

Evrimsel Hesaplama Tekniği Kullanarak Sınav Takvimi Otomasyon Sistemi Geliştirilmesi

Developing Examination Scheduling Automation System By Using Evolutionary Computing Technique

Abdulkadir YALDIR^{a*}, Ceyda BAYSAL^b

^aPamukkale Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 20070, Denizli

^bİzmir Yüksek Teknoloji Enstitüsü, Bilgi İşlem Daire Başkanlığı, 35430, Urla, İzmir

Geliş Tarihi/Received : 25.10.2011, Kabul Tarihi/Accepted : 28.11.2011

ÖZET

Bu çalışma ile üniversitelerde belirli dönemlerde manüel olarak hazırlanan sınav takvimi uygulamaları otomasyon kapsamına alınmıştır. Uygulamada Evrimsel Hesaplama yöntemi kullanılmış olup, gerekli kontroller ve iyileştirmeler üniversite yönetiminin isteği doğrultusunda yapılmıştır. Uygulama iki bölümden oluşmaktadır. Birinci bölümde veriler ilgili birimlerden web ortamında toplanmakta, ikinci bölümde ise geliştirilen masaüstü uygulama çalıştırılarak istenen sonuçlar alınmaktadır. Uygulamanın hayata geçirilmesi ile birlikte sınav takvimi hazırlamak için harcanan zaman kısalmaktadır. Ayrıca uygulama ile kullanılan verilerin birçoğu doğrudan gerçek zamanlı olarak öğrenci işleri otomasyon sistemi veritabanından alınmakta olduğu için; alttan dersi kalan ya da üstten ders alan öğrenci takibi gibi sınav çakışması sorunları da tamamen ortadan kalkacaktır. Çalışmada mevcut öğrenci otomasyon sistemi veritabanına eklenen tablolar açıklanmakta, ayrıca eldeki verilerin Genetik Algoritma yöntemine uygulanma aşamaları verilmektedir. Uygulama aşamasında kullanılan kısıtlar da programın önemli bir parçasını oluşturmaktadır. Bunun dışında programın arayüzleri örnekleriyle birlikte yer almaktadır. Sonuç olarak, oluşturulmuş sınav programı takviminden bir örnek çıktıya da yer verilmiştir.

Anahtar Kelimeler: *Sınav takvimi otomasyonu, Genetik algoritmalar.*

ABSTRACT

In this study, manually (paperwork) and periodically prepared exam scheduling applications at universities have been taken into a computer automation system by developing a software solution. In the software developed by the authors, Evolutionary Algorithm method has been applied and university administration's specific improvement requests have also been taken into consideration while developing the software. The developed software has two parts: First part is about collecting data through the web application, and the second part is the application project, which calculates the final scheduling results. By utilizing the software, a considerable amount of time lost by manually preparing exam schedules will be saved. In addition, with the real-time connection to the student automation system database, numerous problems will be vanished, i.e. students/classes exam scheduling conflicts, etc. In the study, tables added to the current student automation system database have been explained. Additionally, by applying the genetic algorithm methods to the various parts of data have been examined along with the constraints used in the application, which are essential parts of the software. User interfaces have been designed with their sample instances. Finally, exam-scheduling table has been created and an example output of the schedule has been generated in the study.

Keywords: *Exam scheduling automation, Genetic algorithms.*

* Yazışılan yazar/Corresponding author. E-posta adresi/E-mail address : akyaldir@pau.edu.tr (A. K. Yaldir)

1. GİRİŞ

Sınav takvimleri, çoğu üniversitede akademik veya idari personel tarafından manüel olarak hazırlanmakta, bunun sonucu olarak da çok fazla hatalar oluşmakta, bu hatalara bağlı olarak defalarca düzeltmeler yapılmaktadır. Bu işlemin hızlı ve hatasız sonuçlar döndürecek otomasyon sistemleri ile yapılması kaçınılmazdır. Dolayısıyla, bu çalışmadaki temel amaç manüel sınav programı hazırlamakla kaybedilen zamanı en aza indirerek, olası çakışmaları yerleştirme sırasında yok etmek ve sonradan sınav tarihleri üzerinde oynamaları engellemektir. Başka bir açıdan da bakacak olursak takvimin otomasyon sistemi ile hazırlanması sebebiyle insan ilişkilerinde gereksiz memnuniyetsizlik bildirimlerinin de önüne geçilmiş olacaktır. Otomasyon sistemi kapsamında geliştirilen uygulama kendi içinde birçok kabul edilemez hataları ortadan kaldırmaktadır. Geriye kalan ve lükse giren istekler uygulama tarafından karşılanmayacaktır. Karşılanmayacak bu istekler için öğretim elemanının zorunluluk dışı istemediği sınav zamanını seçebilmesi, sınavlarını bir güne toplamak istemesi gibi özel istekleri örneklendirilebilir. Çalışma bu özellikler doğrultusunda oluşturulup çalışır duruma getirilmiştir.

Çalışmada problemin Genetik Algoritmalar yöntemiyle çözümlenmesi amaç edinildi. Genel olarak bu soruna zaman çizelgeleme problemi de denilebilir. Zaman çizelgeleme problemleri için geliştirilen modeller genelde birbirinden oldukça farklıdır. Bu sebeple literatürde bulunan sonuçları karşılaştırmak zordur. Son yıllarda, birçok araştırmacı çözüm yaklaşımlarını Tavlama Benzetimi, Tabu Arama ve Genetik Algoritmalar gibi yapay zekâ yöntemlerine dayandırmaktadır. Bu çalışmalardan farklı olarak yeni bir yaklaşım olan ve probleme bağımlı olmayan Üst-sezgisel Algoritmalar da kullanılmıştır. Yeditepe Üniversitesi'nde yapılan bir çalışma zaman çizelgeleme problemlerinden ders çizelgeleme problemini Üst-sezgisel Algoritmalar yöntemiyle çözüme ulaştırmıştır. Çalışmada sert ve yumuşak kısıtların (yerine getirilmesi zorunlu olan ve olmayan) her biri için bir sezgisel yazılmış, bu sezgisellerin her biri sadece belirli bir kısıtı düzeltmeye çalışmıştır. Problemin performans değerini etkileyen belirli bir ağırlığı bulunmaktadır. Sert kısıtlar daha

önemli olduğu için yumuşak kısıtlara göre ağırlık değerleri fazladır. Böylece, öncelikle sert kısıtlamaların düzeltilmesi sağlanmış, yumuşak kısıtlamalar da en aza indirilmiştir. Testlerde bütün kısıtların bulunduğu çözüm oranı, en iyi çözüm değerleri ortalamaları ve en iyi zamanlama değerleri ortalamaları deneylerde performans ölçütü olarak kullanılarak çözüme ulaşılmıştır (Kalender, 2007).

Literatürde benzer güncel örnek problemleri modellemek için genel amaçlı birçok zaman çizelgeleme oluşturma dilleri ve araçları da mevcuttur. Bu bağlamda, Tsang ve arkadaşları sınav zaman çizelgeleme problemlerini kısmi memnuniyet problemleri olarak belirten yüksek seviyeli bir dil geliştirdiler (Tsang v.d., 1999). Gaspero ve Schaerf ise lokal arama uygulamalarının genel amaçlı zaman çizelgeleme problemleri üzerine kolay uygulanabilmesi için EASYLOCAL++ adı verilen bir yazılım aracı geliştirmiştir (Gaspero ve Schaerf, 2001). De Werra ve arkadaşları ise derslik-öğretmen zaman çizelgeleme problemleri için basit bir model ve bazı olası uzantılar sunmuştur (De Werra v.d., 2002). Problemin birçok değişkenini NP-tamlayıcı olarak göstermek için hesaplanabilirlik karmaşası da çalışılmış durumdadır. Buna rağmen bu ve buna bağlı konuları seslendirmek için daha fazla çalışma gerekmektedir. Ayrıca, olayı daha iyi kavramak ve güncel sınav zaman çizelgeleme araştırmaları geliştirmek için kurumsal destekli uygulamalar da mevcuttur (Al-Yakoob v.d., 2007). Bu tarz zaman çizelgeleme problem çözümü örneklerinden bir başkası da Süleyman Demirel Üniversitesi'nde yapılan "Hemşire Çizelgeleme Problemlerinin Genetik Algoritmayla Çözümü" adlı çalışmadır. Bu çalışma ile Süleyman Demirel Üniversitesi Araştırma ve Uygulama Hastanesi'nde çalışan hemşirelerle yapılan görüşmeler sonucu problem kısıtları belirlenmiş ve aylık hemşire nöbet çizelgeleri bu kısıtlar üzerinden oluşturulmuştur (Çivril, 2009).

Sınav zaman çizelgeleme problemi ile ilgili geliştirilen algoritmalar da bahsetmekte fayda vardır. Son yıllarda konuyla ilgili birçok kayda değer araştırma yapılmış ve bu araştırmalar Que ve arkadaşları tarafından ayrıntılı bir şekilde taranmış ve irdelenmiştir (Que v.d., 2008). Doğal olarak, Tabu Arama (White ve Xie, 2001; Gaspero, 2002;

Paquete ve Stutzle, 2002), Tavlama Benzetimi (Thompson ve Dowsland, 1998; Merlot v.d., 2003; Burke v.d., 2004), Genetik Algoritmalar (Ross v.d., 1996; Wong v.d., 2002), Taklitçi Algoritmalar (Burke v.d., 1996), Karınca Algoritmaları (Dowsland ve Thompson, 2005; Azimi, 2005) ve Çok-amaçlı Evrimsel Algoritmalar (Mumford, 2007) gibi Üst-Sezgisel Algoritmalar çokça çalışılmıştır.

Tavlama benzetimi zor problemlerin çözümünde iyi performans gösteren sezgisel bir yöntem olup, pek çok değişikliğe sahip fonksiyonların en büyük veya en küçük değerlerinin bulunması ve özellikle pek çok yerel en küçük değere sahip doğrusal olmayan fonksiyonların en küçük değerlerinin bulunması için tasarlanmıştır. Yerel arama teknikleri, mümkün çözümlerin sadece küçük bir kısmıyla, birçok problemi en iyi şekilde çözmeye veya en iyi çözümü verme yeteneğine sahiptir. Burada mümkün çözümlerden kasıt mevcut çözümün komşularının araştırılmasıdır. Bu esnada karşılaşılan problemlerden birisi yerel optimum çözüme takılma sorunudur. Birçok arama tekniğinde, incelenecek komşu seçimi belli bir olasılıkla rastgele seçildiğinden dolayı yerel optimumdan kaçma imkânı bulunabilir. Fakat incelenmiş çözümlere tekrar geri dönme olasılığı olduğundan arama, yerel optimum civarında takılabilir. Bu da aşırı zaman kaybına sebep olur. Bu yüzden daha önceden incelenmiş belli sayıda çözüm bir listede tutulur. Bu listede yer alan çözümler tekrar hesaplamaya katılmadığından aramanın tekrarlanması mümkün değildir. Bu şekilde listelerin oluşturulduğu ve ‘tabu listesi’ adı verilen arama yaklaşımına Tabu Arama Algoritması adı verilmektedir.

