

MERKEZİ SINAVLAR İÇİN GENETİK ALGORİTMALAR İLE EN İYİ OTURMA PLANI

(*BEST SEATING PLAN FOR CENTRAL EXAMS USING GENETIC ALGORİHTMS*)

Faruk BULUT¹, Şeyda SUBAŞI²

ÖZ

Ülkemizde yüz binlerce kişinin katılımıyla yapılan merkezi sınavlar vardır. Rastgele yapılan oturma planlarında birbirini tanıyan kişilerin aynı sınav salonunda birbirlerine yakın oturmaları muhtemeldir. Bu kişilerin yardımlaşma ihtimali de vardır. Bu da sınavların güvenilirliğinin yitirilmesine neden olmaktadır. Basında çıkan haberler ve sınava girmiş kişilerin itirafları bu gerçeği doğrular niteliktedir. Amacımız birbirini tanıyan kişilerin aynı sınav salonunda sınava girmelerini engellemektir. Engelleme işlemini, Genetik Algoritmaların oturma düzenini hazırlayan yazılım içerisinde kullanılmasıyla yapılmaktadır. Önerilen bu model sayesinde, birbirini tanıyan kişilerin aynı sınav salonuna yerleştirilmesi kısa bir süre içerisinde engellenmiş olacak ve büyük kitlelere yapılan sınavların güvenilirliği bu sayede artmış olacaktır.

Anahtar Kelimeler: genetik algoritmalar, optimizasyon, sınav salonu yerleştirme

ABSTRACT

In Turkey, there are many central exams to which hundred thousands of people attend. In the same examination hall, there might be some people who know each other due to the random seating plans. There is a huge possibility that these people might help each other. This situation decreases the reliability of these exams. The confessions from these people who take these kinds of exams and the news in the media verify this truth. Our aim is to prevent those people who may know each other from being in the same examination hall. The implementation of the prevention process is made by using Genetic Algorithm in the software that arranges the examination halls. Due to this proposed model, the possibility of placing people who know each other in the same hall can be decreased and at the same time the reliability of these exams will be increased.

Keywords: genetic algorithms, optimization, organizing examination halls

¹ Katip Çelebi Üniversitesi, Müh. ve Mimarlık Fakültesi, Bilgisayar Müh. Bölümü, İzmir, faruk.bulut@ikc.edu.tr (sorumlu yazar)

² Orta Doğu Teknik Üniversitesi, Müh. Fakültesi, Bilgisayar Müh. Bölümü, Ankara, subasi.seyda@metu.edu.tr

1. GİRİŞ

Ülkemizde ÖSYM tarafından her yıl ALES, YDS, YGS, LYS, KPSS ve TUS gibi birçok sınav yapılmaktadır. Ayrıca Anadolu Üniversitesi açık öğretim sınavları ile bazı bakanlıkların yaptığı birçok merkezi sınav vardır. Bu sınavlara yüzbinlerce öğrenci katılmaktadır. Bu sınavların hepsinde karşımıza bir problem olarak sınav salonlarındaki oturma planlarının en iyi şekilde oluşturulamaması çıkmaktadır. Rastgele yapılan oturma planlarında arkadaş, akraba ve tanıdık olan bazı insanların birbirleriyle aynı sınav salonunda sınava girebilmektedirler. Bu durum kopya çekme ihtimalini de artırmaktadır. Özellikle nüfus bakımından küçük ilçe ve illerde yapılan sınavlarda bu ihtimal daha da fazladır. Bu da yapılan sınavların güvenilirliğini ve geçerliliğini yitirmesine neden olmaktadır. Basında çıkan haberler ve sınava girmiş insanların sosyal medyadaki itirafları bu gerçeği doğrular niteliktedir. Medyada çıkan bazı haberlerden Türkiye’de gerçekleşmiş bazı sınavlarda, birbirleriyle tanışıklığı olan öğrencilerin aynı sınıfta hatta arka arkaya oturduklarını öğreniyoruz. Geçtiğimiz yıllarda aynı sınav salonunda beş kardeşin arka arkaya sınava girdikleri medyada yer almıştı [1]. Bunun gibi onlarca örneği web sayfalarında ve sosyal medyadaki itiraflarda bulabiliyoruz. Belki de tespit edilmemiş daha fazla örnek vardır. Görüldüğü üzere bu durum sınavların güvenilirliği açısından büyük bir önem arz etmektedir.

