretim Yılı Bahar Dönemi
Mühendislik Fakültesi - Bilgisayar Mühendisliği
Haftalık Ders Programı

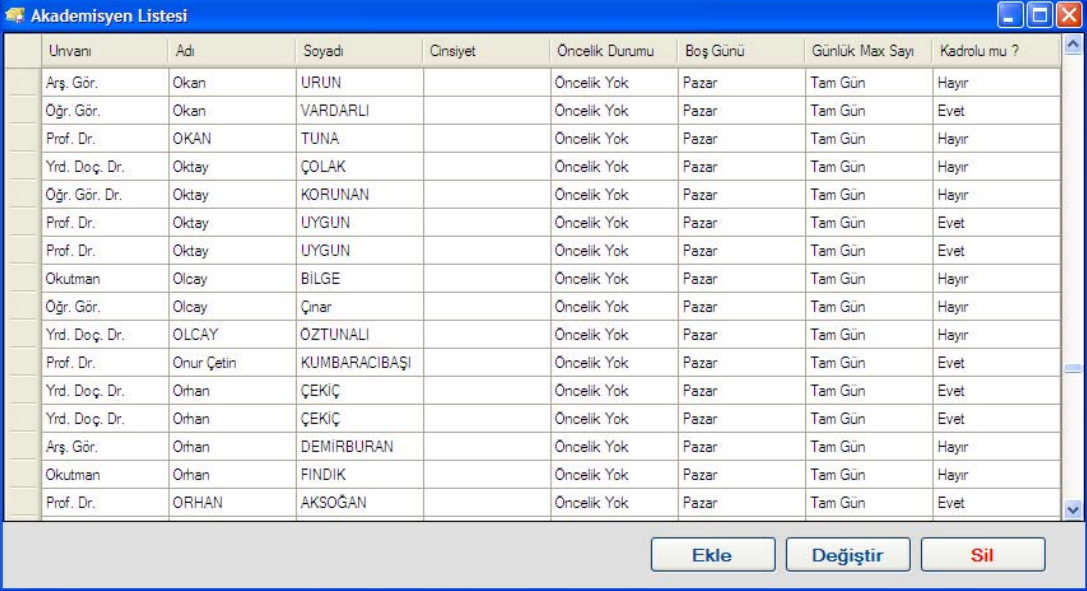
1.Sınıf

	Pazartesi	Salı	Çarşamba	Perşembe	Cuma
09:00 - 10:00	KİM 151 02 (T) Genel Kimya Doç. Dr.Hamit TOPUZ [MZ1]		BİL 133 01 (T) Algoritmalar ve Programlama I Öğr. Gör.Okan VARDARLI [MZ1]		FİZ 181 0101 (U) Fizik I Prof. Dr.İdris Adnan GÜMÜŞ [MZ2]
10:00 - 11:00			BİL 133 01 (T) Algoritmalar ve Programlama I Öğr. Gör.Okan VARDARLI [MZ1]		FİZ 181 0101 (U) Fizik I Prof. Dr.İdris Adnan GÜMÜŞ [MZ2]
11:00 - 12:00	MAT 181 0101 (U) Matematik I Yrd. Doç. Dr.EMİN ÇİFTÇİ [MZ6]	İNG 151 32 (T) İngilizce I OkutmanHülya BOZDAĞ [MZ4]			
12:00 - 13:00	MAT 181 0101 (U) Matematik I Yrd. Doç. Dr.EMİN ÇİFTÇİ [MZ6]	İNG 151 32 (T) İngilizce I OkutmanHülya BOZDAĞ [MZ4]			
13:00 - 14:00		KİM 151 02 (T) Genel Kimya Doç. Dr.Hamit TOPUZ [MZ1]			

Page 1 of 2 | Zoom Factor: 100%

Şekil 4. 7 Hazırlanmış Ders Çizelge Ekranının Rapor Görünümü

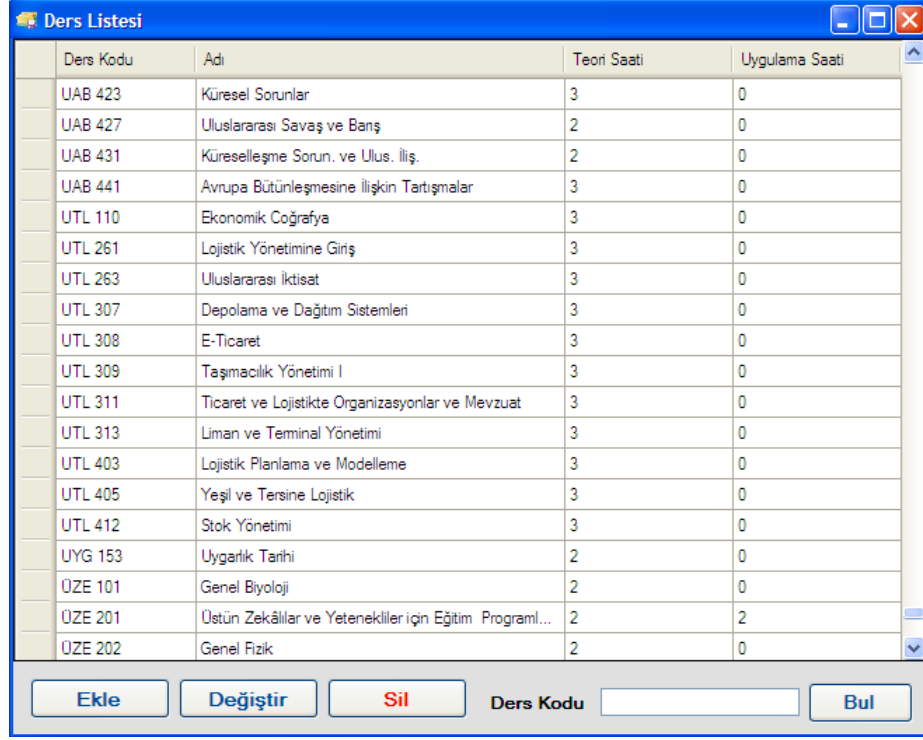
Akademisyen listesi ekranı üzerinde programı oluşturacak ders hocalarının bilgileri kaydedilebilir, değiştirilebilir veya silinebilir niteliktedir. Şekil 4.8’de, “Akademisyen Listesi” ekranı görülmektedir.



Unvanı	Adı	Soyadı	Cinsiyet	Öncelik Durumu	Boş Günü	Günlük Max Sayı	Kadrolu mu ?
Arş. Gör.	Okan	URUN		Öncelik Yok	Pazar	Tam Gün	Hayır
Öğr. Gör.	Okan	VARDARLI		Öncelik Yok	Pazar	Tam Gün	Evet
Prof. Dr.	OKAN	TUNA		Öncelik Yok	Pazar	Tam Gün	Hayır
Yrd. Doç. Dr.	Oktay	ÇOLAK		Öncelik Yok	Pazar	Tam Gün	Hayır
Öğr. Gör. Dr.	Oktay	KORUNAN		Öncelik Yok	Pazar	Tam Gün	Hayır
Prof. Dr.	Oktay	UYGUN		Öncelik Yok	Pazar	Tam Gün	Evet
Prof. Dr.	Oktay	UYGUN		Öncelik Yok	Pazar	Tam Gün	Evet
Okutman	Olcaay	BILGE		Öncelik Yok	Pazar	Tam Gün	Hayır
Öğr. Gör.	Olcaay	Çınar		Öncelik Yok	Pazar	Tam Gün	Hayır
Yrd. Doç. Dr.	OLCAY	OZTUNALI		Öncelik Yok	Pazar	Tam Gün	Hayır
Prof. Dr.	Onur Çetin	KUMBARACIBAŞI		Öncelik Yok	Pazar	Tam Gün	Evet
Yrd. Doç. Dr.	Orhan	ÇEKİÇ		Öncelik Yok	Pazar	Tam Gün	Evet
Yrd. Doç. Dr.	Orhan	ÇEKİÇ		Öncelik Yok	Pazar	Tam Gün	Evet
Arş. Gör.	Orhan	DEMİRBURAN		Öncelik Yok	Pazar	Tam Gün	Hayır
Okutman	Orhan	FINDIK		Öncelik Yok	Pazar	Tam Gün	Hayır
Prof. Dr.	ORHAN	AKSOĞAN		Öncelik Yok	Pazar	Tam Gün	Evet

Şekil 4. 8 Akademisyen Listesi Ekranı

Ders listesi ekranı üzerinde programı oluşturacak derslerin bilgileri kaydedilebilir, değiştirilebilir veya silinebilir niteliktedir. Şekil 4.9’da, “Ders Listesi” ekranı görülmektedir.



Ders Kodu	Adı	Teori Saati	Uygulama Saati
UAB 423	Küresel Sorunlar	3	0
UAB 427	Uluslararası Savaş ve Barış	2	0
UAB 431	Küreselleşme Sorun. ve Ulus. İliş.	2	0
UAB 441	Avrupa Bütünleşmesine İlişkin Tartışmalar	3	0
UTL 110	Ekonomik Coğrafya	3	0
UTL 261	Lojistik Yönetimine Giriş	3	0
UTL 263	Uluslararası İktisat	3	0
UTL 307	Depolama ve Dağıtım Sistemleri	3	0
UTL 308	E-Ticaret	3	0
UTL 309	Taşımacılık Yönetimi I	3	0
UTL 311	Ticaret ve Lojistikte Organizasyonlar ve Mevzuat	3	0
UTL 313	Liman ve Terminal Yönetimi	3	0
UTL 403	Lojistik Planlama ve Modelleme	3	0
UTL 405	Yeşil ve Tersine Lojistik	3	0
UTL 412	Stok Yönetimi	3	0
UYG 153	Uygartık Tarihi	2	0
ÜZE 101	Genel Biyoloji	2	0
ÜZE 201	Üstün Zekâllılar ve Yetenekliler için Eğitim Programl...	2	2
ÜZE 202	Genel Fizik	2	0

Ekle Değiştir Sil Ders Kodu Bul

Şekil 4. 9 Ders Listesi Ekranı

Fakülte listesi ekranı üzerinde programı oluşturacak fakültelerin bilgileri kaydedilebilir, değiştirilebilir veya silinebilir niteliktedir. Şekil 4.10’da, “Fakülte Listesi” ekranı görülmektedir.