Genetik Algoritmalar ise yapay zekânın gittikçe genişleyen bir kolu olan Evrimsel Hesaplama tekniğinin bir parçasını oluşturmaktadır (Elmas, 2003). Evrimsel Hesaplama ilk olarak 1960’larda I. Rechenberg tarafından “Evrimsel Stratejileri (Evolutionstrategie)” isimli eserinde tanıtılmıştır (Rechenberg, 1973). Genetik Algoritmalar, doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir. Karmaşık çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü arar. Genetik Algoritmaların temel ilkeleri ilk kez Michigan Üniversitesi’nde

John Holland tarafından ortaya atılmıştır. İlk olarak Holland evrim teorisi yasalarını Genetik Algoritmalar içinde eniyileme problemleri için kullanmıştır (Holland, 1975). Genetik Algoritmalar problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm kümesi üretir. Böylelikle, arama uzayında aynı anda birçok nokta değerlendirilmekte ve sonuçta bütünsel çözüme ulaşma olasılığı yükselmektedir. Çözüm kümesindeki çözümler birbirinden tamamen bağımsızdır. Her biri çok boyutlu uzay üzerinde bir vektördür (Beasley v.d., 1993a; Beasley v.d., 1993b).

Genetik Algoritmalar problemlerin çözümü için evrimsel süreci bilgisayar ortamında taklit ederler. Diğer eniyileme yöntemlerinde olduğu gibi çözüm için tek bir yapının geliştirilmesi yerine, böyle yapılardan meydana gelen bir küme oluştururlar. Problem için olası pek çok çözümü temsil eden bu küme Genetik Algoritma terminolojisinde nüfus adını alır. Nüfuslar vektör, kromozom veya birey adı verilen sayı dizilerinden oluşur. Birey içindeki her bir elemana gen adı verilir. Nüfustaki bireyler evrimsel süreç içinde Genetik Algoritma işlemcileri tarafından belirlenirler.

Çalışmada ilişkisel veritabanı modeli Microsoft SQL Server 2008 üzerinde oluşturulmuş, kodlama ise Microsoft Visual Studio 2008 üzerinde geliştirilmiştir. Proje ASP.NET ile yazılmış içerisinde LINQ ve AJAX gibi yazılım teknolojileri kullanılmıştır.

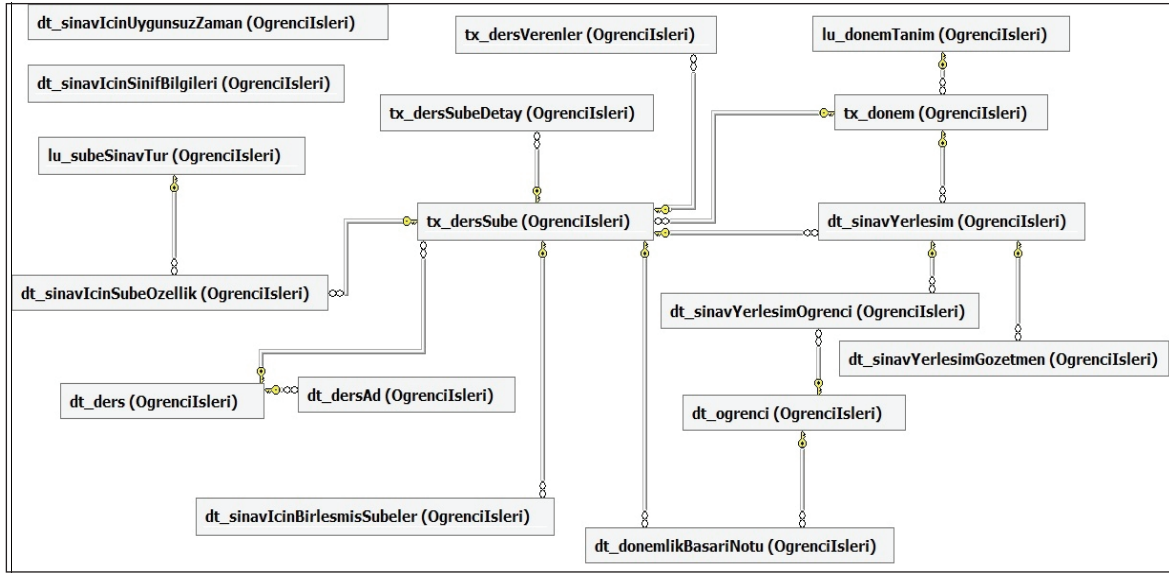
2. SİSTEM TASARIMI

Uygulamada kullanılan tablolar Pamukkale Üniversitesi Pusula Sistemi altyapısında var olan ilişkisel veritabanı modeli esas alınarak oluşturulmuştur. Pusula Sistem Veritabanı Microsoft SQL Server 2008 üzerinde 2008 yılından beri kullanılmaktadır. Veritabanının ana iskeleti de buradaki tablolardır. Uygulama, gereksinim duyulan tabloların mevcut veritabanına eklenmesiyle geliştirilmiştir. Pamukkale Üniversitesinde 2008 yılından itibaren veritabanı yenilenme süreci başlamıştır. Sistem için oluşturulan İlişkisel Veritabanı Modeli Şekil 1’de verilmiştir. Bu yeni sistemde var olan ve bu projede kullanılan tablolar ise kısaca aşağıda açıklanmaktadır.

Sistem veritabanında öğrenim dönemleri 2010 Güz, 2010 Bahar, 2010 Yaz şeklindedir. *lu_donemTanim* tablosunda bu dönemlerin Güz, Bahar ve Yaz şeklinde tanım adları bulunur. *lu_donemTanim* tablosunda bağlı *tx_donem* tablosunda da yıl ve projeye ilgili olmayan *programAnaTur_id* (lisans, yüksek lisans, v.b.) bilgileri tutulmaktadır. *lu_snavTanim* tablosunda ise sınavların türleri yer almaktadır. Burada şu an için sadece final ve ara sınav tanımları bulunur. Üniversite yönetmeliğinde yapılan değişikliklere göre gerekli ilave tanımlar da eklenebilir durumdadır.

dt_ders tablosu ise derslerin kodu ile teorik, pratik, kredi ve AKTS gibi bilgilerini tutan tablodur. *dt_dersAd* tablosu ise *dt_ders* tablosuna bağlı olarak buradaki derslerin isimlerini şu an için sadece Türkçe ve İngilizce olarak iki farklı dilde tutmaktadır.

tx_dersSube tablosu, sınav yerleştirmede ana veri tablosu olacaktır. Sınavlar, bu tablodaki id'ler üzerinden yerleştirilecektir. Tabloda dersler belirli şube numaralarıyla açıldığı dönemde yer almaktadır. Diğer alanlar program için anlamsızdır.



Şekil 1. İlişkisel veritabanı modeli.

tx_dersVerenler tablosu ise açılan şubenin hangi öğretim elemanına bağlı olduğu bilgisini içermektedir. *tx_dersSubeDetay* tablosu ise açılan şubenin hangi bölüme ait olduğu bilgisini tutmaktadır. *dt_programDal* tablosu *tx_dersSubeDetay* tablosunda yer alan *programdal* bilgisini içerir. Bölüm verileri, birçok tablonun bağlanmasıyla oluşmakta olup bu proje için sadece *dt_programDal* tablo verisi yeterli olacaktır.

dt_ogrenci tablosu, öğrenci verilerinin tutulduğu tablodur. Bu tablo, sınav yerleşimi bittikten sonra öğrencilerin dersliklere yerleştirilmesi sırasında kullanılacaktır. *dt_donemlikBasariNotu* tablosu belirlenen dönemlerde hangi öğrencinin hangi dersi aldığını gösterir. Buradan yerleştirilecek

sınava kaç öğrenci gireceği bilgisi alınıp, bu bilgilere göre dersliklere yerleştirme yapılır.

Yukarıda açıklamaları yapılan mevcut tablolara ek olarak geliştirilen proje için aşağıda ayrıntıları verilen tablolar da oluşturulmuştur. Bunların bazıları Pusula sistemindeki veriler tamamlanınca kullanılmayacak olan geçici tablolardır.

Bu tablolardan *lu_subeSnavTur* tablosunda yer alan tanımlar teorik ve uygulama şeklinde iki adettir. Bu tablo, tek bir dersin teorik ve uygulama olarak birden fazla sınavı yapılmak istendiğinde bu sınavları ayırt etmekte kullanılacaktır.

08.00'dan 19.00'a kadar birer saat farkla aralıkların tanımlı olduđu tablo ise *lu_sınavIcinZamanAralik* tablosudur. Bu tablo, dersin sınav süresine bađlı olarak sınavın yerleşim tablosunda yer alacaktır.

dt_sınavIcinSinifBilgileri tablosu ise projede kullanılan geçici bir tablodur. Burada yer alan veriler, farklı tablolardaki tablo bilgilerinin birleştirilmiş halidir. Üniversitenin taşınmazlar projesi hayata geçirildiđi zaman bu tablo verileri doldurulacaktır. Ancak proje içinde bu veriler fonksiyon yardımıyla kullanıldıđı için, veriler tamamlandıđında yapılacak tek deđişiklik fonksiyon içinde; çekilen verinin konumunu deđiştirmek olacaktır. Burada derslik adı, sınav kapasitesi, o derslik için gerekli minimum gözetmen sayısı, dersliđin hangi birimde yer aldıđı (Mühendislik, Fen Edebiyat, v.b.) ve son olarak da laboratuvar sınıfı olup olmadığı bilgisi tutulmaktadır.

dt_birimOgretimGorevlileri tablosu sınavlarda gözetmen olarak yer alacak öğretim elemanlarının listesini tutar. Burada hangi öğretim elemanının hangi programa ve birime bađlı olduđu bilgisi de bulunmaktadır.

dt_sınavIcinSubeOzellik tablosunda derslerin, hangi sınav zamanında, kaç saat süreceđi, yapılıp yapılmayacağı, önceden belirli ise hangi derslikte yapılması gerektiđi ve yapılacaksa ek uygulama sınavı bilgileri yer alır.

dt_sınavIcinBirlesmisSubeler tablosunda ise sınavı beraber yapılmak istenen dersler toplanır. Böylece örnek olarak aynı öğretim elemanının birleştirilmiş dersleri aynı gün ve saate yerleştirilip burada çakışma sorunu ortadan kaldırılmış olur.

dt_sınavIcinUygunsuzZaman tablosunda ise zorunluluk halinde öğretim elemanlarının, hangi dönemde, hangi sınav zamanında, hangi gün ve saatlerde sınav yapamayacağı belirlenir. Program buna göre öğretim elemanlarının sınavlarını belirttikleri tarihlere yerleştirmez.

dt_sınavYerlesim tablosu ise program çalışmasının ilk aşamasını bitirdikten sonra sınav yerleşiminin yazılacağı tablodur. Burada derslerin hangi programa bađlı olduđu (gözetmen atama sırasında kullanılacak), hangi

dönemde, hangi sınav zamanında (ara sınav, final, v.b), hangi derslikte, hangi tarih ve saatte sınavının yapılacağı bilgisi yer almaktadır.