Bu çalışmadaki amaç, merkezi sınavlarda kopya çekme olasılığını mümkün olduğu kadar en aza indirmektir. Hedefimiz birbirini tanıma ihtimali olan kişilerin aynı sınav salonlarında sınava girmelerini yazılım içerisinde kullanılan Genetik Algoritmalar (GA) ile önlemektir. Sınava giren kişilerin birbirini tanıma ihtimali değişik durumlar göz önüne alınarak hesaplanabilir. Öncelikle sınava giren iki kişi aynı okuldan mezun ise, akraba ise, aynı semtte yaşıyor ise, aynı iş yerinde çalışıyor ise bu kişiler arkadaş olabilirler ve sınav ortamında yardımlaşabilirler. Önerilen bu model sayesinde yapılacak oturma planı ile daha önceden birbirlerini tanıyan kişilerin aynı sınav salonunda bulunmaları engellenmiş olacaktır. Yazılım içerisinde yukarıda bahsedilen benzer özelliklere sahip bireylerin olabildiğince birbirlerinden uzak sınav salonlarında oturmaları çok hızlı bir şekilde sağlanacaktır. Bu da öğrencilerin birbirlerinden yardım alma ihtimalini ortadan kaldıracaktır. Dolayısıyla büyük kitlelere yapılan sınavların iptal edilmeleri önlenmiş olacaktır.

Ayrıca mevcut sistemde, kullanılan sınav salonlarına rastgele öğrenci yerleştirme işleminde birbirini tanıyan insanların tespit edilmesi istense, her bir sınav salonundaki kişilerin yakınlıkları tek tek kontrol edilmelidir. Bu inceleme işlemi tam kapsamlı arama (*Exhaustive Search*) yöntemiyle gerçekleştirilebileceği için çok fazla işlem süresi gerektirecektir. Ayrıca birbirini tanıyan bu kişilerin tekrar başka salonlara alınması durumunda ise başka tanıdık kişilere yakın oturmaları ihtimal dâhilindedir. Yeniden yapılan oturma düzeni için kontrol işleminin tekrar yapılması gerekecektir. Bu durum özyinelemeli bir şekilde tekrar edebilecek ve çok daha fazla işlem zamanı gerektirecektir. Önerilen GA modelinde ise yüzbinlerce kişiye uygulanan sınavlarda bile istenilen bir oturma düzeni dakikalar içerisinde yapılabilmektedir.

Belirli bir sınav için sınav salonlarına katılımcıların en uygun şekilde yerleştirilmesi amacıyla önerilmiş algoritmik tabanlı bir çalışma şu ana dek literatürde bulunamamıştır. Fakat bir sınıfa belirli sayıda öğrencilerin en uygun şekilde yerleştirilmesi ile alakalı bazı çalışmalar vardır. Bu çalışmalar, sınav salonu düzeneği oluşturmaktan çok öğrencilerin ders başarılarını artırmayı amaçlamaktadır. Ayrıca üniversitelerde birden fazla derse ait sınavların çakışmalar olmaksızın yapılmasıyla alakalı önerilen değişik yöntemler de vardır.

Sınıf oturma düzeni ile ilgili ilk çalışma Shin-ike ve Iima tarafından yapılmıştır [2]. GA yöntemi kullanılarak bir sınıftaki öğrencilerin en uygun şekilde sınıf sıralarına yerleştirilmesiyle ilgili bir model önerilmiştir. Rastgele ya da boy sırasına göre hazırlanan oturma düzeninin öğrenci başarısını düşürdüğünden bahsedilmiş ve bireysel başarılarla toplam sınıf başarısının artırılması için bir yaklaşım öne sürülmüştür. Öğrencilere değişik sorular içeren bir anket uygulanmış ve elde edilen veriler sınıfın düzenlenmesinde kullanılmıştır. Öğrencilerin görme ve işitme duyarlarının durumu, sınıf arkadaşları ile ilişkileri, bireysel özellikleri, hafıza ve zeka durumları göz önüne alınarak yerleşim yapan GA tabanlı çalışmalarını Japonya'daki devlet okullarına uygulamış ve başarılı sonuçlar elde etmişlerdir. Başka benzer bir makalede ise bir okulda bulunan ve aynı sınıfa başlayacak olan birinci sınıf öğrencilerinin en uygun şekilde sınıflara dağıtılması üzerine bir çalışma yapılmıştır [3]. Öğrencilerin özellikleri baz alınarak hem lineer tam sayı programlama ve hem de sezgisel programlama teknikleri kullanılarak en uygun oturma düzeni önerilmiştir.