Şekil 4. 10 Fakülte Listesi Ekranı

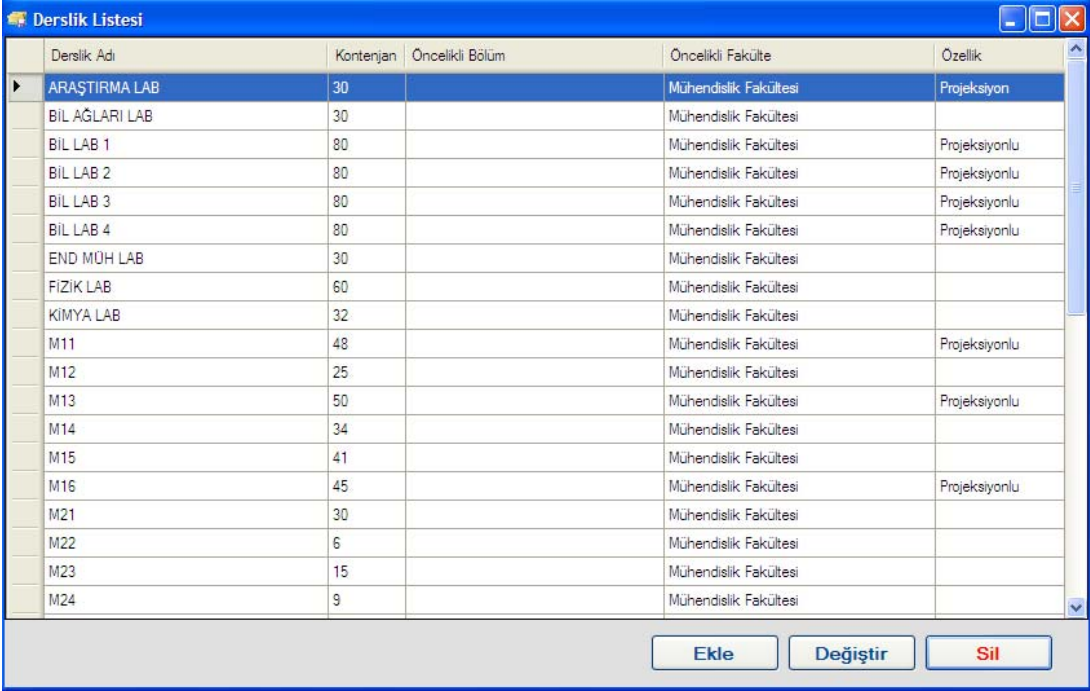
Bölüm listesi ekranı üzerinde programı oluşturacak bölümlerin bilgileri kaydedilebilir, değiştirilebilir veya silinebilir niteliktedir. Şekil 4.11’de, “Bölüm Listesi” ekranı görülmektedir.



Bölüm Adı	Fakülte
Atatürk İlkeleri	Rektörlüğe Bağlı Bölümler
Bankacılık ve Finans	İktisadi ve İdari Bilimler Fakültesi
Bankacılık ve Finans (İngilizce)	İktisadi ve İdari Bilimler Fakültesi
Bilgisayar Mühendisliği	Mühendislik Fakültesi
Bilgisayar Mühendisliği (İngilizce)	Mühendislik Fakültesi
Bilgisayar ve Öğretim Teknolojileri Öğretmenliği	Eğitim Fakültesi
Bilişim	Rektörlüğe Bağlı Bölümler
Çizgi Film - Animasyon	Güzel Sanatlar Fakültesi
Eğitim Bilimleri (Sanal)	Eğitim Fakültesi
Elektronik Mühendisliği	Mühendislik Fakültesi
Endüstri Mühendisliği	Mühendislik Fakültesi
Enerji Sistemleri Mühendisliği	Mühendislik Fakültesi
Felsefe	Fen-Edebiyat Fakültesi
Fizik (Sanal)	Mühendislik Fakültesi
Fotoğraf ve Video	Güzel Sanatlar Fakültesi
Gazetecilik	İletişim Fakültesi
Gemi ve Yat Tasarımı	Mimarlık Fakültesi
Görsel İletişim Tasarımı	İletişim Fakültesi
Görüntü Sanatları	Güzel Sanatlar Fakültesi

Şekil 4. 11 Bölüm Listesi Ekranı

Derslik listesi ekranı üzerinde programı oluşturacak derslik bilgileri kaydedilebilir, değiştirilebilir veya silinebilir niteliktedir. Bu ekran üzerinde kullanıcı tarafından derslere ait derslikler belirlenir. Şekil 4.12’de, “Derslik Listesi” ekranı görülmektedir.



Derslik Adı	Kontenjan	Öncelikli Bölüm	Öncelikli Fakülte	Özellik
ARAŞTIRMA LAB	30		Mühendislik Fakültesi	Projeksiyon
BİL AĞLARI LAB	30		Mühendislik Fakültesi	
BİL LAB 1	80		Mühendislik Fakültesi	Projeksiyonlu
BİL LAB 2	80		Mühendislik Fakültesi	Projeksiyonlu
BİL LAB 3	80		Mühendislik Fakültesi	Projeksiyonlu
BİL LAB 4	80		Mühendislik Fakültesi	Projeksiyonlu
END MÜH LAB	30		Mühendislik Fakültesi	
FİZİK LAB	60		Mühendislik Fakültesi	
KİMYA LAB	32		Mühendislik Fakültesi	
M11	48		Mühendislik Fakültesi	Projeksiyonlu
M12	25		Mühendislik Fakültesi	
M13	50		Mühendislik Fakültesi	Projeksiyonlu
M14	34		Mühendislik Fakültesi	
M15	41		Mühendislik Fakültesi	
M16	45		Mühendislik Fakültesi	Projeksiyonlu
M21	30		Mühendislik Fakültesi	
M22	6		Mühendislik Fakültesi	
M23	15		Mühendislik Fakültesi	
M24	9		Mühendislik Fakültesi	

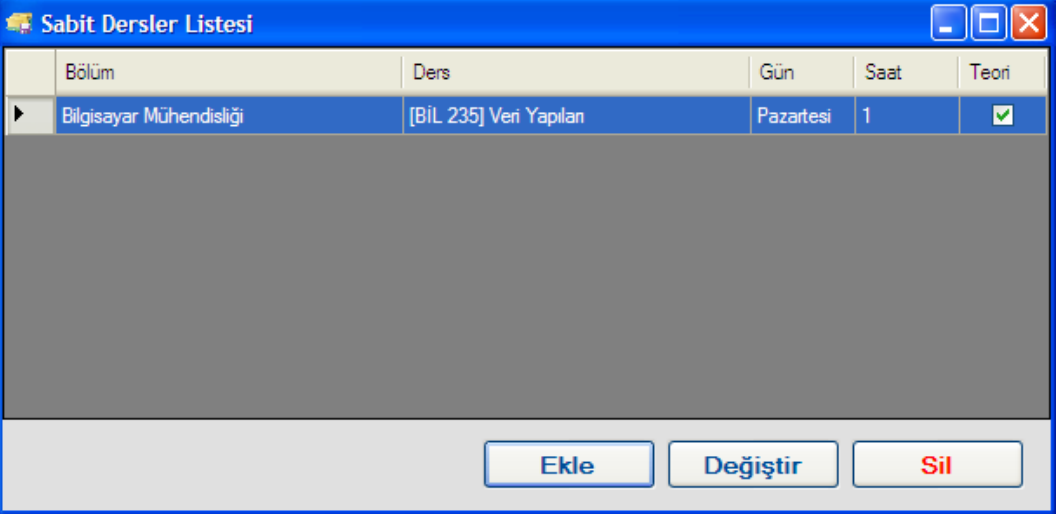
Şekil 4. 12 Derslik Listesi Ekranı

Bölüm Ders Akademisyen ekranı üzerinde kullanıcı, ders programını oluşturacak bilgiler üzerinde işlemler yapabiliyor. Bu ekran üzerinde yer alan; derslik alanı dersin hangi sınıfta yapılacağını, blok ders sayısı dersin ders programında kaçarlı bloklar halinde dağılacağını, günler kısmında dersin hangi günlerde verileceğini, ilk ders saati ve son ders saatinde ise dersin başlama ve bitiş saatleri kullanıcı tarafından belirtiliyor. Daha sonra kullanıcı bu ayarları yaptıktan sonra “çizelge oluştur” düğmesine tıklayarak haftalık ders programını oluşturur. Şekil 4.13’de, “Bölüm Ders Akademisyen” ekranı görülmektedir.

Ders	Şube Kodu	Akademisyen	Derslik	Blok Ders Sayısı	T / U	Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi	İlk Ders Saati	Son Ders Saati	Ders Tipi
[BİL. 301]...	BİL. 301.01	Öğr. Gör. Erd...	M11 [K...	2 Saat	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	09:00	17:00	Z
[BİL. 301]...	BİL. 301.0101	Öğr. Gör. Erd...	BİL. LAB...	2 Saat	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	09:00	17:00	Z
[BİL. 301]...	BİL. 301.02	Öğr. Gör. Erd...	Segniz...	Segniz...	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	Z
[BİL. 301]...	BİL. 301.0201	Öğr. Gör. Erd...	Segniz...	Segniz...	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	Z
[BİL. 301]...	BİL. 301.03	Öğr. Gör. Dr. ...	Segniz...	Segniz...	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	Z
[BİL. 301]...	BİL. 301.0301	Öğr. Gör. Dr. ...	Segniz...	Segniz...	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	Z
[BİL. 319]...	BİL. 319.01	Doğ. Dr. Oruç...	M16 [K...	1 Saat	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	09:00	Segniz	BS
[BİL. 319]...	BİL. 319.01	Doğ. Dr. Oruç...	Segniz...	Segniz...	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	BS
[BİL. 319]...	BİL. 319.01	Doğ. Dr. Oruç...	Segniz...	Segniz...	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	BS
[BİL. 319]...	BİL. 319.01	Doğ. Dr. Oruç...	Segniz...	Segniz...	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	BS
[BİL. 319]...	BİL. 319.01	Doğ. Dr. Oruç...	Segniz...	Segniz...	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	BS
[BİL. 311]...	BİL. 311.01	Öğr. Gör. Ser...	Segniz...	Segniz...	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	BS
[BİL. 311]...	BİL. 311.01	Öğr. Gör. Ser...	Segniz...	Segniz...	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	BS
[BİL. 311]...	BİL. 311.01	Öğr. Gör. Ser...	Segniz...	Segniz...	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	BS
[BİL. 311]...	BİL. 311.01	Öğr. Gör. Ser...	Segniz...	Segniz...	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	BS
[BİL. 311]...	BİL. 311.01	Öğr. Gör. Ser...	Segniz...	Segniz...	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Segniz	Segniz	BS

Şekil 4. 13 Bölüm Ders Akademisyen Listesi Ekranı

Sabit dersler listesi ekranı üzerinde programı oluşturacak sabit ders bilgileri kaydedilebilir, deęiştirilebilir veya silinebilir niteliktedir. Bu ekran üzerinde kullanıcının belirleyeceęi ders/dersler programa istenilen gün ve saatte yerleştirilir. Ekle butonu aracılıęıyla kullanıcı açılan ekran üzerinde sabit olarak yerleşmesini istedięi ders/dersleri belirleyebilir. Bu işlem yerleşimi yapılacak dersin hangi günde, hangi saatte ve dersin teori olup olmadığı belirlenerek yapılabilir. Şekil 4.14’de, “Sabit Dersler Listesi” ekranı görölmektedir.