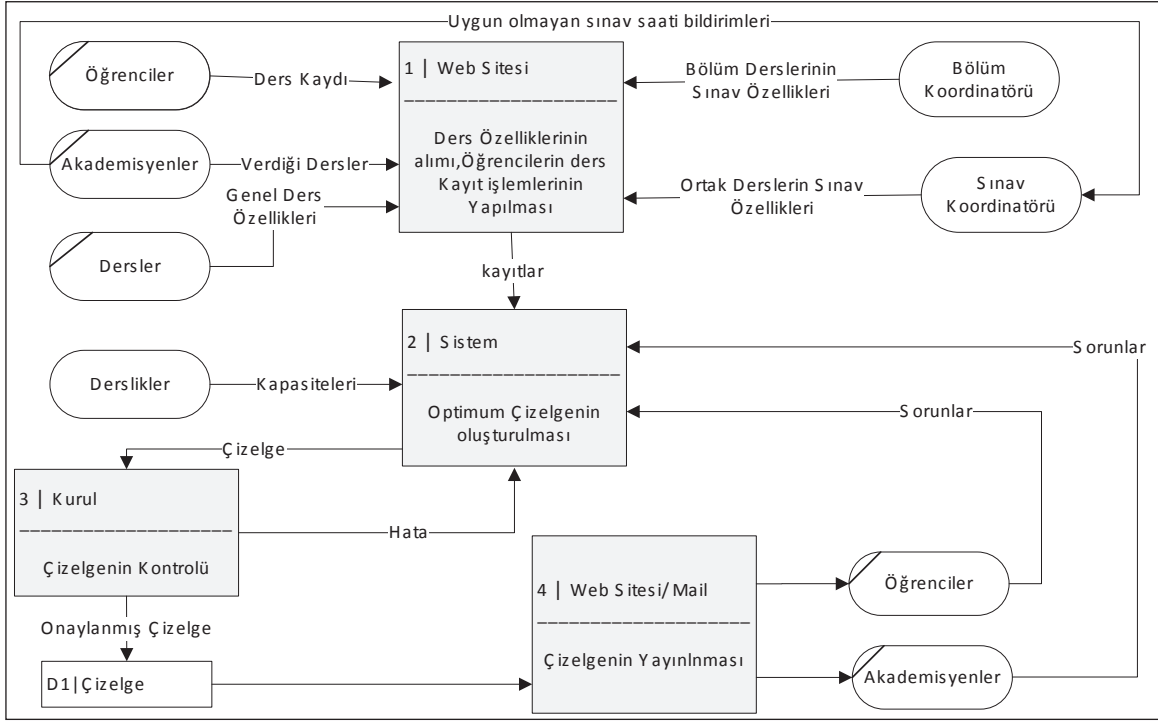
dt_sınavYerlesimGozetmen tablosu ise sınavların yerleşimi bittikten sonra gözetmenlerin belirlenmesiyle doldurulacak tablodur. Bu tablo *dt_sınavYerlesim* ve *dt_nufus* tabloları verilerinin birleşimiyle oluşmaktadır. Bir sınava birden fazla gözetmen atanabileceđi için ayrı bir tablo olarak tasarlanmıştır.

dt_sınavYerlesimOgrenci tablosu da yine *dt_sınavYerlesim* tablosu id'sini alarak bu sınav dersliđinde hangi öğrencilerin sınava gireceđi bilgisini tutmaktadır.

3. GELİŞTİRİLEN UYGULAMA

Sınav programı hazırlayabilmek için öncelikle sınavı yapılacak tüm dersler özellikleriyle birlikte sisteme aktarılmalıdır. Programın bu modülü hangi sınavın kaç saat boyunca süreceđini ya da istenen özel bir derslik olup olmadığı bilgilerini alır. Bu özellikler, bölüm ve fakülte sınav koordinatörleri tarafından web ortamında hazırlanmış arayüzden girilir. Buradaki temel amaç sınavların otomatik diziliminde varsa özel durumların belirtilebilmesi ve programın kalıplaşmış olmaktan kurtarılmasıdır. Veri aktarımı tamamlandıktan sonra uygulama kendi içinde çözüme ulaşmak için çalışmaya başlar. Sisteme ait Genel Bakış Diyagramı Şekil 2'de verilmiştir.

Gerekli ders bilgilerinin toplanması için fakülte ve bölüm koordinatörlerine açılacak ek bir web uygulaması hazırlanmıştır. Bu web uygulaması arayüzü örnekleri Şekil 3, 4 ve 5'de sırasıyla gösterilmektedir. Sınav programı oluşturmada derslerle ilgili öncelikli olarak bilinmesi gereken şey hangi ders şubelerinin (öğrenci sayısına göre ya da farklı öğretim elemanlarının aynı dersi vermesi nedeniyle parçalanmış - farklı zamanlarda işlenen-dersler gibi), hangileriyle beraber yapılmak istendiđidir. Mesela Atatürk İlkeleri ve İnkılâp Tarihi dersi bir fakültenin tüm bölümlerinde farklı gün ve saatlerde, farklı öğretim elemanlarıyla işlenebilmektedir. Ancak genelde tüm fakülte, sınavı tek bir günde gerçekleştirir. Ya da bir öğretim elemanı, aynı dersi normal ve ikinci öğretimden iki ayrı gruba



Şekil 2. Genel bakış diyagramı.

veriyor olabilir, ancak bu dersin sınavını doğal olarak aynı zamanda yapmak isteyebilmektedir.

Bu tarz birleştirmeler yapılmadığı durumda program bu sınavları aynı öğretim elemanına ait olduğu için çakıştırmamaya yani aynı gün ve saate yerleştirmemeye çalışacaktır. Bu durumda öğretim elemanı aynı ders için iki farklı sınav hazırlamak zorunda kalacaktır. İlk olarak, hazırlanacak sınav tipi daha sonra da ders şubeleri görüntülenmek istenen bölümler seçilir ve buradan birleştirilmek istenen ders şubeleri seçilip kaydedilir. Bu ekranda bölüm koordinatörleri için seçilen bölümlerin kodunu taşıyan dersler gösterilir. Fakülte koordinatörü içinse, koordinatörün temsil ettiği fakülteye ait olan, ama bölüm kodu taşımayan, yani MAT, AIT gibi ortak alan dersleri gösterilir. Bu şekilde farklı ekranlara aynı ders düşmesi ve farklı birleştirmeler yapılması engellenmiş olur.

Şube birleştirme işlemi bittikten sonra sınav yapılacak derslerin listesi birleşmiş halleri ile gelir. Bu aşamada da sınavın kaç saat süreceği, isteniyorsa hangi dersliklerde yapılacağı bilgilerinin girilmesi beklenir. Dilim, sınavın süreceği saat bilgisidir. Ayrıca seçili sınav döneminde sınavı yapılmayacak dersler de bu kısımda belirlenir. Son olarak, dersin normal

sınavına ek olarak uygulama sınavı da yapılmak istenirse; uygulama alanı seçilir ve aynı özellikler uygulama sınavı için de doldurulur. Bu işlemlerin yapılacağı arayüzler Şekil 4 ve 5'de verilmiştir.

Ek olarak, sadece fakülte koordinatörünün göreceği ekranda öğretim elemanlarının idari veya görevlendirme gibi nedenlerden dolayı sınav yapamayacakları ya da yapmak istemedikleri günler özel şartlar altında kaydedilir. Bu ekranda arama yapılarak istenen öğretim elemanı seçilir ve bu öğretim elemanının seçili sınav döneminde sınav yapmak istemediği gün ve saatler seçilerek kaydet butonuna basılır. Şekil 6'da ilgili arayüz gösterilmektedir.

Sınav programı çalışmasının temel ekranı bir Windows uygulaması olarak tasarlanmıştır. Bunun en önemli sebebi program çalışma süreci uzun olması beklendiğinden, web ortamındaki zaman aşımı süresi doldu hatasıyla karşılaşmak istenmemesidir.

Oluşturulan Windows uygulaması sadece fakülte koordinatörlerinin bilgisayarlarına kurulacak olup, istenen tüm veriler girildikten sonra sınav takviminin oluşturulması tek bir kişinin sorumluluğunda olacaktır.

Program çalıştırılmadan önce ilk yapılacak işlem, sınav günlerinin belirlenmesidir. Sınav tipi (Ara Sınav, Final, v.b) seçiminden sonra sınav zamanının başlayacağı tarih ve bu zamanın süresi gün olarak girilir. Bu şekilde başlangıç gününden itibaren seçili gün kadar sınav günü listelenir. Eğer liste içinde sınav yapılmak istenmeyen resmi tatil günleri ya da hafta sonu tatilleri varsa çıkartılır.

Sonuç olarak listede programın kullanacağı günler sabitlenmiş olur. Sınavın yapılacağı günlerin seçildiđi arayüz Şekil 7’de sunulmuştur.

Gün seçimi yapıldıktan sonra, program çalışmaya başlatılmadan, belirli gün, saat ve derslikte yapılması özellikle istenen sınavlar varsa bu aşamada bu tür yerleştirmeler de esneklik açısından yapılabilir. Önceden yerleştirme işlemi bittiğinde sınav programı oluştur butonu kullanılabilir hale gelecektir. Durum Şekil 8’de gösterilmektedir.

Programda ilk olarak sınavların yerleşeceği tablo, veri tablosu olarak oluşturulur. Burada başta belirlenmiş günler için seçili fakültenin derslikleri 08.00’den 19.00’a kadar olan zaman dilimi birer saatlik parçalar halinde kullanılır. Bu şekilde *dt_yerlesim* veri tablosu örnek olarak Tablo 1’deki gibi oluşturulur. *dolu* alanı o gün, saat ve derslikte yerleşmiş sınav olup olmadığını anlamak için kullanılacaktır. Bu yerleşim alanlarından, herhangi bir sınava atanan yerin dolu alanı *true* değerine dönüştürülür. Diğer sınavlara atanacak yerler ise *dolu* alanı *false* olan satırlar arasından seçilir.