Bu alanda yapılmış benzer çalışmalardan bazıları da üniversitelerde yapılan sınavların sınav salonlarına en uygun şekilde atanması konusundadır. Bilindiği üzere birden fazla ders alan herhangi bir öğrencinin aynı anda aldığı iki veya daha fazla dersten sınav olamaz. Aynı şekilde bir öğrenci bir günde en fazla bir sınav olabilir. Bu gibi kısıtlar göz önüne alınarak en kısa zaman diliminde sınavların tamamının yapılabilmesi için değişik yöntemler kullanılarak belirli çözümler öne sürülmüştür. Bir çalışmada [4] sınavlar ile ilgili zaman tablosunun (*timetabling*) oluşturulmasında GA tabanlı bir modelin nasıl oluşturulabileceği anlatılmıştır. Bu alanda Qu ve arkadaşları da yaptıkları bir çalışmada önerilmiş tüm yöntemleri kısaca anlatarak listelemiş ve aralarındaki farkları göstermişlerdir [5]. Yedi farklı arama tekniği ile zamanlama problemlerinin nasıl çözüldüğü üç farklı üniversitenin veri kümeleri kullanılarak incelenmiş ve karşılaştırma yapılmıştır.

Üzerinde çalışmamız bu konu çok sayıda kişinin katıldığı sınavlarda, birbirini tanıyan adayların aynı sınıfa ya da yakın sıralara yerleştirilmesini önlemektedir. Bu amaçla problem bir optimizasyon işlemi olarak modellenmiştir. Ayrıca hesaplama süresi açısından hızlı çalışan bir çözüm yöntemi önerilmiştir. Üzerinde çalışılan bu konu Türkiye gibi yüzbinlerce hatta milyonlarca insanın aynı anda tek oturumla sınav olmalarına imkan sağlaması konusunda bir ilk olma özelliği taşımaktadır.

Çalışmamızın geriye kalan kısmı beş bölüm daha içermektedir. Takip eden ikinci bölümde konu ile alakalı ön bilgiler; üçüncü bölümde çalışmada kullanılan gen modeli ve önerilen yöntemler; dördüncü bölümde yazılım modelinin zaman karmaşıklığı analizi; beşinci bölümde elde edilen sonuçlar ile sonraki çalışmaların neler olabileceği konusundaki öneriler ve son bölümde değerlendirmeler yer almaktadır.

2. ÖN BİLGİ

Algoritmik karmaşıklar ve hesaplama süresinin fazlalığı bazı çok işlem gerektiren problemlerin çözümünü zorlaştırmaktadır. Bu durum, hızlı ve kolay çözüm veren yeni yöntemlerin arayışına neden olmuştur. Özellikle kör arama (*blind search*) ve sert (*hard*) optimizasyon teknikleri yerine, son yıllarda yumuşak hesaplama (*soft computing*) ve evrimsel algoritma (*evolutionary algorithm*) kullanımı ön plana çıkmıştır [6]. Evrimsel algoritmaların alt dalı olan genetik algoritmalar da, bu arayışlar içinde önemli bir yer tutmaya başlamıştır. Uygulama başarıları artan ve sürekli geliştirilen genetik algoritmalar, diğer yumuşak hesaplama yöntemleri ile birlikte hibrid (*hybrid*) çözümler içerisinde kullanılmaktadır [7].

Genetik algoritmalar karmaşık düzene sahip problemlerin çözümünü gerçekleştirmek için kullanılan sezgisel araştırma yöntemlerinden biridir. Doğadaki canlıların geçirdiği süreci örnek alır. İyi nesillerin hayatta kalması ve kötü nesillerin yok olması mantığına dayalı olarak çalışır. Bu algortmada, bir önceki nesilden doğan yeni çözüm bireyleri kullanılır. Bu bireyler şartlara uyum sağlayıp yaşamlarını devam ettirir. Ancak yeni bireyler bir önceki nesilden gelen iyi genleri muhafaza edebileceği gibi kötü genleri de barındırabilir. Bu durumda kötü genleri barındıran bireyler yaşamlarını sürdüremeyecektir. GA'nın genel özellikleri şu şekilde sıralanabilir:

- Uygun çözümler için birden fazla *popülasyon* kullanılabilir,
- Önceden var olan çözümlerin özelliklerini kullanıp yeni çözümler üretebilir,
- Önceden var olan bir çözümün düzenini rastgele değiştirip yeni bir çözüm üretebilir,
- Kötü genli çözümleri elenir, iyi genli çözümleri korunur.