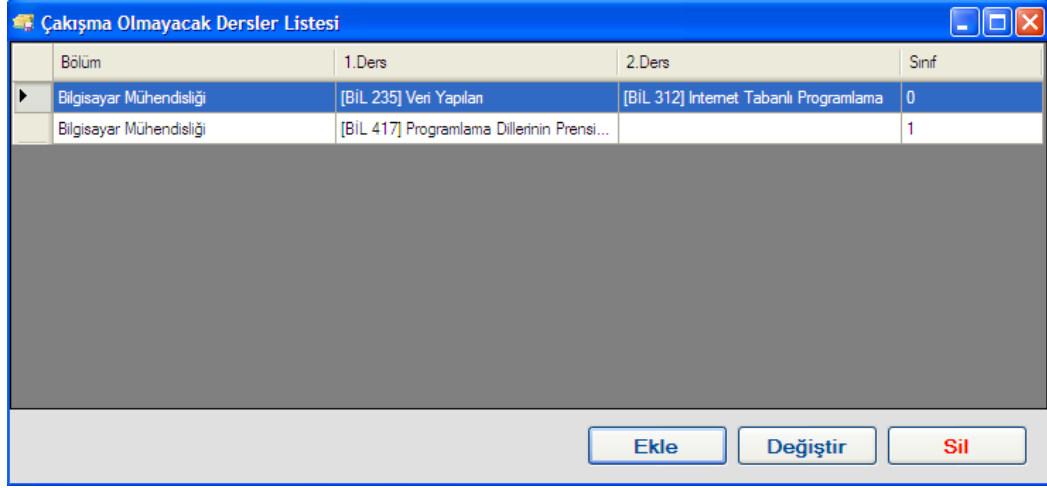


Bölüm	Ders	Gün	Saat	Teori
Bilgisayar Mühendislięi	[BİL 235] Veri Yapılan	Pazartesi	1	<input checked="" type="checkbox"/>

Ekle Deęiştir Sil

Şekil 4. 14 Sabit Dersler Listesi Ekranı

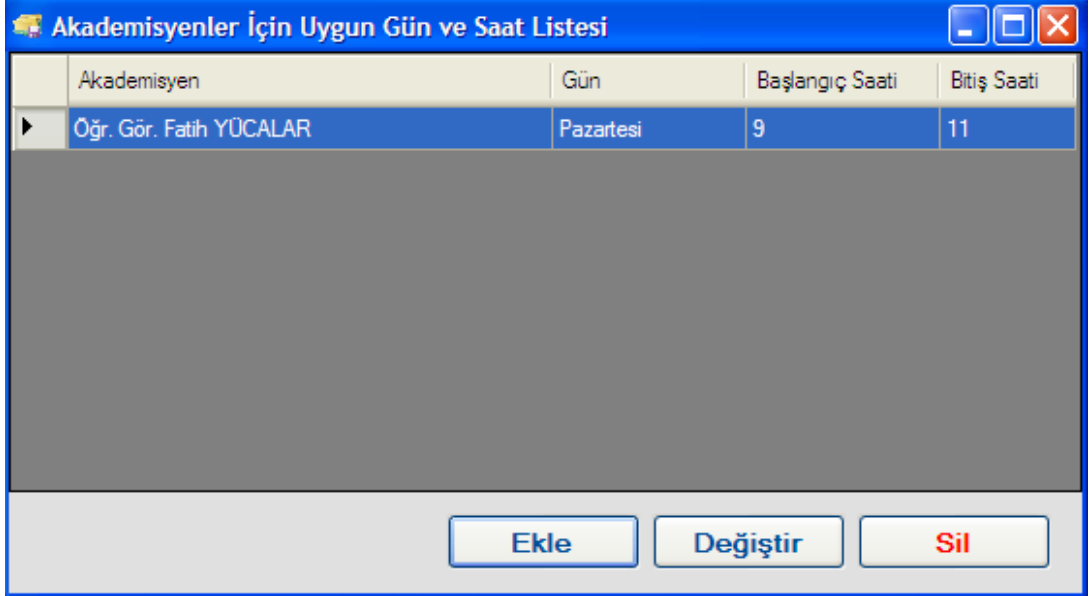
Çakışma olmayacak dersler listesi ekranı üzerinde programı oluşturacak ders bilgileri kaydedilebilir, değiştirilebilir veya silinebilir niteliktedir. Bu ekran üzerinde kullanıcının belirleyeceği ders/derslerin program içerisindeki çakışmaları önlenir. Ekle butonu aracılığıyla kullanıcı açılan ekran üzerinde çakışmamasını istediği ders/dersleri belirleyebilir. Bu işlem seçilen herhangi iki ders arasında olabileceği gibi, seçilen herhangi bir dersin çakışmamasını istediği ilgili bölümdeki ilgili sınıfa ait tüm dersleri kapsayabilir. Şekil 4.15’de, “Çakışma Olmayacak Dersler Listesi” ekranı görülmektedir.



Bölüm	1.Ders	2.Ders	Sınıf
Bilgisayar Mühendisliği	[BİL 235] Veri Yapılan	[BİL 312] İnternet Tabanlı Programlama	0
Bilgisayar Mühendisliği	[BİL 417] Programlama Dillerinin Prensi...		1

Şekil 4. 15 Çakışma Olmayacak Dersler Listesi Ekranı

Akademisyenler için uygun gün ve saatler listesi ekranı üzerinde programı oluşturacak ders bilgileri kaydedilebilir, değiştirilebilir veya silinebilir niteliktedir. Bu ekran üzerinde kullanıcı tarafından akademisyenlerin ders vereceği gün ve saat aralıkları belirlenir. Ekle butonu aracılığıyla kullanıcı açılan ekran üzerinde akademisyenler için belirli gün ve saat aralıklarına sahip ders girişi yapabilmelerini sağlar. Bu işlem ilgili dersin hangi günde ve hangi saat aralıklarında verileceğini belirlemekle yapılabilir. Şekil 4.16'da, "Akademisyenler İçin Uygun Gün ve Saatler Listesi" ekranı görülmektedir.



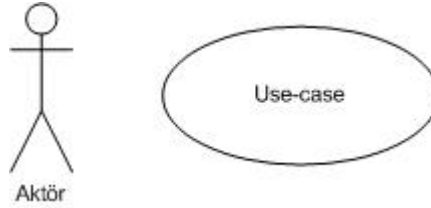
Akademisyen	Gün	Başlangıç Saati	Bitiş Saati
Öğr. Gör. Fatih YÜCALAR	Pazartesi	9	11

Ekle Değiştir Sil

Şekil 4. 16 Akademisyenler İçin Uygun Gün ve Saatler Listesi Ekranı

4.3 Programın Use-Case Diyagramı

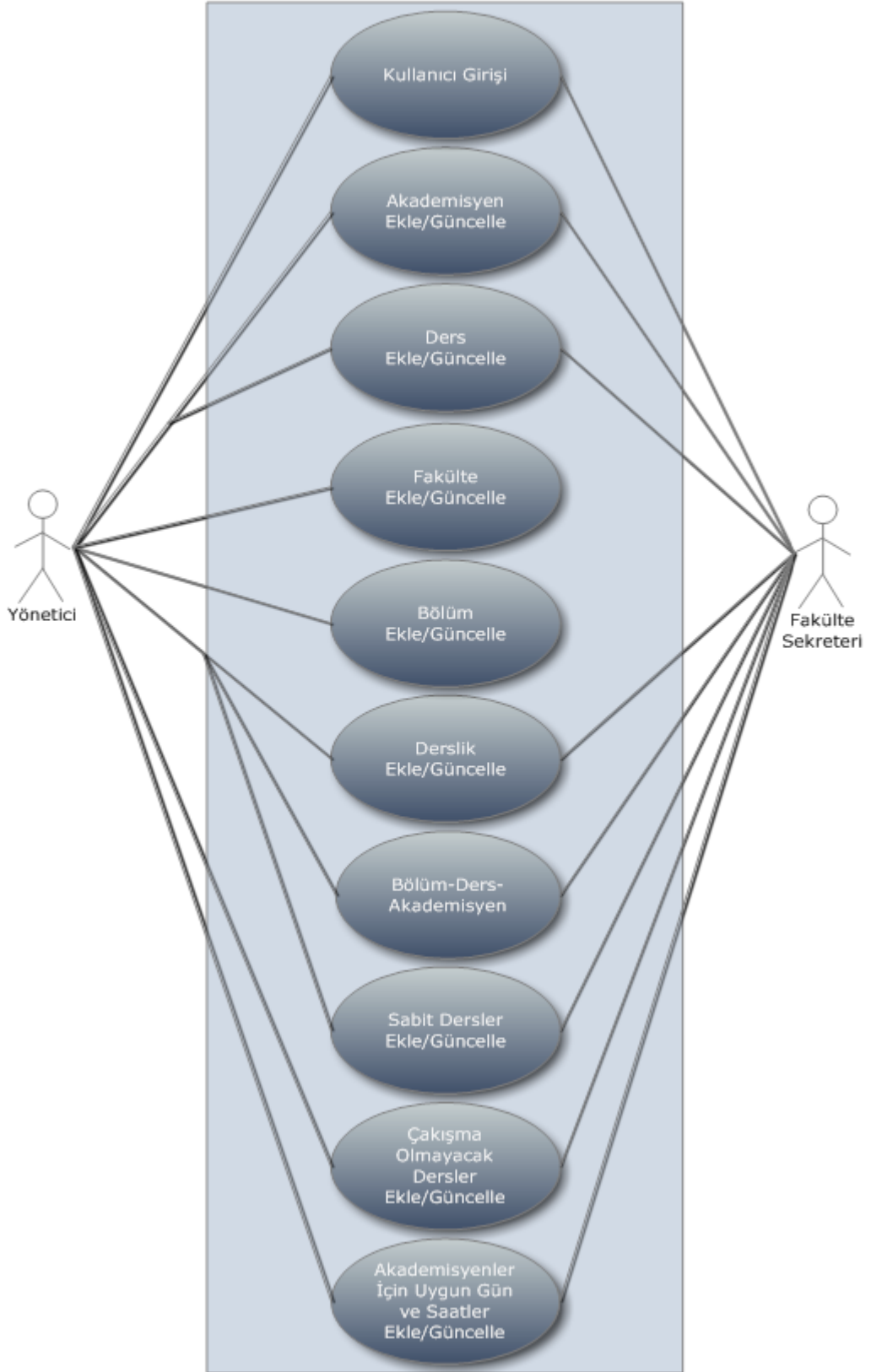
Use-case, bir kullanıcı ile sistem arasındaki etkileşimi göstermek üzere kullanılan senaryolar topluluğudur. Bir use-case diyagramında aktörler ile use case'ler arasındaki ilişki gösterilir. Bu diyagram türünün iki temel elemanı aktör ve use case'dir.