Tablo 1. Yerleşim planı tablosu.

id	gün	derslik	saat	dolu
1	11. 10. 2010	202	08.00-09.00	False
2	11. 10. 2010	202	09.00-10.00	False
3	11. 10. 2010	203	08.00-09.00	False
4	11. 10. 2010	203	09.00-10.00	False
5	12. 10. 2010	202	08.00-09.00	False
6	12. 10. 2010	202	09.00-10.00	False
7	12. 10. 2010	203	08.00-09.00	False
8	12. 10. 2010	203	09.00-10.00	False
9	12. 10. 2010	204	08.00-09.00	False
10	12. 10. 2010	204	09.00-10.00	False

Eğer önceden sınav yerleştirme alanında yerleştirilmiş sınav varsa; o sınavların yerleştiđi gün, saat ve derslikler bu tabloda yer almaz. Birleştirilmiş şubeler özelliklerinin (öğrencileri gibi) birleştirilmiş haliyle, oluşturulan *Sube* nesnesine atanır. *Sube* nesnesinin özellikleri *sube_id*, şube öğrencileri, şubenin ait olduđu öğretim elemanı, şubenin bađlı bulunduđu program, şubenin sınav süresi, varsa yapılmak istenen özel derslik bilgisi gibi verilerdir. Aynı şekilde önceden yerleştirilmiş sınavlar bu listede yer almaz.

Veriler yerleştikten sonra Evrimsel Hesaplamanın ilk ayađı olan kromozom oluşturulur. Burada kromozom 2 boyutlu tamsayı bir dizidir. Bu dizi tanımı *int [,]. kromozom = new int [şube sayısı, yerleşim sayısı]*. şeklindedir. *kromozom [0,0]*. değeri *Sube* nesnesinin 0. indisindeki şubenin atandıđı yerleşim alanının veri tablosu üzerindeki id’sini verir. Eğer *Sube* nesnesinin 0. indisindeki şubenin sınavı birden fazla saate ya da öğrenci sayısı nedeniyle birden fazla dersliğe yerleşecekse yerleşim tablosunda birden fazla yere atanacak demektir. Bu durumda yerleşim tablosuna yerleşeceği diđer id *kromozom[0,1]*. alanına atanır. *Sube* nesnesinin 0. indis şubesi yerleştikten sonra 1. indisindeki şube yerleşimi *kromozom[1,0]*. alanına atanarak yerleşim devam eder. Program içinde bu şekilde olası çözümlerin atandıđı 10 kromozom oluşturulur. Oluşturulan bu 10 kromozom ilk nesil ebeveynleri temsil eder.

Oluşturulan olası çözümlerin doğru çözümlerden birini verip vermediđini bulabilmek için bir uygunluk fonksiyonu oluşturulur. Burada uygunluk fonksiyonu, olası çözüm içerisinde çakışmalara ve istenmeyen durumlara neden olan yerleşim sayısının ceza puanı olarak hesaplanmasıdır. Bu nedenle önerilen çözüm uygunluk fonksiyonunun 0 olduđu durumlarda bulunmuş demektir. Ceza puanı hesaplanırken 4 ana problem dikkate alınır. Bunlardan ilki aynı gün ve aynı saate yerleştirilmiş sınavların ortak öğrencisinin olmasıdır. Öğrenci çakışması denen bu durumda aynı gün ve saatteki sınavları alan tek bir öğrenci olsa dahi bu sınavlardan biri olası çözümün ceza puanına eklenir.

Pamukkale Üniversitesi

Öğrenci Bilgi Sistemi

SINAV: Final

BİRİM: MÜHENDİSLİK FAKÜLTESİ

PROGRAM: BILGISAYAR MÜHENDİSLİĞİ(2007) **EKLE**

PROGRAMLAR: BILGISAYAR MÜHENDİSLİĞİ
BILGISAYAR MÜHENDİSLİĞİ(2007)

Dersleri Göster

Kod	Ders	Sube No	Görevlendirme	O. Sayı	Seç	
<input type="checkbox"/>	CENG 204	ALGORİTMALAR	1	GÖRHAN GÜNDÜZ	8	<input type="checkbox"/>
<input type="checkbox"/>	BİM 108	AYRIK MATEMATİKSEL YAPILAR	1	NECDET GÜNER	3	<input type="checkbox"/>
<input type="checkbox"/>	CENG 106	AYRIK MATEMATİKSEL YAPILAR	1	NECDET GÜNER	87	<input type="checkbox"/>
<input type="checkbox"/>	BİM 306	BİÇİMSİZ DİLLER VE OTOMATA TEORİSİ	1	EMRE ÇOMAK	40	<input type="checkbox"/>
<input type="checkbox"/>	BİM 442	BİLGİSAYAR GRAFİKLERİ	1	EMRE ÇOMAK	33	<input type="checkbox"/>
<input type="checkbox"/>	CENG 104	BİLGİSAYAR MÜHENDİSLİĞİ SEMİNERİ	1	EVGIN GÖÇERİ	82	<input type="checkbox"/>
<input type="checkbox"/>	BİM 106	BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ - II	1	EVGIN GÖÇERİ	1	<input type="checkbox"/>
<input type="checkbox"/>	BİM 102	BİLGİSAYAR PROGRAMLAMA - II	1	GÖRHAN GÜNDÜZ	3	<input type="checkbox"/>
<input type="checkbox"/>	BİM 104	BİLGİSAYAR PROGRAMLAMA LABORATUVARI - II	1	GÖRHAN GÜNDÜZ	1	<input type="checkbox"/>
<input type="checkbox"/>	BİM 208	BİRLEŞTİRİCİ DİLİYLE PROGRAMLAMA	1	GÖKHAN UÇKAN	8	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	2	ABOULKADIR YALDIR	6	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	4	EMRE ÇOMAK	6	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	8	EVGIN GÖÇERİ	1	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	6	GÖKHAN UÇKAN	4	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	3	GÖRHAN GÜNDÜZ	5	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	5	KADIR KAVAKLIOĞLU	4	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	7	MERİÇ ÇETİN	1	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	1	SEZAI TOKAT	1	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	9	TURAN TOLGAY KIZILELMA	2	<input type="checkbox"/>
<input type="checkbox"/>	BİM 204	DOSYA VE VERİ İŞLEMİ	1	GÖRHAN GÜNDÜZ	20	<input type="checkbox"/>
<input type="checkbox"/>	BİM 206	ELEKTRONİK	1	MERİÇ ÇETİN	18	<input type="checkbox"/>
<input type="checkbox"/>	BİM 340	GENEL EKONOMİ	1	AYDIN SARI	14	<input type="checkbox"/>
<input type="checkbox"/>	BİM 458	İNSAN KAYNAKLARI YÖNETİMİ	1	CELALEDDİN SERİNKAN	27	<input type="checkbox"/>
<input type="checkbox"/>	BİM 406	MESLEK SEMİNERİ - II	1	MERİÇ ÇETİN	33	<input type="checkbox"/>
<input type="checkbox"/>	BİM 304	MİKROİŞLEMCİLER VE MİKROBİLGİSAYARLAR	1	GÖKHAN UÇKAN	40	<input type="checkbox"/>
<input type="checkbox"/>	CENG 102	NESNEYE YÖNELİK PROGRAMLAMA	1	GÖRHAN GÜNDÜZ	91	<input type="checkbox"/>
<input type="checkbox"/>	CENG 208	OLASILIK VE İSTATİSTİK	1	KADIR KAVAKLIOĞLU	14	<input type="checkbox"/>
<input type="checkbox"/>	BİM 210	OLASILIK VE İSTATİSTİK	1	KADIR KAVAKLIOĞLU	27	<input type="checkbox"/>
<input type="checkbox"/>	CENG 202	PROGRAMLAMA DİLLERİ	1	TUĞRUL YILMAZ	10	<input type="checkbox"/>
<input type="checkbox"/>	BİM 202	PROGRAMLAMA DİLLERİ	1	TUĞRUL YILMAZ	21	<input type="checkbox"/>
<input type="checkbox"/>	BİM 334	SAYISAL ELEKTRONİK	1	GÖKHAN UÇKAN	24	<input type="checkbox"/>
<input type="checkbox"/>	BİM 436	YAPAY SINIR AĞLARI	1	SEZAI TOKAT	37	<input type="checkbox"/>
<input type="checkbox"/>	BİM 302	YAZILIM MÜHENDİSLİĞİ	1	TURAN TOLGAY KIZILELMA	34	<input type="checkbox"/>
<input type="checkbox"/>	BİM 404	YÖNEYLEM ARAŞTIRMASI	1	ÖZCAN MUTLU	35	<input type="checkbox"/>

BİRLEŞTİR

SINAVI AYNİ ZAMANDA OLACAK DERS ŞUBELERİ:

Şubeler	
BİM 210 OLASILIK VE İSTATİSTİK şube:1 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	Sİİ
CENG 208 OLASILIK VE İSTATİSTİK şube:1 (253-BİLGİSAYAR MÜHENDİSLİĞİ)	
BİM 202 PROGRAMLAMA DİLLERİ şube:1 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	Sİİ
CENG 202 PROGRAMLAMA DİLLERİ şube:1 (253-BİLGİSAYAR MÜHENDİSLİĞİ)	
CENG 106 AYRIK MATEMATİKSEL YAPILAR şube:1 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	Sİİ
BİM 108 AYRIK MATEMATİKSEL YAPILAR şube:1 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	
BİM 340 GENEL EKONOMİ şube:1 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	Sİİ
BİM 402 BİTİRME PROJESİ şube:1 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	
BİM 402 BİTİRME PROJESİ şube:2 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	
BİM 402 BİTİRME PROJESİ şube:3 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	
BİM 402 BİTİRME PROJESİ şube:4 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	
BİM 402 BİTİRME PROJESİ şube:5 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	Sİİ
BİM 402 BİTİRME PROJESİ şube:6 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	
BİM 402 BİTİRME PROJESİ şube:7 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	
BİM 402 BİTİRME PROJESİ şube:8 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	
BİM 402 BİTİRME PROJESİ şube:9 (233-BİLGİSAYAR MÜHENDİSLİĞİ)	

Şekil 3. Ders birleştirme arayüzü.

Pamukkale Üniversitesi

Öğrenci Bilgi Sistemi

SINAV: Final

BİRİM: MÜHENDİSLİK FAKÜLTESİ

PROGRAM: BİLGİSAYAR MÜHENDİSLİĞİ(2007) EKLE

PROGRAMLAR:

BİLGİSAYAR MÜHENDİSLİĞİ
 BİLGİSAYAR MÜHENDİSLİĞİ(2007)

[Dersleri Göster](#)

Kod	Ders	Sube No	Görevlendirme	O.Sayı	Dilim	Özel Sınıf	Sınav	Uygulama
CENG 204	ALGORİTMALAR	1	GÜRHAN GÜNDÜZ	8	2 dilim	YOK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Uygulama Özellikleri Dilim: 1 Dilim Sınıf: PC-LAB1 (Kapasite:35)
BİM 108	AYRIK MATEMATİKSEL YAPILAR	1	NECDET GÜNER	3	2 dilim	YOK	<input checked="" type="checkbox"/>	
CENG 106	AYRIK MATEMATİKSEL YAPILAR	1	NECDET GÜNER	87	2 dilim	YOK	<input checked="" type="checkbox"/>	
BİM 306	BİÇİMSSEL DİLLER VE OTOMATA TEORİSİ	1	EMRE ÇOMAK	40	2 dilim	YOK	<input checked="" type="checkbox"/>	
BİM 442	BİLGİSAYAR GRAFİKLERİ	1	EMRE ÇOMAK	33	2 dilim	YOK	<input checked="" type="checkbox"/>	
CENG 104	BİLGİSAYAR MÜHENDİSLİĞİ SEMİNERİ	1	EVGIN GÖÇERİ	82	2 dilim	YOK	<input checked="" type="checkbox"/>	

Şekil 4. Ders uygulama sınavı özellik giriř arayüzü.