Genetik Algoritmalar belirli bir uzunlukta kromozoma sahip olmalı ve uygun çözümleri çoğaltmak için bir uygunluk fonksiyonuna (*fitness function*) sahip olmalıdır. Alternatif çözümleri çoğaltma aşaması çaprazlama (*crossing over*) ve mutasyon (*mutation*) gibi işlemlerle olmaktadır. Yeni oluşan her bir çözüm bireyinin uygun olmaması durumunda ise tamir operatörü (*repair operator*) kullanılır. Tamir edilebilen genler korunurken diğerleri elenir. Genetik algoritmalarının çalışma mantığını oluşturan ilgili bazı terimler şu şekildedir [8]:

2.1. Gen

İçerisinde genetik bilgiyi barındıran, kromozomun yapısında bulunan en küçük birime gen (*gene*) denir. Bilgeleri taşıyan bu genlerin bir araya gelmesiyle çözüm kümesini oluşturan bir dizi yani bir kromozom meydana gelir. Programlama yapısında bu gen yapıları ve içerdiği bilgi programcının tanımlamasına bağlıdır.

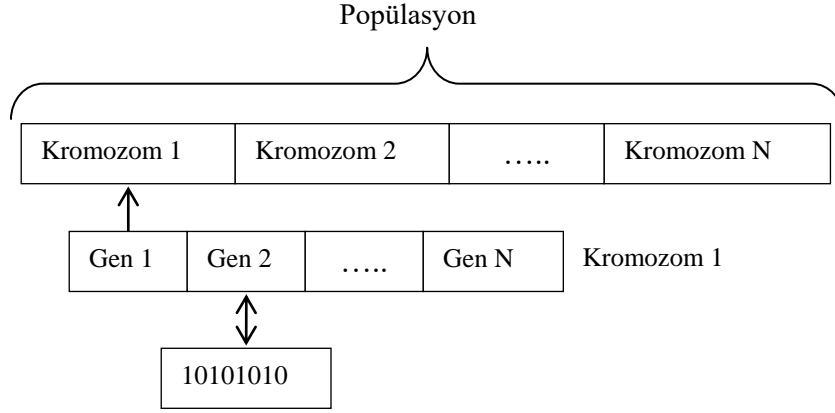
2.2. Kromozom

Bir ya da birden fazla genin bir araya gelerek çözüme ait tüm bilgiyi içeren yapılara diğer bir deyişle dizilere kromozom (*chromosome*) denir. Kromozomlar problemin olası çözüm bilgilerini içerir. Kromozomların yapısı program için büyük bir öneme sahiptir. Bu yapıların içeriği üzerinde çalışılan probleme ve isteğe göre tasarlanabilir.

2.3. Popülasyon

Popülasyon (*Population*), çözüm bilgilerini içeren birçok kromozomun bir araya gelerek oluşturduğu çözüm yığına denir. Bir yığında bulunacak kromozom sayısı programcı tarafından belirlenir. Program sırasında işlemler bu yığınlar üzerinde yapılır. Popülasyonun büyüklüğü programın çözüme ulaşma süresini etkilemektedir. Yığındaki kromozom sayısı çok fazla olduğunda bu süre uzayabilir, çok az olduğunda ise çözüme ulaşmakta sorunlar

oluşabilir. Kromozom sayısı kullanıcı tarafından doğru bir şekilde ve sayıda belirlenmelidir. Şekil 1’de gen, kromozom ve popülasyon yapısı örnek olarak verilmektedir.



Şekil 1. Popülasyon şeması

2.4. Genetik Operatörler

Genetik algoritmalarda çözüme ulaşmak için nesil çeşitliliği sağlamak gerekmektedir. Nesil çeşitliliğini sağlamak için yeni kromozomlar oluşturulurken kullanılan üç temel operatör vardır. Bunlar uygunluk fonksiyonu, çaprazlama ve mutasyon’dur.

2.4.1. Uygunluk Fonksiyonu

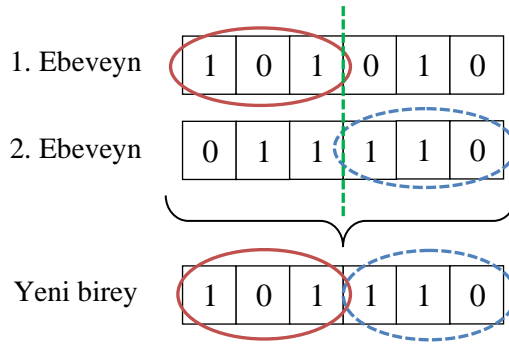
Uygunluk Fonksiyonu, çaprazlama, mutasyon ve diğer aşamalar sonucunda elde edilen kromozomların, çözümde gösterdikleri başarı derecesini belirleyen bir fonksiyondur. Bu fonksiyon, probleme uygun olarak programcı tarafından tasarlanır. Bu sayede hangi hangi olası çözümün yani kromozomun silineceği ya da bir sonraki nesle taşınacağı belirlenir.