Aktör, modellemekte olduğumuz sistem ile etkileşimde bulunan bir kişi ya da diğer bir sistemi temsil eder. Use-case ise kullanıcının bir işlemini yerine getirmek üzere yapılan eylemleri yapan sistemin, dışarıdan bir görünüşüdür.

Örneğimizde ders programı sisteminde, sistemle etkileşimde olan iki ana aktör ya da rol bulunmaktadır. Yönetici aktörü genel bir kullanıcı olmayıp, ders programı sistemi içerisinde yer alan modüllere sınırsız bir erişim sağlar. Fakülte sekreteri aktörü genel bir kullanıcı olup, ders programı sistemi içerisinde yer alan modüllerin belirli bir kısmına erişim sağlar. Diyagramdan da görüldüğü gibi ders programı sistemi kullanıcı girişi, akademisyen, ders, fakülte, bölüm, derslik, bölüm-ders-akademisyen, sabit dersler, çakışma olmayacak dersler ve akademisyenler için uygun gün ve saatler gibi on ana use case'den oluşmaktadır. Bu use case'ler içerisinde fakülte kullanıcısı bölüm ve fakülte alanlarında yetkiye sahip değildir. Şekil 4.17'de, "Haftalık Ders Çizelgeleme Problemi Use Case Diyagram" ekranı görülmektedir.

Haftalık Ders Çizelgeleme Problemi



Şekil 4. 17 Haftalık Ders Çizelgeleme Problemi Use Case Diyagramı

4.4 Programın İş Birliđi (Collaboration) Diyagramı

Bir sistemin amacını yerine getirebilmesi için, sistemin bütün parçalarının işlerini yerine getirmesi gerekir. Bu işler genellikle birkaç parçanın beraber çalışmasıyla mümkün olabilir. Bu tür ilişkileri göstermek için iş birliđi (collaboration) diyagramları kullanılır. İşbirliđi diyagramları, mesaj yollayan ve alan nesnelerin yapısal organizasyonları üzerinde duran bir tür etkileşim diyagramıdır. Bu diyagramlar, yazılım sisteminin statik yapısını ve dinamik davranışlarını birlikte gösterir.

Entity Class: Uzun süre kullanacağımız bilgileri modellerken kullandığımız sınıflardır.

Boundary Class: Yazılım ürünü ile ortam arasındaki etkileşimi modeller. Genellikle giriş/çıkış ve raporlama işlemleri ile ilişkilendirilmiş sınıflardır.

Control Class: Karmaşık işlemlerin ve algoritmaların yürütüldüğü sınıflardır.



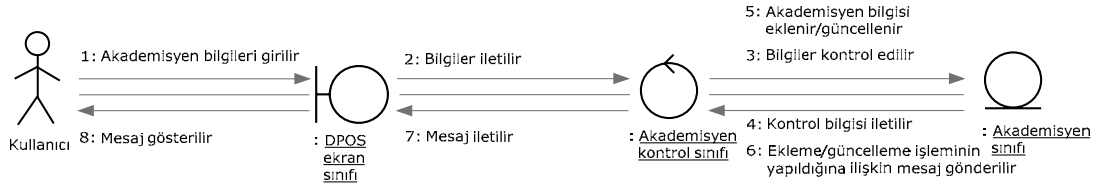
Örneğimizde ders programı sistemini oluşturan tablo yapısı ele alınmıştır. Sistemin temelini oluşturan 5 temel tablo yapısının iş birliđi diyagramı üzerinde durulup bu tablo yapılarına ait ekleme, güncelleme ve silme işlemlerini gerçekleştiren iş birliđi diyagramları üzerinde durulmuştur.

Akademisyen Ekleme/Güncelleme Use-Case'i

Use-Case Tanımı: Bu use-case akademisyene ait bilgilerin eklenmesi veya güncellenmesini sağlar.

Senaryo:

1. Kullanıcı, akademisyenin bilgilerini girer.
2. Sistem bilgileri kontrol eder.
3. Sistem bilgilerin eklendiği/güncellendiğine ilişkin mesaj gönderir.



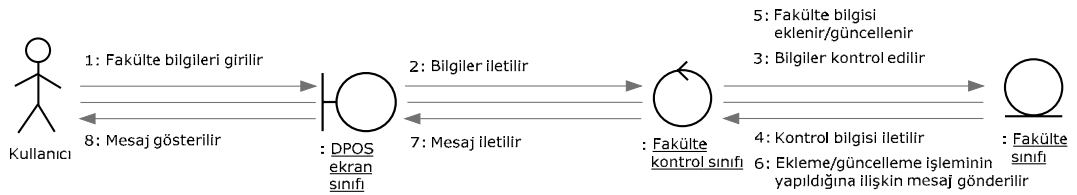
Şekil 4. 18 Akademisyen Ekleme/Güncelleme Use-Case'i

Fakülte Ekleme/Güncelleme Use-Case'i

Use-Case Tanımı: Bu use-case fakülteye ait bilgilerin eklenmesi veya güncellenmesini sağlar.

Senaryo:

1. Kullanıcı, fakülte bilgilerini girer.
2. Sistem bilgileri kontrol eder.
3. Sistem bilgilerin eklendiği/güncellendiğine ilişkin mesaj gönderir.



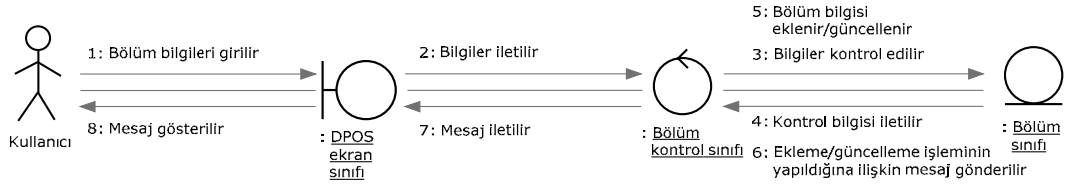
Şekil 4. 19 Fakülte Ekle/Güncelleme Use-Case'i

Bölüm Ekleme/Güncelleme Use-Case'i

Use-Case Tanımı: Bu use-case bölüme ait bilgilerin eklenmesi veya güncellenmesini sağlar.

Senaryo:

1. Kullanıcı, bölüm bilgilerini girer.
2. Sistem bilgileri kontrol eder.
3. Sistem bilgilerin eklendiği/güncellendiğine ilişkin mesaj gönderir.



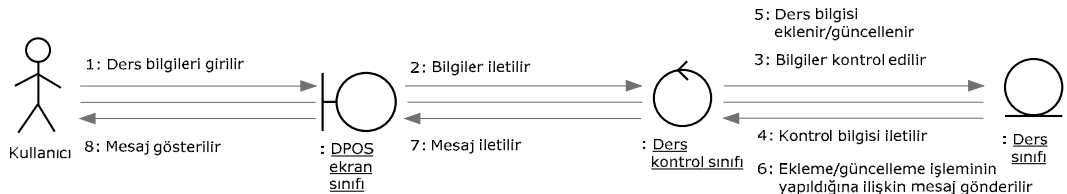
Şekil 4. 20 Bölüm Ekleme/Güncelleme Use-Case'i

Ders Ekleme/Güncelleme Use-Case'i

Use-Case Tanımı: Bu use-case derslere ait bilgilerin eklenmesi veya güncellenmesini sağlar.

Senaryo:

1. Kullanıcı, ders bilgilerini girer.
2. Sistem bilgileri kontrol eder.
3. Sistem bilgilerin eklendiği/güncellendiğine ilişkin mesaj gönderir.



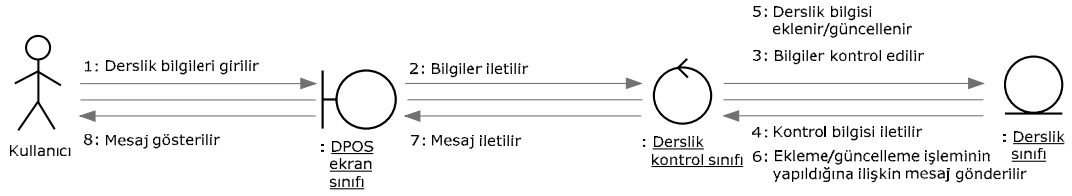
Şekil 4. 21 Ders Ekleme/Güncelleme Use-Case'i

Derslik Ekleme/Güncelleme Use-Case'i

Use-Case Tanımı: Bu use-case dersliklere ait bilgilerin eklenmesi veya güncellenmesini sağlar.

Senaryo:

1. Kullanıcı, derslik bilgilerini girer.
2. Sistem bilgileri kontrol eder.
3. Sistem bilgilerin eklendiği/güncellendiğine ilişkin mesaj gönderir.