İkinci problem ise bir öğretim elemanın aynı gün ve saat içerisinde birden fazla sınav yapması durumudur. Bu durumda da aynı güne yerleştirilmiş sınavlardan biri öğretim elemanı çakışması olarak olası çözümün ceza puanına eklenir.

Diđer bir problem de birinci öğretim sınavlarının saat 17.00'den sonra yapılmasıdır. Bu durumda sınav yapacak öğretim elemanına ek mesai ücreti ödenmesi gerektiğinden normal öğretim sınavları 17.00'dan önce bitmelidir. Yine sınav saatlerinden herhangi biri 17.00 ve sonrasında denk gelen normal öğretim sınavları olası çözümün ceza puanına eklenecektir. Son olarak kontrol edilen problem önceden kaydedilen, öğretim elemanının uygun olmadığını belirttiđi saatlere yerleştirilen sınavlarının olması halidir. Bu zamanlara yerleştirilen sınavlar da ceza puanına yazılır. Sonuç olarak burada belirtilen 4 çakışma sorununa da cevap verebilen çözüm, istenen çözümlerden biri olacaktır.

Ebeveynlerden yeni nesil çocuklar üretebilmek için olası çözüm örnekleri olan kromozomlara bazı mutasyonlar uygulanmıştır. Bu mutasyonlar tamamen ceza puanına katkı sağlayan sorunlar üzerinden oluşturulmakla beraber 4 adettir. Mutasyonların adları farklı olsa da mantıkları tamamen aynıdır. Her biri işlem yapacağı kromozomun sorunlu sınavına farklı bir yerleşim id'si atar.

Aynı sırada gidersek ilk mutasyonumuz *ogrenciCakismaMutasyonu* adını alır. Bu mutasyonda ortak öğrencisi bulunan ve aynı gün ve saate atanmış derslerden birinin yerleşim yeri (id'si) değiştirilir. Burada değişiklik yapılan sınavın yeni yerleşim yeri, önceki gün ve saatin dışındaki alandan seçilmektedir. Bu şekilde akıllı mutasyonla çözüme ulaşma hızlandırılmış olmaktadır. İkinci çakışma durumu bir öğretim elemanının aynı günde birden fazla sınav yapması haliydi. Bu nedenle ikinci mutasyona *ogretimElemaniCakismaMutasyonu* adı verildi. Bu mutasyonda da değişiklik

Pamukkale Üniversitesi
Öğrenci Bilgi Sistemi

SINAV: Final
BİRİM: MÜHENDİSLİK FAKÜLTESİ
PROGRAM: BİLGİSAYAR MÜHENDİSLİĞİ(2007) EKLE

PROGRAMLAR: BİLGİSAYAR MÜHENDİSLİĞİ BİLGİSAYAR MÜHENDİSLİĞİ(2007) Dersleri Göster

Kod	Ders	Sube No	Görevlendirme	O.Sayı	Dilim	Özel Sınıf	Sınav	Uygulama	
<input type="checkbox"/>	CENG 204	ALGORİTMALAR	1	GÜRHAN GÜNDÖZ	8	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 108	AYRIK MATEMATİKSEL YAPILAR	1	NECDET GÜNER	3	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CENG 106	AYRIK MATEMATİKSEL YAPILAR	1	NECDET GÜNER	87	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 306	BİÇİMSİZ DİLLER VE OTOMATA TEORİSİ	1	EMRE ÇOMAK	40	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 442	BİLGİSAYAR GRAFİKLERİ	1	EMRE ÇOMAK	33	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CENG 104	BİLGİSAYAR MÜHENDİSLİĞİ SEMİNERİ	1	EVGİN GÖÇERİ	82	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 106	BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ - II	1	EVGİN GÖÇERİ	1	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 102	BİLGİSAYAR PROGRAMLAMA - II	1	GÜRHAN GÜNDÖZ	3	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 104	BİLGİSAYAR PROGRAMLAMA LABORATUVARI - II	1	GÜRHAN GÜNDÖZ	1	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 208	BİRLEŞTİRİCİ DİLİYLE PROGRAMLAMA	1	GÖKHAN UÇKAN	8	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	2	ABDULKADİR YALDIR	6	3 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	4	EMRE ÇOMAK	6	3 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	8	EVGİN GÖÇERİ	1	3 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	6	GÖKHAN UÇKAN	4	3 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	3	GÜRHAN GÜNDÖZ	5	3 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	5	KADIR KAVAKLIOĞLU	4	3 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	7	HERİÇ ÇETİN	1	3 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	1	SEZAR TOKAT	1	3 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 402	BİTİRME PROJESİ	9	TURAN TOLGAY KIZILELMA	2	3 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 204	DOSYA VE VERİ İŞLEMİ	1	GÜRHAN GÜNDÖZ	20	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 206	ELEKTRONİK	1	HERİÇ ÇETİN	18	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 340	GENEL EKONOMİ	1	AYDIN SARI	14	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 342	GENEL İŞLETME	1	CELALETDİN SERİNKAN	9	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 458	İNSAN KAYNAKLARI YÖNETİMİ	1	CELALETDİN SERİNKAN	27	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 406	MESLEK SEMİNERİ - II	1	HERİÇ ÇETİN	33	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 304	MİKROİŞLEMLER VE MİKROBİLGİSAYARLAR	1	GÖKHAN UÇKAN	40	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CENG 102	NESNEVE YÖNELİK PROGRAMLAMA	1	GÜRHAN GÜNDÖZ	91	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CENG 208	OLASILIK VE İSTATİSTİK	1	KADIR KAVAKLIOĞLU	14	1 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 210	OLASILIK VE İSTATİSTİK	1	KADIR KAVAKLIOĞLU	27	1 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	CENG 202	PROGRAMLAMA DİLLERİ	1	TUĞRUL YILMAZ	10	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 202	PROGRAMLAMA DİLLERİ	1	TUĞRUL YILMAZ	21	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 334	SAYISAL ELEKTRONİK	1	GÖKHAN UÇKAN	24	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 436	YAPAY SİNİR AĞLARI	1	SEZAR TOKAT	37	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 302	YAZILIM MÜHENDİSLİĞİ	1	TURAN TOLGAY KIZILELMA	34	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	BİM 404	YÖNEYLEM ARAŞTIRMASI	1	ÖZCAN MUTLU	35	2 dilim	YOK	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Kaydet

Önceki Son

Şekil 5. Ders özellik girişi arayüzü.

Sicil No TC no Ad veya Soyad 0064 Ara

Sicil No	TC No	Adı Soyadı	Birim
Seç 0064	0109277902002	ABDULLAH TAHSİN TOLA	DEVRELER VE SİSTEMLER

UYGUN OLMAYAN ZAMAN SEÇİM:

Tarih: 01.10.2010
Zaman Aralık: 08:00-09:00 Ekle

SINAV İÇİN UYGUN OLMAYAN ZAMANLARIM:

Ad Soyad	Tarih	Zaman Aralık	
ABDULLAH TAHSİN TOLA	01.10.2010	08:00-09:00	Sil
ABDULLAH TAHSİN TOLA	01.10.2010	09:00-10:00	Sil

Önceki Son

Şekil 6. Öğretim elemanı uygun olmayan zaman giriř arayüzü.

Sınav Programı

Sınav Günleri Sınav Programı Oluřturma Gözetmen Seçimi Gözetmen Ekleme Öğrenci Dağılım

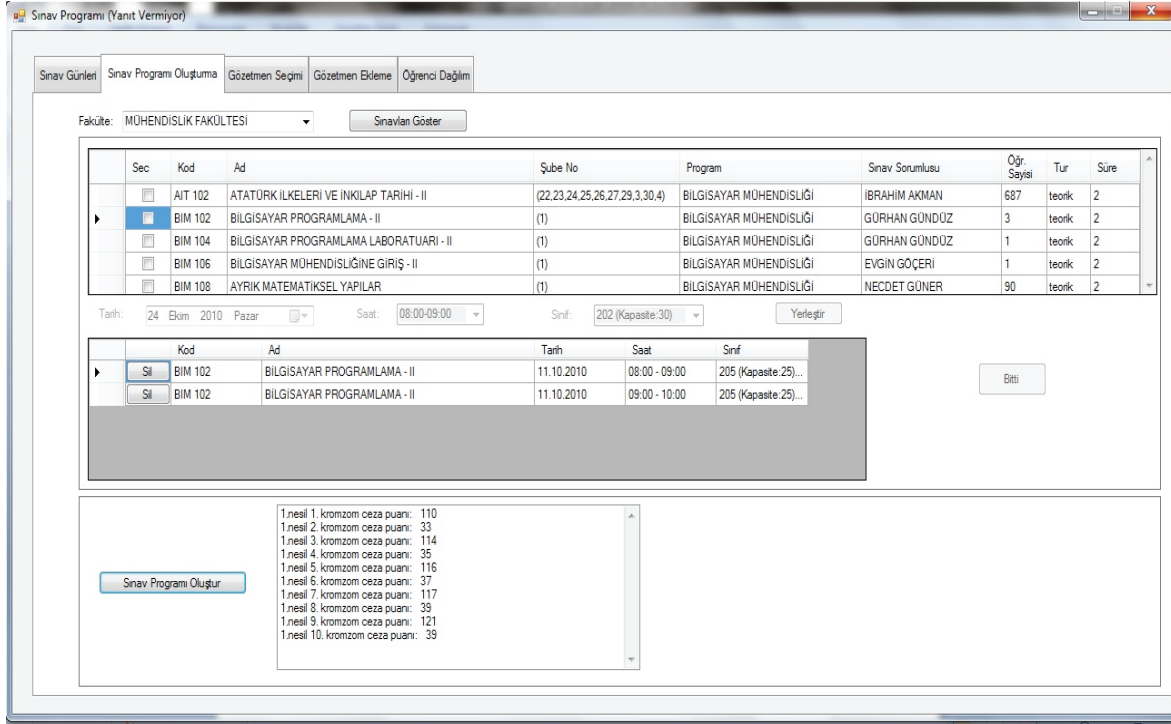
Sınav Tip: Final
Bařlangıç Tarihi: 11 Ekim 2010 Pazartesi
Gün: 12 Tamam

Sınav olacak günler:

- 11 Ekim 2010 Pazartesi
- 12 Ekim 2010 Salı
- 13 Ekim 2010 Çarřamba
- 14 Ekim 2010 Perřembe
- 15 Ekim 2010 Cuma
- 17 Ekim 2010 Pazartesi
- 18 Ekim 2010 Pazartesi
- 19 Ekim 2010 Salı
- 20 Ekim 2010 Çarřamba
- 21 Ekim 2010 Perřembe
- 22 Ekim 2010 Cuma

Tarih Kaldır

Şekil 7. Sınav günü belirleme arayüzü.