Uygunluk değeri bulunurken öncelikle kromozomdaki öğrenci ikililerinin aralarındaki benzerliklere bakılır. Benzerlik ne kadar az ise diğer bir deyişle farklılık ne kadar fazla ise bu kromozomun uygunluk değeri o kadar fazla olur. Öğrenci ikililerinde aynı sıraya denk gelme sayısı benzerliği oluşturur. Hesaplamaya dahil edilecek popülasyondaki öğrenci sayısının N olduğu yerde x kromozomunun F uygunluk değeri şu şekilde hesaplanabilir:

$$F(x) = \frac{\sum_i^N \text{Benzerlikler}}{N} \quad (1)$$

2.4.2. Çaprazlama

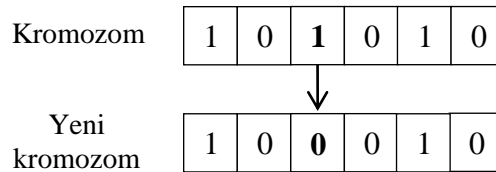
Çaprazlama işlemi, iki kromozomun da yapısından alınan belirli parçaların bir araya gelip yeni kromozomun oluşturması işlemine denir. Şekil 2’de kesim noktası tam ortadan olacak şekilde bir çaprazlama örneği verilmiştir. Çalışmamızda kesim noktasını rastgeleleştirerek bir çaprazlama işlemi yapılmıştır. Bu işlem sonucunda oluşan bozuk genler tamir operatörü ile onarılmaya çalışılmıştır.



Şekil 2. Çapraz değişim şeması

2.4.3. Mutasyon

Genetik Algoritmaların uygulamasında belli bir döngü sürecinden sonra oluşan kromozomlar birbirlerine benzemeye başlarlar. Bu durum optimal ve doğru çözüme ulaşmayı engeller. Kromozomlarda çaprazlama operatörü kullanılsa da ileri nesillerde kromozom çeşitliliği sağlanamaz. Şekil 3'de görüldüğü üzere çeşitliliğin sağlanması amacıyla kromozomlar mutasyon diye isimlendirilen bir işleme tabi tutularak kendi kendini değiştirir. Ancak mutasyon işleminin oranı doğru belirlenmelidir. Yapılan deneysel uygulamalarda bu oranın yüksek alınması çözümlerin doğruluktan uzaklaşmasına ve yanlış çözüm sayısının artmasına sebep olmuştur. Oranın düşük tutulması ise çeşitliliğin oluşmasını engeller. Bu nedenle bu oran üzerinde çalışılan uygulamanın yapısına göre en uygun şekilde seçmelidir. Yapılan değişik uygulamalarda mutasyon oranı gen sayısının en fazla %15'i kadar seçildiği gözlemlenmiştir. Bu uygulamada deneme yanılma yöntemiyle belirlenen en uygun mutasyon oranının %5 olduğu saptanmıştır.



Şekil 3. Mutasyon örneği

2.4.4. Tamir Operatörü (Düzenleyici Algoritma)

GA uygulamalarındaki çaprazlama ve mutasyon işlemlerinde uygunluğu bozulan kromozomların onarılıp tekrar kullanıma hazır hale getirilmesi zorunlu bir işlemdir. Tamir edilen genin tekrar kullanılabilmesi için uygunluk fonksiyonu yardımına ihtiyaç duyulur. Bilindiği gibi uygunluk fonksiyonu, kromozomların çözümde gösterdikleri başarı derecesini belirleyen bir değerlendirme işlevidir. Hangi kromozomların bir sonraki nesle aktarılacağı ve hangi kromozomların yok olacağı uygunluk değerlerinin büyüklüğüne göre karar verilir.