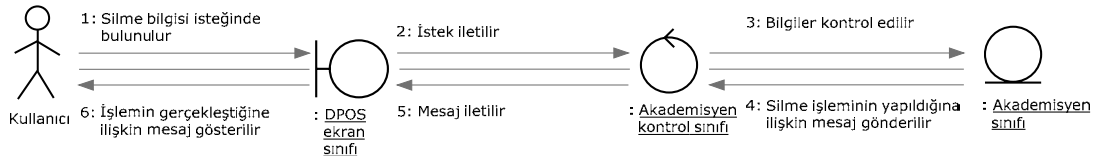
Şekil 4. 22 Derslik Ekleme/Güncelleme Use-Case'i

Akademisyen Silme Use-Case'i

Use-Case Tanımı: Bu use-case akademisyene ait bilgilerin silinmesini sağlar.

Senaryo:

1. Kullanıcı, akademisyenin bilgilerini girer.
2. Sistem bilgileri kontrol eder.
3. Sistem bilgilerin silindiğine ilişkin mesaj gönderir.



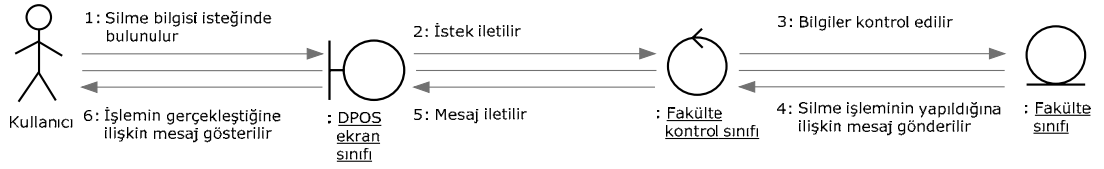
Şekil 4. 23 Akademisyen Silme Use-Case'i

Fakülte Silme Use-Case'i

Use-Case Tanımı: Bu use-case fakülteye ait bilgilerin silinmesini sağlar.

Senaryo:

1. Kullanıcı, fakülte bilgilerini girer.
2. Sistem bilgileri kontrol eder.
3. Sistem bilgilerin silindiğine ilişkin mesaj gönderir.



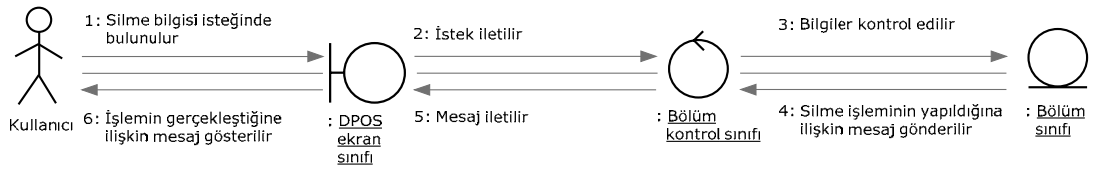
Şekil 4. 24 Fakülte Silme Use-Case'i

Bölüm Silme Use-Case'i

Use-Case Tanımı: Bu use-case bölüme ait bilgilerin silinmesini sağlar.

Senaryo:

1. Kullanıcı, bölüm bilgilerini girer.
2. Sistem bilgileri kontrol eder.
3. Sistem bilgilerin silindiğine ilişkin mesaj gönderir.



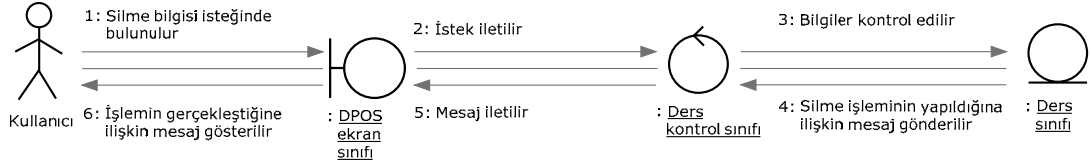
Şekil 4. 25 Bölüm Silme Use-Case'i

Ders Silme Use-Case'i

Use-Case Tanımı: Bu use-case derslere ait bilgilerin silinmesini sağlar.

Senaryo:

1. Kullanıcı, ders bilgilerini girer.
2. Sistem bilgileri kontrol eder.
3. Sistem bilgilerin silindiğine ilişkin mesaj gönderir.



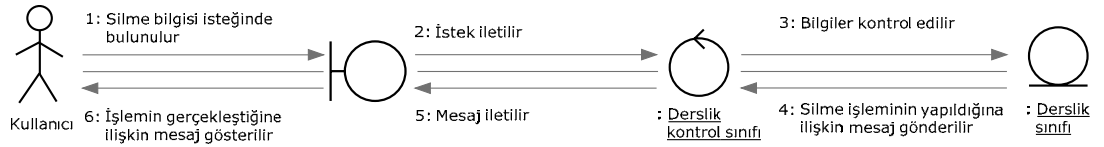
Şekil 4. 26 Ders Silme Use-Case'i

Derslik Silme Use-Case'i

Use-Case Tanımı: Bu use-case dersliklere ait bilgilerin silinmesini sağlar.

Senaryo:

1. Kullanıcı, derslik bilgilerini girer.
2. Sistem bilgileri kontrol eder.
3. Sistem bilgilerin silindiğine ilişkin mesaj gönderir.



Şekil 4. 27 Derslik Silme Use-Case'i

4.5 Sistemin Bilgi Tabanı

Kısıtlar

Zaman çizelgeleme kısıtları, Maltepe Üniversitesi'nin ihtiyaçlarına göre düzenlenmiştir. Kısıtlar, eğitim kalitesinin iyileştirilmesi, öğrenci ve öğretim elemanlarının performansının yükseltilmesi, öğretim elemanı isteklerinin karşılanması ve idari ihtiyaçlara cevap verilebilmesi esas alınarak oluşturulmuştur. Tüm bu kısıtlar, 2 ana grupta toplanabilir: zorunlu kısıtlar ve zayıf kısıtlardır. Takip eden kısımda, sistemimizin desteklediği kısıtlar detaylı olarak anlatılmıştır.

Zorunlu Kısıtlar

- Ders yerleşimi yapılırken dersi veren hocaların ders verebileceği gün(ler) dikkate alınır. Bu günlerde hocaya ders ataması yapılır. Bu işlem “Akademisyenler için uygun gün ve saatler ekranı” altında yapılır. Bu ekran daha çok ders-saat ücretli hocaların kullanımı için tasarlanmıştır.
- Ders yerleşirken dersi veren hocaların o saatte başka sınıfta dersi olup olmadığı kontrol edilir.
- Yerleşecek dersin laboratuvar dersi de varsa teori ve laboratuvar saatleri aynı gün içerisinde olmak koşuluyla en az bir saatlik arayla ders yerleşimi yapılır.
- Ders yerleşimi aynı gün içinde ise, dersin önce teorisi sonra uygulaması yerleştirilir.
- Laboratuvar /Uygulama ders saatleri art arda olabilir.
- “Sabit dersler listesi ekranı” altına girilen dersler öncelikli olarak belirtilen dağılım şekline göre yerleştirilir.

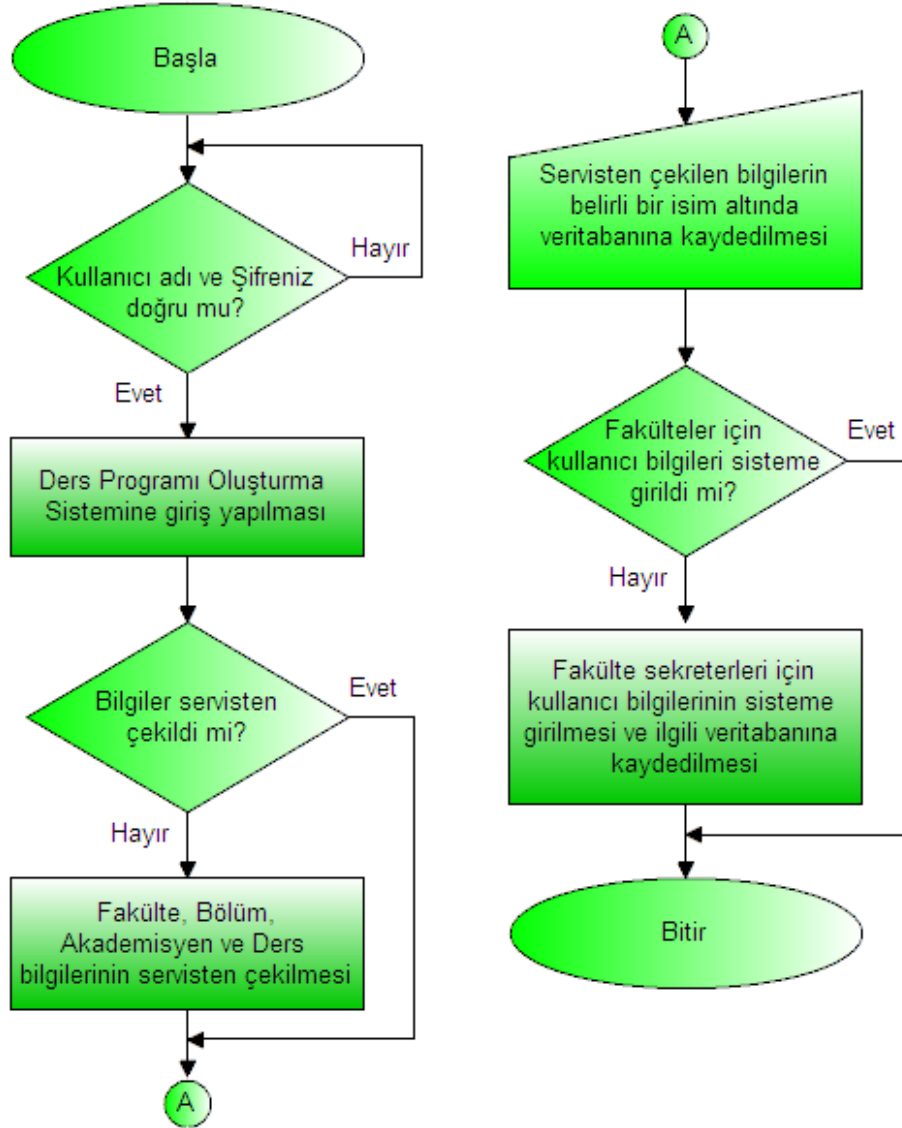
- Dersler önceden belirtilen günlere göre haftanın altı günü programa yerleştirilebilir.