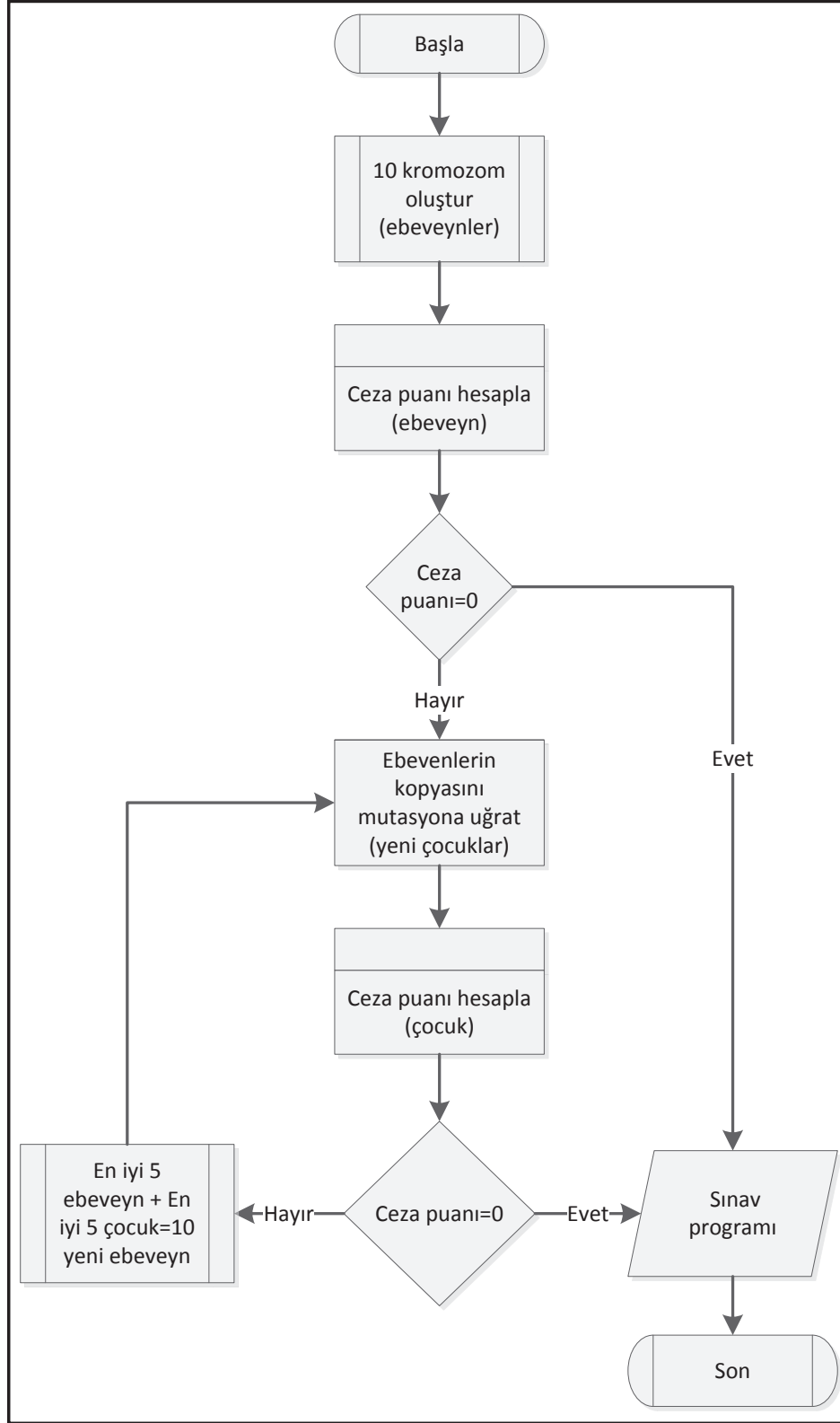


Şekil 8. Sınav programı oluşturma arayüzü.

yapılan sınava önceki yerleşim gününden farklı günlerden biri atanmaktadır. Diğer mutasyonumuz, sınavı 17.00'de ve sonrasına yerleştirilmiş birinci öğretim derslerinin sayısını sıfırlamaya çalışmaktadır. Burada da sorunlu sınavların yerleşimleri, saat 17.00'den önceki yerleşim yerleriyle değiştirilir. Bu mutasyona da *birinciOğretimÇakısmaMutasyonu* denilmiştir. Son mutasyon çeşidi de öğretim elemanlarının, kabul edilebilir nedenlerle istemediği zaman dilimlerine yerleştirilen sınavlarını, değiştirmeye yöneliktir. *uygunsuzZamanMutasyonu* denen bu mutasyonda da değişecek sınav zamanı, öncekinden farklı olan yerleşim alanları arasından seçilir. Ebeveyn oluşumundan sonra, ilk çocuklar bu ebeveynlerin mutasyona uğratılmasıyla oluşturulur. 10 ebeveyn kromozomunun mutasyonlarla 10 çocuk kromozomu olur. Bu kısımda çocuk kromozomlarının her biri uygunluk fonksiyonuna sokulur. Buradan istenen sonuç elde edilmezse; bir döngü oluşturulur ve burada, oluşan 10 çocuk ve 10 ebeveynin en

iyi 5'er elemanı yeni nesle geçer. Bu şekilde uygunluk fonksiyonu değeri en düşük olan 10 kromozom, yeni nesli oluşturmuş olur. Bu yeni nesil döngü içindeki yeni ebeveynlerdir. Daha sonra bu ebeveynlerin her biri tekrar mutasyona uğratılarak yeni çocuklar oluşturulur. Çözüm bu çocuklar arasında da değil ise, yeni nesil ebeveyn ve çocukların en iyileri yani uygunluk fonksiyonu en düşük olanlarından yeniden oluşturulur. Bu döngü uygun çözüm bulunana kadar sürer. Her mutasyon sonucu kesin olmamakla birlikte bir öncekinden daha iyi sonuçlar çıkmaktadır. Her yeni nesil önceki nesillerin en iyileri arasından seçilmektedir. Bu nedenle çözüme en yakın olası çözüm hiçbir zaman kaybedilmeyerek hep ileriki nesillere taşınır.

Tüm çakışmaların sona erdiği yani ceza puanının 0 olduğu bir çözüm bulunduğu döngüden çıkılarak çözüm veritabanına yazılır. Bu süreçle ilgili ayrıntılı Veri Akış Diyagramı Şekil 9'da verilmiştir.



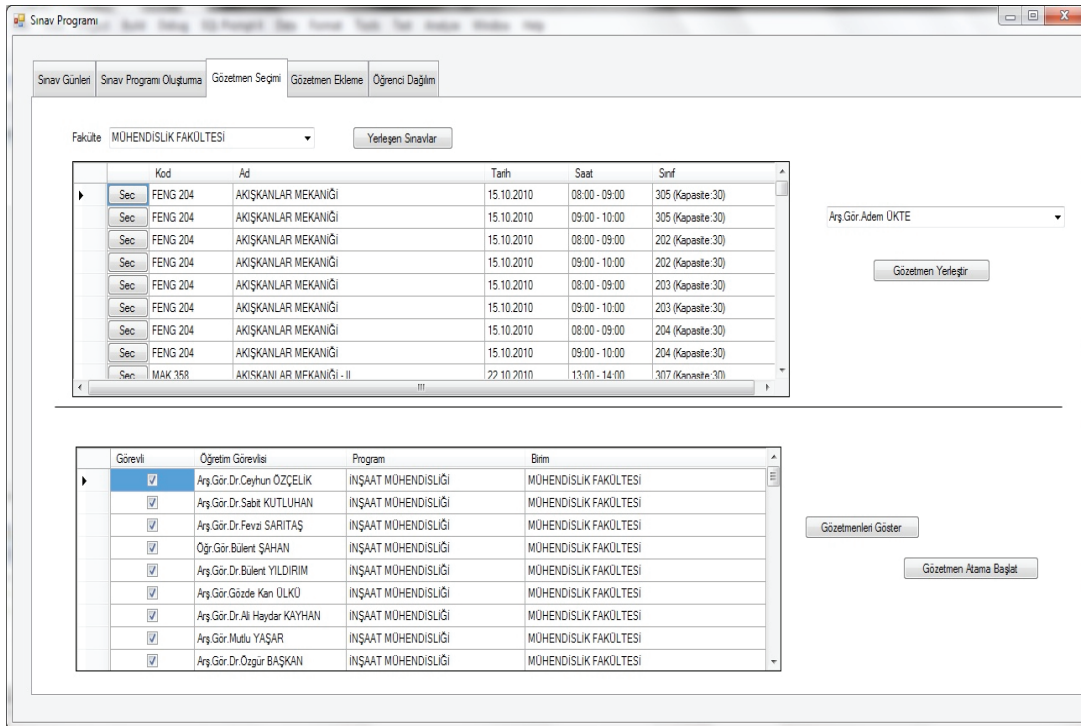
Şekil 9. Veri akış diyagramı.

Sınavların hangi gün ve saatte hangi derslikte yapılacağı yerleştirildikten sonra, sıra gözetmen belirleme işlemine gelmiştir. Bu üçüncü kısımda istenirse belirli gözetmenler, belirli sınav dersliklerine önceden yerleştirilebilirler. Ancak sınav saati ne kadar ise gözetmenin, sınavın dersliğine o süre boyunca yerleştirilmesine dikkat edilmelidir. Önceden gözetmen yerleştirme işlemi bittikten sonra *Gözetmen Göster* butonuyla seçili fakülteye bağlı gözetmenlik yapacak öğretim elemanlarının listesi görüntülenir. Burada tüm öğretim elemanları görevli konumunda gelir ancak o sınav döneminde raporlu olan ya da sınavlara giremeyecek olan kişilerin görevli kısmındaki imler kaldırılır. Bu şekilde gözetmen atama düğmesine tıklandığında önceden yerleştirilmiş sınav sayısı, listede seçili olan gözetmen sayısına bölünüp kişi başına düşen görev sayısı bulunarak program başlatılmış olur. Arka planda ilk olarak görevli listesi veri tablosu şeklinde görev sayıları başta 0 olmak üzere atanır. Daha sonra yerleşen sınav listesi sırasıyla taranarak, bunlar görev sayısı tamamlanmamış kendi bölümünün öğretim elemanı varsa öncelik tanınacak şekilde eşleştirilir.