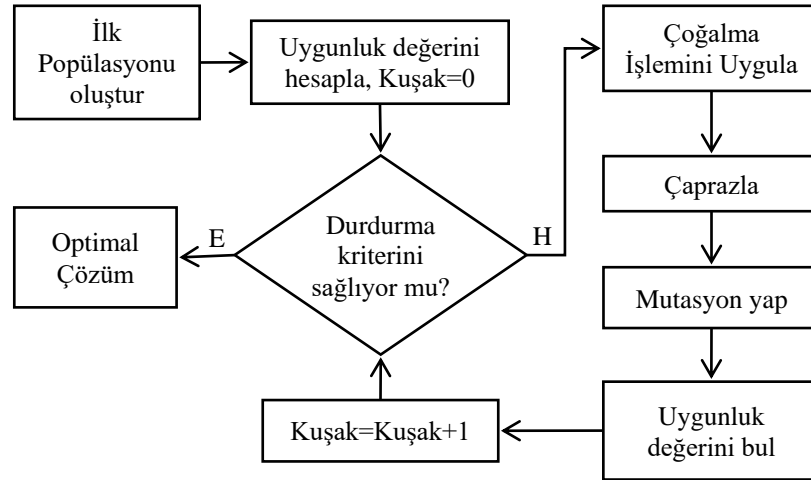
Bu çalışmada oluşan bozuk genler için basit bir tamir operatörü kullanılmıştır. Bu operatörde önce mutasyon tekniği ile kromozomun bir geni değiştirilerek uygunluğu kontrol edilmiştir. Uygunluğu sağlamayan kromozom, en fazla gen sayısının yarısı kadar bir

iterasyonda tamir edilmeye çalışılmıştır. Tamir edilmesine rağmen her iterasyonda uygunluk değeri belirlenen düzeyin altında kalan genler ise popülasyon dışına atılmıştır.

2.5. GA Evreleri

GA'nın çalışma prensibi Şekil 4'de de görüldüğü gibi şu aşamalardan oluşur [9]:

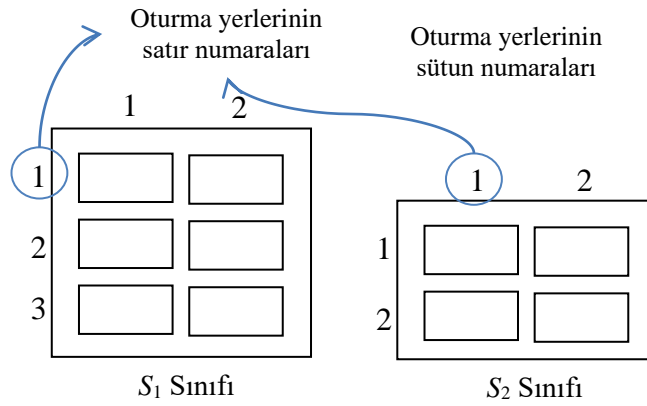
1. Problemin olası çözümlerini barındıran n kromozomdan oluşan rastgele bir popülasyon oluşturulur.
2. Toplumdaki her kromozom için uygunluk değeri hesaplanır.
3. Aşağıdaki adımlar izlenerek yeni toplum üretilir:
 - i. Toplumdan uygunluk değerlerine göre iki kromozom (ata) seçilir. Uygunluk değeri fazla olanın kromozomun seçilme şansı yüksektir.
 - ii. Bu iki kromozom çapraz değişime tabi tutulur ve yavru birey oluşturulur.
 - iii. Mutasyon ihtimaline göre yeni yavru üzerinde mutasyon işlemi uygulanır. Bu işlem her yavru için yapılmak zorunda değildir. Kromozomun taşıdığı genlerin durumuna göre değişmektedir.
 - iv. Oluşan yeni yavru topluma (havuza) eklenir.
4. Eğer aranan çözüme ulaşılmışsa çıktı olarak verilir.
5. Aranan çözüme ulaşılamadı ise yeni oluşan toplum üzerinde yukarıdaki basamaklar yeniden uygulanır.



Şekil 4. GA akış diyagramı

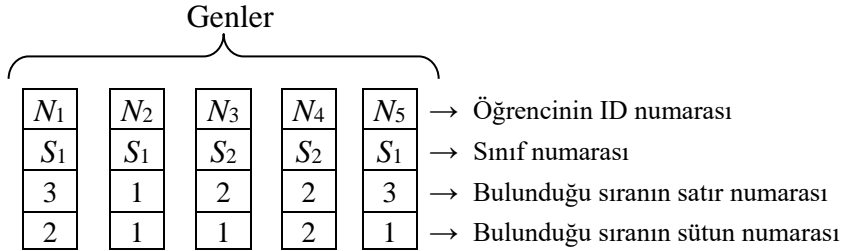
3.YÖNTEM

Genetik Algoritmalar kullanılarak en iyi oturma planının oluşturulması için özgün ve yeni bir model geliştirilmiştir. Önerilen bu modelin tasarımını açıklamak amacıyla temsili bir örnek verilmiştir. N adet öğrenci, belirli kapasitedeki S adet sınıfa, birbirlerini tanıyan öğrenciler aynı sınıfta bulunmayacak şekilde yerleştirilecektir. Örneğin elimizde biri 6 diğeri 4 kişilik kapasiteye sahip iki sınıf ile 10 öğrenci olsun. Sınıfların ID numaraları $\{S_1, S_2\}$, öğrencilerin ID numaraları ise $\{N_1, N_2, \dots, N_{10}\}$ şeklinde olsun. Matris şeklinde düşünülen sınıfların yerleşim planı Şekil 5'de simule edildiği gibi olsun:



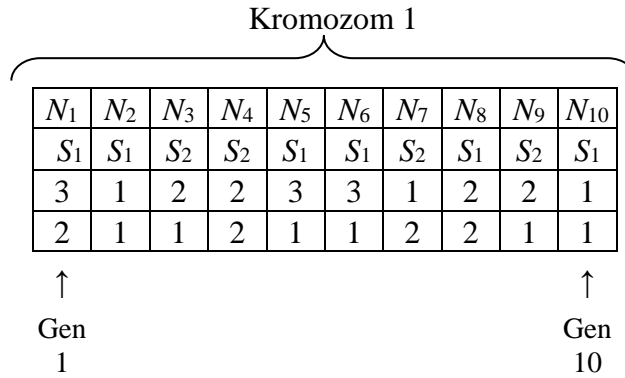
Şekil 5. Sınıfların düzen şeması

10 öğrencinin rastgele yapılmış olan oturma düzenleri Şekil 6'da görüldüğü gibi çalışmamızda gen olarak kullanılacaktır.



Şekil 6. Genlerin yapısı

Şekil 7'de de görüldüğü üzere her öğrenciye ait genin bulunduğu yapı, olası çözümü içeren kromozom olacaktır.



Şekil 7. Kromozomun yapısı

Gen ve kromozom yapılarının uygulama sürecinden önce nasıl belirleneceği önemli bir konudur. Doğru sonuçların en kısa hesaplama süreci içerisinde en az sistem kaynağı kullanılarak elde edilmesi belirlenen bu yapıya bağlıdır.

Yapılan uygulamada öğrencilerin sadece kimlik bilgileri (ad, soyad, okul...) ve sınıf bilgileri (sayısı, mevcudu) kullanılmaktadır. Öğrencilerin birbirlerini tanıma durumları ise aşağıdaki gibi senarize edilsin:

- $N_1 - N_5$ 'in soyadları aynı (akraba olabilirler)
- $N_5 - N_6$ aynı sokakta oturuyor (arkadaş olabilirler)
- $N_6 - N_9$ aynı okulda okuyor (arkadaş olabilirler)

Bu durumları göz önünde bulundurarak en iyi oturma planını oluşturmak için aşamalar adım adım şu şekilde olur:

Adım 1.

Öncelikle 4 tane kromozomdan oluşan rastgele bir popülasyon üretilmiş olsun ve gen havuzuna yerleştirilsin. Örnek olarak üretilen kromozomlar uygunluk değerlerine göre sıralanışı aşağıdaki gibi olsun. Her bir kromozomun uygunluk değerleri ise F ile gösterilmektedir. Uygunluk değeri, kromozomdaki dizide bulunan verilerin bir fonksiyon yardımıyla sıralı bir şekilde incelenmesi ile belirlenmektedir.

Bu uygulama bir eniyileme problemi olduğu için tasarlanan uygunluk fonksiyonu ceza bazlı bir skorlama sistemi ile çalışmaktadır. 0 ceza puanı olan bir kromozom hedef çözümlerden birisi olarak kabul edilmektedir.

Bir kromozom düzeninde diğer bir deyişle sınav salonunda yan yana veya arka arkaya oturan ve birbirleri arasında tanışıklık olan bir çift öğrenci bulunması durumunda bir ceza puanı belirlenmektedir. Bu uygulamada bu ceza puanı -1 olarak belirlenmiştir.

Birbirlerini tanıyan çiftlerden aynı sınav salonda birbirinden uzakta oturanlar için de ayrı bir ceza puanı sistemi geliştirilebilir. Aralarındaki sıra sayısına ters orantılı olarak kromozoma ceza puanı verilerek en iyi oturma düzeninin elde edilmesi hedeflenebilir.