Zayıf Kısıtlar

- Kuramsal saatleri 2 ve 3 olan dersler, günde bir saatlik dilimlere, 4 ve üstü olanlar günde iki saati aşmayan dilimlere bölünecektir. Ayrıca kuramsal saati 3 ve 3'ün katı olan dersler 75 dakikalık dilim olarak bölünebilir.
- 3 kuramsal saati olan dersler otomatik olarak Pazartesi-Çarşamba-Cuma günlerine planlanacaktır. (Böyle bir ders kullanıcı tarafından istenilen günlere yerleştirilebilir.)
- 2 kuramsal saati olan dersler Pazartesi-Çarşamba-Cuma günlerinden herhangi ikisine planlanacaktır. (Böyle bir ders kullanıcı tarafından istenilen günlere yerleştirilebilir.)
- Derslerin uygulamalı kısımları, istenirse 4 saati aşmamak koşuluyla blok halinde yapılabilecektir.
- Alan seçmeli, bölüm seçmeli ve üniversite seçmeli ders kuşağı haftanın her günü 9-15 kuşağında yer alacaktır.
- Türk Dili, Atatürk İlkeleri ve İnkılâp Tarihi ve İNG kodlu dersler haftanın her günü 15-17 kuşağında yer alacaktır.

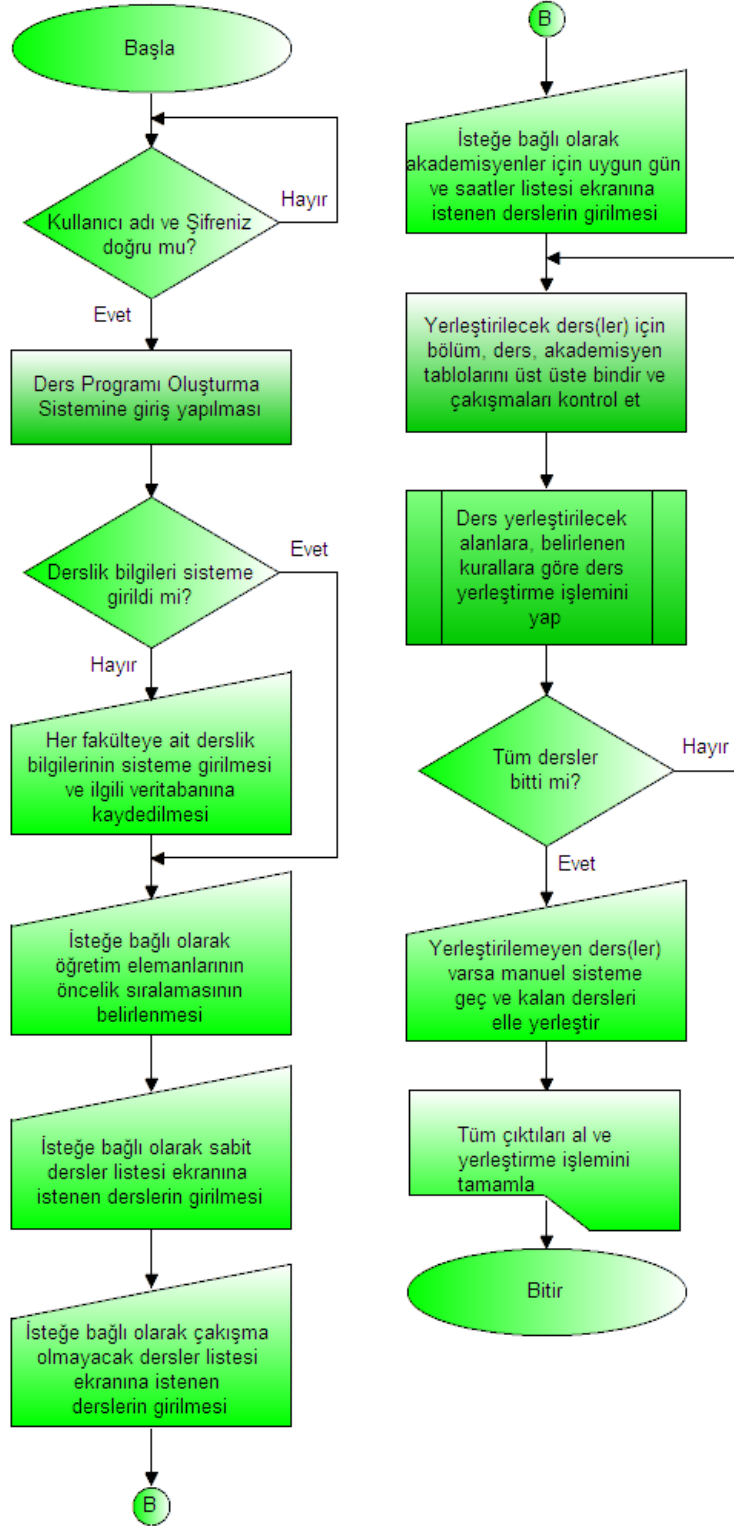
4.6 Programın Genel Algoritması

Ders program sisteminin kurallarından oluşan bilgi tabanındaki bilgileri çıkarmak için bir program geliştirilmiştir. Bu program bilgi tabanındaki bilgilere dayanarak ders yerleştirme işlemlerini yapmaktadır. Program belirli bir süre içinde bulabildiği en iyi yerleştirmeyi son çıktı olarak almaktadır. Eğer işlem sonunda yerleştirilemeyen dersler varsa elle yerleşemeyen ders listesinden ders yerleştirme işlemi yapılmaktadır. Şekil 4.28’de, “Programın Yönetici Kullanıcıları İçin Genel Algoritma” ekranı görülmektedir.



Şekil 4. 28 Programın Yönetici Kullanıcıları İçin Genel Algoritması

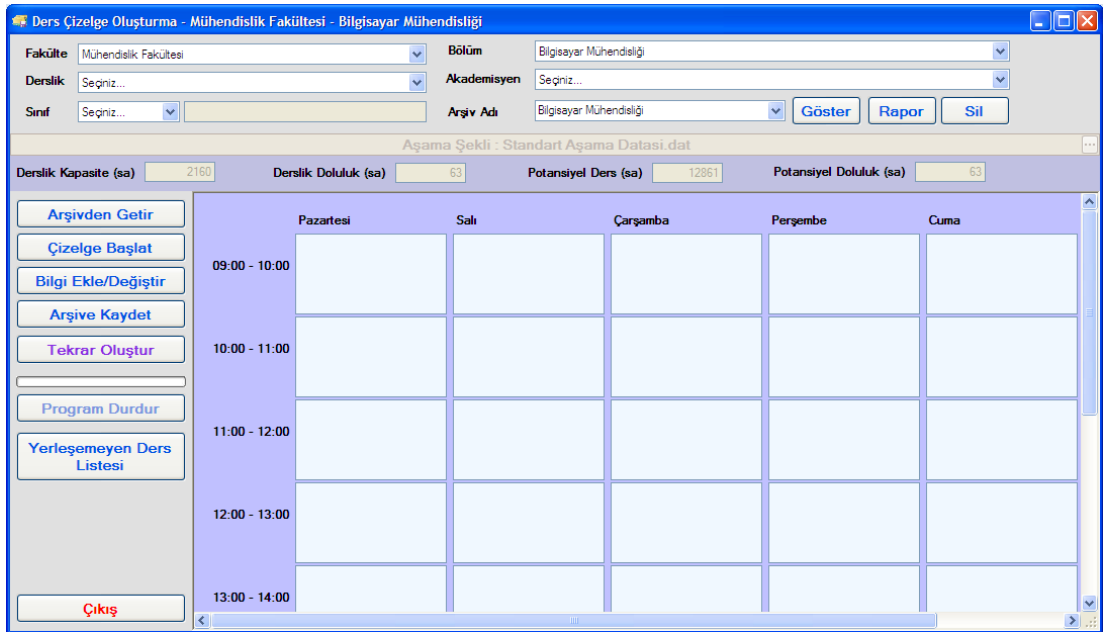
Şekil 4.29’da, “Programın Fakülte Kullanıcıları İçin Genel Algoritma” ekranı görülmektedir.



Şekil 4. 29 Programın Fakülte Kullanıcıları İçin Genel Algoritması

4.7 Programın İşleyişi

Programı kullanacak fakültelerdeki görevli kişiler kullanıcı girişi sayfasından kendilerine ait kullanıcı parolasıyla sisteme giriş yaparlar. Daha sonra arşivden getir, Çizelge başlat, bilgi ekle/değiştir, arşive kaydet, tekrar oluştur, program durdur, yerleşemeyen ders listesi ve çıkış işlemlerinin yer aldığı ana ekrana ulaşılmaktadır. Şekil 4.30'da, "Programın Yönetildiği Ana Ekran" görülmektedir.



Şekil 4. 30 Programın Yönetildiği Ana Ekran

Programa öncelikli olarak dört tane bilginin veritabanından getirilmesi gerekmektedir. Bunlar; akademisyen listesi, fakülte listesi, bölüm listesi ve dersler listesidir. Daha sonra derslik listesi bölümünden her fakülteye ait dersliklerin girişi yapılarak arşive kaydet bölümünden bilgilerin veritabanına kaydedilmesi gerekmektedir. Eğer bir ders belirli bir yere yerleştirilmek isteniyorsa bu elle başta yerleştirilmekte ya da sabit dersler listesi bölümünden dersin istenilen gün ve saatinin girilmesiyle sağlanmaktadır.

Ders yerleştirme işlemleri yapılırken 5 temel tablo kullanılmıştır. Bu tablolar;

- Akademisyen tablosu,
- Fakülte tablosu,
- Bölüm tablosu,
- Dersler tablosu,
- Derslikler tablosudur.

Oluşturulan bu tablolar haftanın 5 günü (istenirse cumartesi gününde dâhil edilebilir) 9.00-18.00 saatleri arasında ders yerleşimi yapılacak şekilde ayarlanmıştır.