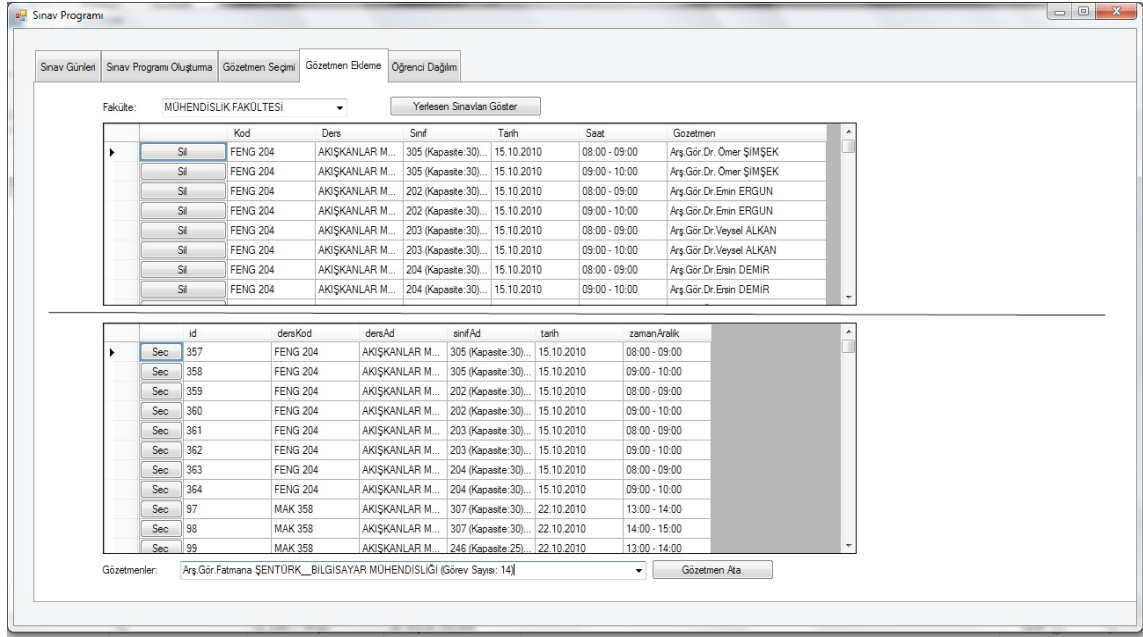
Tüm sınavlara gözetmen ataması yapıldıktan sonra, yerleşen gözetmenlerin aynı gün ve saatte görevlendirmeleri olup olmadığı kontrol

edilir. Bu çakışmalar sıfıra inene kadar bir döngü içerisinde gözetmenler birbiriyle yer değiştirir. Aynı gün ve saatte görevli gözetmen kalmadığında gözetmen atama işlemi bitmiş olur. Bu işlemler de Şekil 10'daki arayüzden yapılmaktadır.

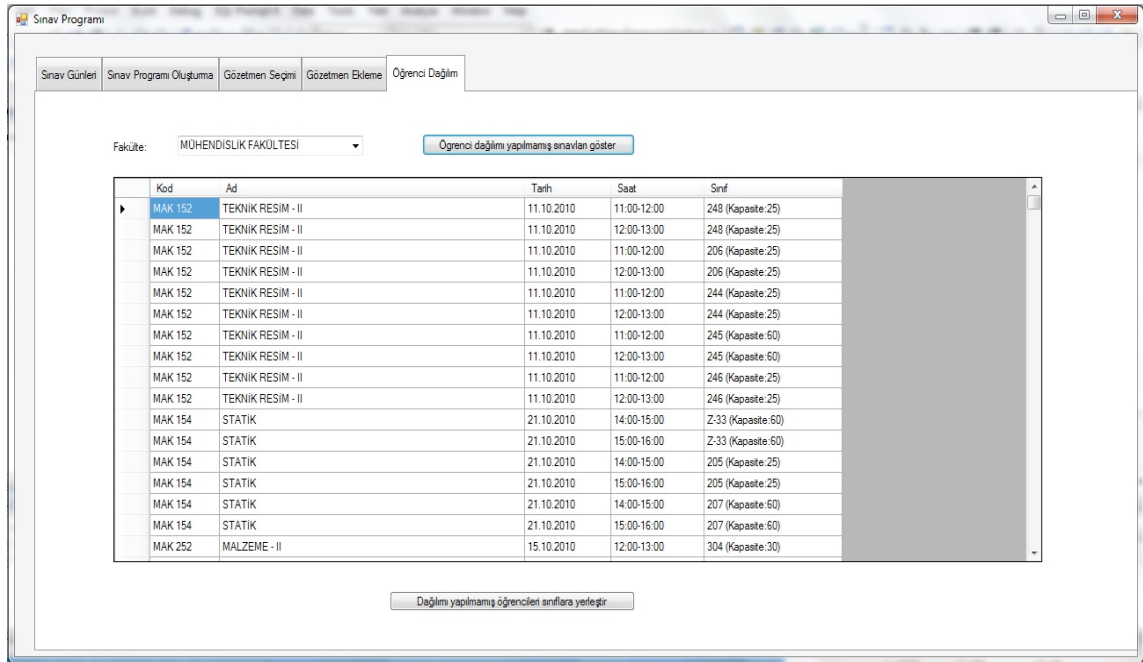
Gözetmen atamaları bittikten sonra istenmeyen durumlar karşısında gerektiğinde gözetmenlerin görevleri birbirleriyle bu alanda değiştirilebilir. İlk listede sınavlar görevli gözetmenleriyle yer almaktadır. Buradan istenen sınav silinerek aynı sınav bir sonraki listeden seçilir ve istenen gözetmenle tekrar eşleştirilebilir. Şekil 11'de gösterilen bu ekranda ayrıca hiçbir gözetmen silinmeden; istenen dersliklere ek gözetmen de atanabilir. Sınavlar zamanları ve gözetmenleriyle birlikte belirlendikten sonra hangi öğrenci hangi derslikte sınava gireceği bilgisi de bu ekrandan elde edilebilir. Burada, fakülte seçimi yapıldıktan sonra henüz öğrencileri yerleştirilmemiş sınavlar listelenir. *Dağılımı yapılmamış öğrencileri sınıflara yerleştir* butonuna tıklandığında program sınavları sırasıyla tarayarak bu sınavlara girecek öğrencileri numara sırasına göre listeler. Dersliklerin kapasitesine göre öğrenci listesinin bir üstünden, bir altından öğrenci almak suretiyle öğrenci yerleşimini gerçekleştirir. İşlemin yapıldığı arayüz Şekil 12'de gösterilmektedir.



Şekil 10. Gözetmen atama arayüzü.



Şekil 11. Gözetmen deđişikliđi yapma arayüzü.



Şekil 12. Öğrenci dağılımı arayüzü.

Çalışma sonucu örnek bir rapor Mühendislik Fakültesi Endüstri Mühendisliđi Bölümü için oluşturulmuştur. Bu şekilde her bölüm için ayrı raporlar gerekli parametreler girildiđi

takdirde alınabilir. Şekil 13'deki raporda hangi dersin sınavının, hangi gün ve saatte, hangi gözetmen eşliđinde yapılacağı rahatlıkla görülebilmektedir.

T.C. MÜHENDİSLİK FAKÜLTESİ ENDÜSTRİ MÜHENDİSLİĞİ BÖLÜMÜ 2009-2010 BAHAR YARIYILI FİNAL SINAV TARİHLERİ					
Sınav Tarihi	Sınav Saati	Ders Kodu	Dersin Adı (Öğr. Sayısı)	Sınav Yeri	Gözetmen
11.10.2010 Pazartesi	13:00 - 15:00	IENG 102	ENDÜSTRİ MÜHENDİSLİĞİ UYGULAMALARI (37)	249	ÇİĞDEM ERSAN ÖNER ATALAY
	15:00 - 17:00	IENG 102	ENDÜSTRİ MÜHENDİSLİĞİ UYGULAMALARI (37)	249	ÇİĞDEM ERSAN ÖNER ATALAY
12.10.2010 Salı	09:00 - 11:00	ENM 212	MÜHENDİSLİK MEKANIĞI (16)	308	BANU YETKİN EKREN HASAN AKYER
13.10.2010 Çarşamba	09:00 - 11:00	ENM 340	VERİTABANI YÖNETİMİ (37)	249	BANU YETKİN EKREN OLCAY POLAT
	15:00 - 17:00	IENG 206	MÜHENDİSLER İÇİN İSTATİSTİK (2)	205	BANU YETKİN EKREN
	17:00 - 19:00	IENG 206	MÜHENDİSLER İÇİN İSTATİSTİK (2)	205	BANU YETKİN EKREN
14.10.2010 Perşembe	08:00 - 09:00	ENM 438	KARAR DESTEK SİSTEMLERİ (29)	202	BANU YETKİN EKREN
	09:00 - 11:00	ENM 102	ENDÜSTRİ MÜHENDİSLİĞİ UYGULAMALARI (37)	249	BANU YETKİN EKREN OLCAY POLAT
		ENM 436	DENEYSEL TASARIM (27)	202	FIGEN TURAN
	11:00 - 13:00	ENM 102	ENDÜSTRİ MÜHENDİSLİĞİ UYGULAMALARI (37)	249	BANU YETKİN EKREN OLCAY POLAT
		ENM 436	DENEYSEL TASARIM (27)	202	FIGEN TURAN
	15:00 - 17:00	ENM 344	DIŞ TİCARET (35)	Z-32	BANU YETKİN EKREN OLCAY POLAT
17:00 - 19:00	ENM 344	DIŞ TİCARET (35)	Z-32	BANU YETKİN EKREN OLCAY POLAT	
15.10.2010 Cuma	13:00 - 15:00	ENM 306	ÜRETİM PLANLAMA VE KONTROLÜ (48)	207	BANU YETKİN EKREN OLCAY POLAT
	15:00 - 17:00	ENM 306	ÜRETİM PLANLAMA VE KONTROLÜ (48)	207	BANU YETKİN EKREN OLCAY POLAT
18.10.2010 Pazartesi	09:00 - 11:00	ENM 210	TEMEL ÜRETİM YÖNTEMLERİ (15)	206	BANU YETKİN EKREN
	11:00 - 13:00	ENM 342	TAM ZAMANINDA ÜRETİM VE UYGULAMALARI (45)	Z-33	BANU YETKİN EKREN OLCAY POLAT
	13:00 - 15:00	ENM 208	MÜHENDİSLİK EKONOMİSİ (12)	206	BANU YETKİN EKREN
	15:00 - 17:00	ENM 208	MÜHENDİSLİK EKONOMİSİ (12)	206	BANU YETKİN EKREN
	17:00 - 19:00	ENM 210	TEMEL ÜRETİM YÖNTEMLERİ (15)	206	BANU YETKİN EKREN
19.10.2010 Salı	11:00 - 13:00	ENM 204	ERGONOMİ (40)	250	BANU YETKİN EKREN OLCAY POLAT
	15:00 - 17:00	ENM 202	YÖNEYLEM ARAŞTIRMASI - I (11)	244	BANU YETKİN EKREN
	17:00 - 19:00	ENM 202	YÖNEYLEM ARAŞTIRMASI - I (11)	244	BANU YETKİN EKREN
20.10.2010 Çarşamba	08:00 - 09:00	CEKO 206	ENDÜSTRİYEL PSİKOLOJİ (2)	206	OLCAY POLAT
	09:00 - 10:00	SBR 161	YÜZME (9)	206	OLCAY POLAT
	09:00 - 11:00	ENM 106	PROGRAMLAMAYA GİRİŞ (1)	244	BANU YETKİN EKREN
		ENM 442	FINANSAL YÖNETİM (18)	206	OLCAY POLAT
	11:00 - 13:00	ENM 106	PROGRAMLAMAYA GİRİŞ (1)	244	BANU YETKİN EKREN
		ENM 442	FINANSAL YÖNETİM (18)	206	OLCAY POLAT
	13:00 - 15:00	ENM 444	E - TİCARET (23)	206	OLCAY POLAT
	15:00 - 17:00	ENM 444	E - TİCARET (23)	206	OLCAY POLAT
	17:00 - 19:00	SBR 161	YÜZME (9)	206	OLCAY POLAT
21.10.2010 Perşembe	08:00 - 09:00	CENG 192	GÖRSEL PROGRAMLAMA (19)	244	RAMAZAN HAKAN ÖZCAN
			GÖRSEL PROGRAMLAMA (25)	208	RECEP YURTSEVEN
			GÖRSEL PROGRAMLAMA (30)	305	ARZUM ULUKÖY
	09:00 - 11:00	ENM 206	İSTATİSTİK (12)	Z-32	OLCAY POLAT SEMIH COŞKUN
			İŞ GÜVENLİĞİ VE İŞ HUKUKU (30)	202	OLCAY POLAT
11:00 - 13:00	ENM 404	İŞ GÜVENLİĞİ VE İŞ HUKUKU (30)	202	OLCAY POLAT	
22.10.2010 Cuma	08:00 - 09:00	ENM 302	İŞ ETÜDÜ (46)	Z-33	SEMIH COŞKUN TURAN TOLGAY KIZILELMA
	09:00 - 11:00	ENM 402	TAHMİN TEKNİKLERİ (35)	249	ENGİN TAN TURAN TOLGAY KIZILELMA
	11:00 - 13:00	ENM 402	TAHMİN TEKNİKLERİ (35)	249	ENGİN TAN TURAN TOLGAY KIZILELMA

SINAV GÖZETMENLERİNİN ORTAK ZORUNLU DERSLERİN SINAV SAATİNDEN EN AZ 10 DAKİKA ÖNCE MİSAFİR ÖĞRETİM ÜYESİ VE SINAV KOORDİNASYON ODASINDA TOPLANMALARI GEREKMEKTEDİR.

Şekil 13. Yerleşen program örnek çıktısı.

4. SONUÇ VE ÖNERİLER

Çalışma denemeleri, örnek uygulama alanı olan Pamukkale Üniversitesi Mühendislik Fakültesi'nde 2009–2010 Öğretim Yılı Bahar dönemi final programı için yapılmıştır. Uygulamayı benzerlerinden ayıran en önemli özellik ise geliştirilen sistemin Pamukkale Üniversitesi Öğrenci İşleri ve Personel Bilgi Sistemi ile veri bütünlüğü sağlayarak çalışıyor olmasıdır. Program eldeki verilerle tek bir fakülte üzerinde denenmiş olsa da, yapısı itibarıyla gerekli veriler tamamlandığında tüm üniversite için çalıştırılabilir duruma getirilmiştir.

Projenin üniversite bünyesinde uygulanması için her fakülte-yüksekokul-meslek yüksek okulu gibi akademik birimler için birer birim sınav koordinatörü ve bu birimlerdeki her bölüm için de birer bölüm sınav koordinatörü görevlendirilmesi gerekmektedir. Bölüm sınav koordinatörleri sınav döneminden önce sınav birleştirme ve sınav özellikleri girme işlemlerini bitirmelidirler. Aynı şekilde birim sınav koordinatörleri de birimlerinin ortak alan dersleriyle ilgili işlemlerini yapmalıdırlar. Bu işlemler bitiminde birim sınav koordinatörleri kendi birimleri için bu kademeli sınav programı oluşturma programını çalıştırarak, kendi birimlerine ait sınav takvimlerine ulaşabilirler.

Çalışma, Pamukkale Üniversitesi Öğrenci Bilgi Sistemi veritabanında yer alan tabloların ve ders şubelerine erişilebilecek bir view'in eklenmesi ile diğer üniversitelerin öğrenci bilgi sistemlerine de rahatlıkla entegre edilebilir.

Bu çalışmada, problemin analizi sırasında programı kullanacak kişilerle görüşülüp bu kişilerin farklı isteklerini gerçekleştirebilecek bir bütünlük içine sokmak karşılaşılan ilk zorluk olmuştur. Bu süreç tamamlandıktan sonra problemin Genetik Algoritmaya ne şekilde uygulanacağı da zaman alan problemlerden biri olmuştur. Ölçütlerin belirlenip uygunluk fonksiyonunun oluşturulması sıkıntı olmasa da kromozom yapısının oluşturulması düşündürücü bir aşamayıdır.

Çalışma, sınavları, bu sınavlarda görev alacak gözetmenleri, sınavların yapılacağı derslikler ile hangi öğrencilerin hangi derslikte

sınava gireceđi gibi verileri oluşturmaktadır. Sınavlar çakışma olmayacak şekilde rastgele yerleştirilmekte ve listeler oluşturulmaktadır.

Uygulamaya ek olarak sınavlara zorluk dereceleri de verilip gün içindeki sınav saatleri bu derecelere göre düzenlenebilirdi. Ayrıca ileride üniversitedeki derslik oturma planları çıkarıldıktan sonra ÖSYM sistemine benzer şekilde hangi sıraya hangi öğrencinin oturacağı da düzenlenebilir ve oturma planları dersliklere önceden asılabilir.

Bu çalışma ile Genetik Algoritmalar ile yapılan sıradan zaman çizelgeleme problemlerine, Pamukkale Üniversitesi Öğrenci İşleri Otomasyonu veritabanı verilerine göre şekillenmiş yeni bir çözüm tekniđi eklenmiş oldu.

5. KAYNAKLAR

Al-Yakoob, S.M., Sherali, H.D. and Al-Jazzaf, M. 2007. "A mixed-integer mathematical modeling approach to exam timetabling," *Computational Management Science*, 2007.

Azimi, Z.N. 2005. Hybrid heuristics for examination timetabling problem, *Applied Mathematics and Computation*, Vol. 163, No. 2, pp. 705–733, 2005.

Beasley, D., Bull, D.R. and Martin, R.R. 1993a. An Overview of Genetic Algorithms: Part 1, *Fundamentals*. *University Computing*, Vol.15 (2), pp. 58-69.

Beasley, D., Bull, D.R. and Martin, R.R. 1993b. An Overview of Genetic Algorithms: Part 2, *Research Topics*. *University Computing*, Vol. 15 (4), pp. 170-181.

Burke, E.K., Newall, J. and Weare, R.F. 1996. A memetic algorithm for university exam timetabling, *The Practice and Theory of Automated Timetabling* (E. K. Burke and P. Ross, eds.), Vol. 1153 of *Lecture Notes in Computer Science*, pp. 241–250, Springer.

Burke, E.K., Bykov, Y., Newall, J. and Petrovic, S. 2004. A time-predefined local search approach to exam timetabling problems, *IIE Transactions*, Vol. 36, pp. 509–528.

- Çivril, H. 2009. Hemşire Çizelgeleme Problemlerinin Genetik Algoritma ile Çözümü, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Isparta.
- De Werra, D., Asratian, A.S. and Durand, S. 2002. Complexity of some special types of timetabling problems, *Journal of Scheduling*, Vol. 5, pp. 171–183.
- Dowland, K.A. and Thompson, J. 2005. Ant colony optimization for the examination scheduling problem, *Journal of the Operational Research Society*, Vol. 56, No. 4, pp. 426–438.
- Elmas, Ç. 2003. Bulanık Mantık Denetleyiciler, Birinci Baskı, s. 24, Seçkin Yayınevi, s. 225, Ankara.
- Gasparo, L.D. and Schaerf, A. 2001. “Tabu search techniques for examination timetabling”, *Practice and Theory of Automated Timetabling: Selected Papers from the 3rd International Conference*. (E. K. Burke and W. Erben, eds.). Vol. 2079 of *Lecture Notes in Computer Science (LNCS)*, pp. 104–117.
- Gasparo, L.D., 2002. “Recolour, shake and kick: A recipe for the examination timetabling problem”, *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling*. (E. K. Burke and P.D. Causmaecker, eds.), (KaHo St.-Lieven, Gent, Belgium), pp. 404–407.
- Holland, J. 1975. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan
- Kalender, M. 2007. Ders Çizelgeleme Programı, TMMOB Elektrik Mühendisleri Odası İstanbul Şubesi, İstanbul.
- Merlot, L.T.G., Boland, N., Hughes, B.D. and Stuckey, P.J. 2003. “A hybrid algorithm for the examination timetabling problem”, *Practice and Theory of Automated Timetabling: Selected Papers from the 4th International Conference*., vol. 2740 of *Springer Lecture Notes in Computer Science*, pp. 207–231.
- Mumford, C.L. 2007. “An order based evolutionary approach to dual objective examination timetabling”, *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007)*, Honolulu, Hawaii 1-5th April.
- Paquete, L. and Stutzle, T. 2002. “Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem”, *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling*. (E. Burke and P. D. Causmaecker, eds.).
- Qu, R., Burke, E.K., McCollum, B., Merlot, L. and Lee, S. 2008. “A survey of search methodologies and automated system development for examination timetabling”, *Journal of Scheduling*.
- Rechenberg, I. 1973. *Evolution Strategy: Optimization of Technical Systems According to the Principles of Biological Evolution*, Frommann-Holzboog, Stuttgart.
- Ross, P., Corne, D. and Terashima-Marn, H. 1996. “The phase transition niche for evolutionary algorithms in timetabling”, *Selected papers from the First International Conference on the Theory and Practice of Automated Timetabling (PATAT 95)* (E. K. Burke and M. A. Trick, eds.), Vol. 1153, pp. 309-324, *Lecture Notes in Computer Science*, Springer-Verlag, NY, 1996.
- Thompson J. and Dowland, K. 1998. A robust simulated annealing based examination timetabling system, *Computers & Operations Research*, Vol. 25, pp. 637–648.
- Tsang, E., Mills, P. and Williams, R. 1999. “A computer aided constraint programming system”, *The 1st International Conference on the Practical Application of Constraint Technologies and Logic Programming (PACLPL)*, pp. 81–93.
- White, G.M. and Xie, B.S. 2001. “Examination timetables and tabu search with longerterm memory,” *Practice and Theory of Automated Timetabling: Selected Papers from the 3rd International Conference*. (E. K. Burke and W. Erben, eds.)..
- Wong, T., Cote, P. and Gely, P. 2002. “Final exam timetabling: A practical approach”, *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2002)*. Vol. 2, pp. 726– 731.