Eğer bir ders yerleştirilecekse bu 5 tablonun tamamına yerleştirilmeli ve bu tablolarda yerleştirilen saatlerde başka bir ders bulunmamalıdır. Yerleştirilen saatlere yazmak üzere genel bir kod geliştirilmiş ve dersin işleneceği saatlere yazılmıştır. Kod oluşturulurken dersle ilgili tüm bilgileri barındırması istenmiştir. Buna göre oluşturulan örnek bir kod şu şekildedir.

BİL 221 01 (T)	Dersin kodunu göstermektedir. Fakültede resmi olarak kullanılan kod sistemidir. İlk üç harf dersi verebilecek bölümü temsil eder. 4. rakam kaçınıcı sınıfa ait olduğunu gösterir. 7 ve 8. rakamlar ise dersin şube numarasını gösterir. “T” sembolü ise dersin teori mi yoksa uygulamamı olduğunu gösterir.
Bilgisayar Yapısı	Dersin adını temsil etmektedir.
Yrd. Doç. Dr. Ali AKMAN	Dersi veren öğretim elemanını temsil etmektedir.
[MA3]	Dersin verileceği dersliği göstermektedir. Burada mühendislik asma kat 3 nolu dersliğe ders verilmektedir.

BİL 221 01 01 (U)	Dersin kodunu göstermektedir. Fakültede resmi olarak kullanılan kod sistemidir. İlk üç harf dersi verebilecek bölümü temsil eder. 4. rakam kaçınıc sınıfa ait olduğunu gösterir. 7 ve 8. rakamlar ise dersin şube numarasını gösterir. 9 ve 10. rakamlar ise ilgili şubeye ait uygulama dersinin olduğunu gösterir. “U” sembolü ise dersin teori mi yoksa uygulamamı olduğunu gösterir.
Bilgisayar Yapısı	Dersin adını temsil etmektedir.
Yrd. Doç. Dr. Ali AKMAN	Dersi veren öğretim elemanını temsil etmektedir.
[MA3]	Dersin verileceği dersliđi göstermektedir. Burada mühendislik asma kat 3 nolu dersliđe ders verilmektedir.

Veriler girilip gerekli ayarlamalar yapıldıktan sonra ders dağıtım işlemine başlanmaktadır. Ders dağıtım işlemi otomatik olarak yapılırken veriler girildikten sonra hiçbir müdahale yapılamamaktadır. Program öncelikli olarak yukarıda verilen kurallara göre dersleri öncelik sırasına göre dizmektedir. Daha sonra dersin verilebileceği saatleri tespit eder ve bu saatlerden en uygun olana dersi rastgele yerleştirir. Program verilen bir süre içerisinde bulabildiği en iyi yerleştirme işlemi yapmaktadır. Eğer verilen kısıtlamalara göre yerleştirilemeyen dersler varsa bu dersler yerleşemeyen dersler listesinden elle yerleştirilmektedir. Ayrıca baştan bazı derslerin yerleşim işlemi programa bırakılmak istenmezse bu dersler elle programa yerleştirilebilir. Şekil 4.31’de, “Elle Ders Yerleştirme İşleminin Yapıldığı Ekran” görülmektedir.

	Fakülte	Bölüm	Ders	Akademisyen	Sınıf	Dersin Tipi	Ders Sayısı
▶	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[ATA 151 07] Atatürk İlkeleri Ve İnkılap Tarihi I	Yrd. Doç. Dr.Orhan ÇEKİÇ	2	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[ATA 151 07] Atatürk İlkeleri Ve İnkılap Tarihi I	Yrd. Doç. Dr.Orhan ÇEKİÇ	2	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[ATA 151 10] Atatürk İlkeleri Ve İnkılap Tarihi I	Yrd. Doç. Dr.Orhan ÇEKİÇ	1	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[ATA 151 10] Atatürk İlkeleri Ve İnkılap Tarihi I	Yrd. Doç. Dr.Orhan ÇEKİÇ	1	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 133 01] Algoritmalar ve Programlama I	Öğr. Gör.Okan VARDARLI	1	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 133 01] Algoritmalar ve Programlama I	Öğr. Gör.Okan VARDARLI	1	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 133 0101] Algoritmalar ve Programlama I	Öğr. Gör.Okan VARDARLI	1	Uygulama	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 133 0101] Algoritmalar ve Programlama I	Öğr. Gör.Okan VARDARLI	1	Uygulama	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 133 02] Algoritmalar ve Programlama I	Öğr. Gör.Okan VARDARLI	1	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 133 02] Algoritmalar ve Programlama I	Öğr. Gör.Okan VARDARLI	1	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 133 0201] Algoritmalar ve Programlama I	Öğr. Gör.Okan VARDARLI	1	Uygulama	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 133 0201] Algoritmalar ve Programlama I	Öğr. Gör.Okan VARDARLI	1	Uygulama	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 133 0202] Algoritmalar ve Programlama I	Öğr. Gör.Okan VARDARLI	1	Uygulama	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 133 0202] Algoritmalar ve Programlama I	Öğr. Gör.Okan VARDARLI	1	Uygulama	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 221 01] Bilgisayar Yapısı	Yrd. Doç. Dr.Ali AKMAN	2	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 221 0101] Bilgisayar Yapısı	Yrd. Doç. Dr.Ali AKMAN	2	Uygulama	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 237 01] Nesneye Yönelik Programlama	Yrd. Doç. Dr.Turgay Tugay BILGIN	2	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 237 01] Nesneye Yönelik Programlama	Yrd. Doç. Dr.Turgay Tugay BILGIN	2	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 237 01] Nesneye Yönelik Programlama	Yrd. Doç. Dr.Turgay Tugay BILGIN	2	Kuramsal	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 237 0101] Nesneye Yönelik Programlama	Yrd. Doç. Dr.Turgay Tugay BILGIN	2	Uygulama	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 237 0101] Nesneye Yönelik Programlama	Yrd. Doç. Dr.Turgay Tugay BILGIN	2	Uygulama	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 237 0101] Nesneye Yönelik Programlama	Yrd. Doç. Dr.Turgay Tugay BILGIN	2	Uygulama	2
	Mühendislik Fakültesi	Bilgisayar Mühendisliği	[BİL 237 02] Nesneye Yönelik Programlama	Yrd. Doç. Dr.Turgay Tugay BILGIN	2	Kuramsal	2

Şekil 4. 31 Elle Ders Yerleştirme İşleminin Yapıldığı Ekran

Ders dağıtım işlemi elle yapılırken ders seçildikten sonra ilgili tüm alanlar dolmakta ve sadece dersin verileceği derslik seçilmektedir. Şekil 4.32’de, “Elle Ders Yerleştirme İşlemi Yapılırken Doluluk ve Boşluk Alanlarını Gösteren Ekran” görülmektedir.

Şekil 4. 32 Elle Ders Yerleştirme İşlemi Yapılırken Doluluk ve Boşluk Alanlarını Gösteren Ekran

En sonunda oluşturulmuş ders programı kullanıcının göster butonunu tıklamasıyla ekranda gözükür. Şekil 4.33’de, “Oluşturulmuş Ders Çizelge Ekranı” görülmektedir.

Şekil 4. 33 Oluşturulmuş Ders Çizelge Ekranı

Tüm derslerin yerleşimi yapıldıktan sonra oluşan ders programının çıktısı alınabilmektedir. Bunun için kullanıcı ilk olarak oluşan ders programını arşive kaydetmesi gerekir. Daha sonra kullanıcı rapor butonunu tıkladığı takdirde ise, oluşmuş ders programı pdf formatına aktarılıp kaydedilmeye hazır hale getirilir. Şekil 4.34’de, “Oluşturulan Ders Çizelge Rapor Ekranı” görülmektedir.

2010 - 2011 Eğitim - Öğretim Yılı Bahar Dönemi Mühendislik Fakültesi - Bilgisayar Mühendisliği Haftalık Ders Programı					
1.Sınıf					
	Pazartesi	Salı	Çarşamba	Perşembe	Cuma
09:00 - 10:00	KİM 151 02 (T) Genel Kimya Doç. Dr.Hamit TOPUZ [MZ1]		BİL 133 01 (T) Algoritmalar ve Programlama I Öğr. Gör.Okan VARDARLI [MZ1]		FİZ 181 0101 (U) Fizik I Prof. Dr.İdris Adnan GÜMÜŞ [MZ2]
10:00 - 11:00			BİL 133 01 (T) Algoritmalar ve Programlama I Öğr. Gör.Okan VARDARLI [MZ1]		FİZ 181 0101 (U) Fizik I Prof. Dr.İdris Adnan GÜMÜŞ [MZ2]
11:00 - 12:00	MAT 181 0101 (U) Matematik I Yrd. Doç. Dr.EMİN ÇİFTÇİ [MZ6]	İNG 151 32 (T) İngilizce I OkutmanHülya BOZDAĞ [MZ4]			
12:00 - 13:00	MAT 181 0101 (U) Matematik I Yrd. Doç. Dr.EMİN ÇİFTÇİ [MZ6]	İNG 151 32 (T) İngilizce I OkutmanHülya BOZDAĞ [MZ4]			
13:00 - 14:00		KİM 151 02 (T) Genel Kimya Doç. Dr.Hamit TOPUZ [MZ1]			

Şekil 4. 34 Oluşturulan Ders Çizelge Rapor Görünümü

5. SONUÇ

Günümüz teknolojisine uygun olarak, kurumlarda işlemlerin bilgisayarlar vasıtasıyla otomatik olarak yaptırılması büyük önem taşımaktadır. Geliştirilen bu sistemler burada olduğu gibi bir uzmanın düşünme ve bilgisini yöneten sistemler olabileceği gibi sezgisel yöntemler ya da yapay zekâ teknikleri kullanılarak da geliştirilebilir. Yapılmış olan bu çalışmanın diğer çalışmalardan ayıran en önemli özelliği network'u ortak bir platform olarak kullanmasıdır. İstendiğinde sistem otomatik olarak kendisi ders yerleştirme işlemini yapmakta ya da istenirse elle işlemler yapılabilmektedir.

Bu çalışmada Maltepe Üniversitesi'ne ait Fakülteler esas alınarak bir ders çizelgeleme problemi analiz edilmiş ve problemin çözümüne yönelik başarılı sonuçlar elde edilmiştir. Diğer ticari ders çizelgeleme programları incelenmiş, avantaj ve dezavantajları uygulama tasarlanırken ve geliştirilirken göz önünde bulundurulmuştur. Kullanıcıdan çeşitli bilgi girişlerini alabilmek için C#.NET programlama dili ile kullanıcı dostu arayüzler geliştirilmiştir. Girilen verilerin saklanabilmesi için SQL Server veritabanı sunucu yazılımı kullanılmıştır.

Sonuç olarak, ders çizelgeleme problemini çözen bir algoritma ve çözümleri gösteren kullanışlı bir arayüz geliştirilmiştir.

KAYNAKLAR

1. EK Burke, DG Elliman, PH Ford and RF Weare, "*Examination Timetabling in British Universities - A Survey*", Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling (ICPTAT'95, Napier University, Edinburgh, UK, 30th Aug - 1st Sept 1995), pp 423-434.
2. EK Burke, DG Elliman and RF Weare, "*The Automation of the Timetabling Process in Higher Education*", Journal of Educational Technology Systems, vol 23, no 4, pp 257-266, Baywood Publishing Company, 1995.
3. French S., (1982), "*Sequencing and Scheduling. An introduction to the Mathematics of the Job shop*".
4. "*Resource Allocation Problems. Algorithmic Approaches*", Toshihide Ibaraki and Naoki Katoh. (1988).
5. "*Computers and Intractability: A guide to the theory of NP-Completeness*", Garey M.R and Johnson D.S. (1979).
6. "*A Promising Genetic Algorithm Approach to Job-Shop Scheduling, Rescheduling, and Open-Shop Scheduling Problems*", Hsiao-Lan Fang, Peter Ross, Dave Corne, in Proceedings of the Fifth International Conference on Genetic Algorithms, S. Forrest (ed), San Mateo: Morgan Kaufmann, pages 375--382, 1993.
7. "*Successful Lecture Timetabling with Evolutionary Algorithms*", Peter Ross, Dave Corne, Hsiao-Lan Fang, Appears in Workshop Notes, ECAI'94 Workshop W17: Applied Genetic and other Evolutionary Algorithms.

8. Burke E. K., Petrovic S., “Recent Research Directions in Automated Timetabling”, *European Journal of Operational Research - EJOR*, 140/2, 2002, 266-280.
9. Zhang L., Lau S.K., “Constructing university timetable using constraint satisfaction programming approach”, *Proceedings of the 2005 International Conference on Computational Intelligence for Modeling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC’05)*, Vienna, Austria, 55-60 (2005).
10. Botsalı, A.R., “A Timetabling Problem: Constraint and Mathematical Approaches”, Master Dissertation, *The Institute of Engineering and Sciences of Bilkent University*, Ankara, 1-15 (2000).
11. Daskalaki S., Birbas T., Housos E., “An Integer Programming Formulation for a Case Study in University Timetabling”, *European Journal of Operational Research*, 153 (1): 117-135 (2004).
12. Legierski W., Widawski R., “System of Automated Timetabling”, *25th Int. Conf. Information Technology Interfaces ITI 2003*, Cavtat, Croatia, 495-500, (2003).
13. Kakmacı, Ö., “Eğitim Kurumlarında Karşılaşılan Çizelge Hazırlama Problemlerinin Çözümüne İlişkin Yaklaşımlar”, Yüksek Lisans Tezi, *Osmangazi Üniversitesi Fen Bilimleri Enstitüsü*, Eskişehir, 9-13 (2003).
14. Schaerf, A. “Local Search Techniques for Large High School Timetabling Problems”, *Systems, Man and Cybernetics, Part A, IEEE Transactions*, 29 (4), 368-377 (1999).

15. Öner, A., “Deniz Harp Okulunda Çizelge/Atama Problemleri ve Çözüm Yöntemleri”, Doktora Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 59-75 (2002).
16. Şahin, T., “Goal Programming Approach to Solve the Timetabling Problem at Turkish Military Academy”, Master Dissertation, *The Institute of Economics and Social Sciences of Bilkent University*, Ankara, 1-13 (2004).
17. Burke E. K., Newall J.P., “A Multistage Evolutionary Algorithm for the Timetable Problem”, *IEEE Transactions On Evolutionary Computation*, 3 (1): 63-74 (1999).
18. MirHassani S.A., “Improving Paper Spread in Examination Timetables Using Integer Programming”, *Applied Mathematics and Computation*, 179: 702–706 (2006).
19. Lai L., Hsueh N., Huang L., Chen T., “An Artificial Intelligence Approach to Course Timetabling”, *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, Washington D.C., USA, 389-396, (2006).
20. Daskalaki S., Birbas T., “Efficient Solutions for a University Timetabling Problem Through Integer Programming”, *European Journal of Operational Research*, 160 (1): 106-120 (2005).
21. Santos H.G., Ochi L.S., Sauza M.J.F., “A Tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem”, *Journal of Experimental Algorithmics (JEA)*, 10: 2-9 (2005).
22. Bhatt, V., Sahajpal R., “Lecture Timetabling Using Hybrid Genetic Algorithms”, *Intelligent Sensing and Information Processing, Proceedings of International Conference on 2004*, Chennai, India, 29-34 (2004).

23. Burke E. K., McCollum B., Meisels A., Petrovic S., Qu R., “A Graph-based Hyperheuristic for Educational Timetabling Problems”, *European Journal of Operational Research*, 176: 177–192 (2007).
24. Cambazard H., Demazeau F., Jussien N., David P., “Interactively Solving School Timetabling Problems Using Extensions of Constraint Programming”, *Practice and Theory of Automated Timetabling V, 5th International Conference*, Pittsburgh, PA, USA, Revised Selected Papers. Springer Lecture Notes in Computer Science, 3616: 190-207 (2005).
25. Erdoğan, P., “Atama Modellerinde Timetabling Problemi ve Ders Çizelgeleme İçin Uygulama Geliştirme Denemesi”, Doktora Tezi, *Atatürk Üniversitesi Sosyal Bilimler Enstitüsü*, Erzurum, 76-82 (2003).
26. Adamidis P., Arapakis P., “Evolutionary Algorithms in Lecture Timetabling”, *Evolutionary Computation, Proceedings of the 1999 Congress*, Washington, USA, 2: 1145-1151 (1999).
27. Abbas, A.M., Tsang, E.P.K., “Constraint-based timetabling-a case study”, *Computer Systems and Applications, ACS/IEEE International Conference*, Beirut, Lebanon, 67-72 (2001).
28. Blum, C., Correia, S., Paechter, B., Rossi-Doria, O., Snoek, M., “A GA Evolving Instructions for a Timetable Builder”. *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2002)*, Gent, Belgium, 120-123 (2002).
29. Tam V., Ting D., “Combining the Min-Conflicts and Look-Forward Heuristics to Effectively Solve a Set of Hard University Timetabling Problems”, *15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03)*, Sacramento, California, USA, 492-496 (2003).

30. Sigl B., Golub M., Mornar V., “Solving Timetable Scheduling Problem Using Genetic Algorithms”, *Information Technology Interfaces, ITI 2003*, Proceedings of the 25th International Conference, Dubrovnik, Croatia, 519 – 524 (2003).
31. Mahdi, O., Aion, R.N., Zainuddin, R., “Using a Genetic Algorithm Optimizer Tool to Generate Good Quality Timetables”, *Electronics, Circuits and Systems, ICECS 2003*, Sharjah, United Arab Emirates, Proceedings of the 2003 10th IEEE International Conference, 3: 1300-1303 (2003).
32. Dimopoulou M., Miliotis P., “An Automated University Course Timetabling System Developed in a Distributed Environment: A Case Study”, *European Journal of Operational Research*, 153 (1): 136-147 (2004).
33. Piechowiak S., Kolski C., “Towards a Generic Object Oriented Decision Support System for University Timetabling: An Interactive Approach”, *International Journal of Information Technology & Decision Making*, 3 (1): 179–208 (2004).
34. Parthiban, P., Ganesh, K., Narayanan, S., Dhanalakshmi, R., “Preferences Based Decision-Making Model (PDM) for Faculty Course Assignment Problem”, *Engineering Management Conference Proceedings*, IEEE International, Singapore, 3: 1338- 1341 (2004).
35. Santiago-Mozos R., Salcedo-Sanz S., DePrado-Cumplido M., Bousño-Calzón C., “A Two-Phase Heuristic Evolutionary Algorithm for Personalizing Course Timetables: A Case Study in a Spanish University”, *Computers & Operations Research*, 32 (7): 1761-1776 (2005).

36. Beligiannis G.N., Moschopoulos C.N., Kaperonis G.P., Likothanassis S.D., “Applying evolutionary computation to the school timetabling problem: The Greek case”, *Computers & Operations Research*, In Press, Corrected Proof, Available online 10 October 2006 (2006).
37. Al-Yakoob S.A., Sherali H.D., “A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations”, *European Journal of Operational Research*, 180 (3): 1028-1044 (2007).
38. Head C., Shaban S., “A Heuristic Approach to Simultaneous Course/Student Timetabling”, *Computers & Operations Research*, 34 (4): 919-933 (2007).

ÖZGEÇMİŞ

Ahmet USTA, 1984 yılında Rize / Güneysu'da doğdu. Öğrenimlerini sırasıyla Adacami İlkokulu ve Mimar Sinan Endüstri Meslek Lisesi Bilgisayar Teknik Lisesi Bölümü'nde tamamladı. 2003 yılında Karadeniz Teknik Üniversitesi Rize Meslek Yüksekokulu Bilgisayar Programcılığı Bölümünü kazandı ve 2005 yılında bölüm ikincisi olarak mezun oldu. 2006 yılında Dikey Geçiş Sınavı (DGS) ile Maltepe Üniversitesi Mühendislik – Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümünü kazandı ve 2009 yılında mezun oldu. Yaz stajlarını, Maltepe Üniversitesi Uzaktan Eğitim Sistemi altında (İstanbul, 2008 yaz, 30 iş günü) yaptı. Ocak 2011'de, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Yüksek Lisans programında yüksek lisans öğrenimi yapmaya hak kazandı. Haziran 2009 – Temmuz 2010 tarihleri arasında, çalışmalarını sürdürmekte olduğu Maltepe Üniversitesi Uzaktan Eğitim Birimi altında yazılımcı olarak görev aldı. Ağustos 2010'da, halen çalışmalarını sürdürmekte olduğu Maltepe Üniversitesi Bilişim Bölüm Başkanlığı altına atandı.