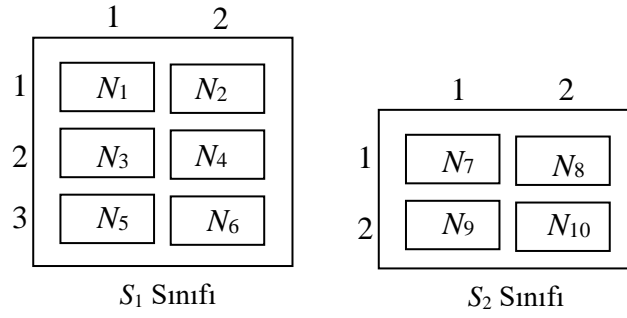
$$F = \frac{-\sum_i^N \text{Ceza Puanı}}{N} \quad (2)$$

Kromozom 1 ($F=1/10$)

N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
S_1	S_1	S_1	S_1	S_1	S_1	S_2	S_2	S_2	S_2
1	1	2	2	3	3	1	1	2	2
1	2	1	2	1	2	1	2	1	1

Şekil 8. Kromozom 1'in yapısı

1. inci kromozomda bulunan genler mevcut sınav salonlarına yerleştirildiğinde şu şekilde bir düzenek elde edilmektedir:



Şekil 9. Kromozom 1'deki sınıfların düzeni

Bu aşamada F uygunluk değerlerine göre çaprazlama işlemleri gerçekleştirilmektedir. Her çaprazlama sonunda yeni bir birey oluşmaktadır. Oluşan her bir yavru bireyin doğru çözüm olup olmadığı kontrol edilmektedir. Eğer doğru çözüme ulaşamadı ise çaprazlama işlemi devam etmektedir. Kontrol işlemi her çaprazlama işlemi sonunda belirlenen bir fonksiyon ile yapılmaktadır. Oturma planında her bir bireyin ön, arka, sağ ve sol yanında oturan başka bir birey arasında yakınlık olup olmadığı kontrol edilmektedir. Bu düzenekte birbirini tanıyan ve yan yana oturan sadece bir çift olduğu için toplam ceza puanı -1; F değeri ise $1/10$ 'dur.

Kromozom 2 ($F=5/10$)

N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
S_2	S_1	S_1	S_2	S_2	S_2	S_2	S_2	S_1	S_2
2	1	1	1	2	2	1	2	2	1
1	1	2	1	2	2	1	1	3	2

Kromozom 3 ($F=3/10$)

N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
S_2	S_1	S_1	S_2	S_2	S_1	S_2	S_1	S_1	S_1
2	1	1	1	2	2	1	3	3	2
1	1	2	1	2	2	2	1	2	1

Kromozom 4 ($F=1/10$)

N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
S_1	S_2	S_1	S_2	S_2	S_1	S_2	S_1	S_1	S_1
1	2	1	2	1	2	1	3	3	2
1	1	2	2	1	2	2	1	2	1

Şekil 10. Kromozomların gösterilişi

Adım 2.

Şekil 10'da da görüldüğü gibi *Kromozom 3* ve *Kromozom 4*'ün çaprazlanması sonucunda oluşan yeni *Yavru Birey 1* görülmektedir. *Kromozom 1*'deki ilk yarısındaki genler ile *Kromozom 2*'nin son yarısındaki genler birleşerek *Yavru Birey 1*'i oluşturmaktadır. Burada yapılan çaprazlama işleminde kesme işlemi örnek olması açısından tam ortadan yapılmıştır.

Kromozom 3									
N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
S_2	S_1	S_1	S_2	S_2	S_1	S_2	S_1	S_1	S_1
2	1	1	1	2	2	1	3	3	2
1	1	2	1	2	2	2	1	2	1

Yavru Birey 1 ($F=2/10$)									
N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
S_2	S_1	S_1	S_2	S_2	S_1	S_2	S_1	S_1	S_1
2	1	1	1	2	2	1	3	3	2
1	1	2	1	2	2	2	1	2	1

Kromozom 4									
N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
S_1	S_2	S_1	S_2	S_2	S_1	S_2	S_1	S_1	S_1
1	2	1	2	1	2	1	3	3	2
1	1	2	2	1	2	2	1	2	1

Şekil 10. Çaprazlanma işlemi

Yavru Birey 1, kromozom havuzuna konularak başka kromozomlarla çaprazlanması beklenmektedir. İlerleyen adımlarda da ideal sonuç bulunana dek çaprazlamalar yapılmakta ve her iterasyonda genetik açıdan bozuk çözüm bireyleri gen havuzundan silinmektedir.

Adım 3.

Aşağıdaki şekilde *Kromozom 1*'in ilk yarısındaki genlerle ve *Kromozom 4*'ün son yarısındaki genlerin çaprazlanması sonucunda oluşan yeni *Yavru Birey 2* görülmektedir. Burada aynı sınav sırasına oturan birden fazla öğrenci olduğu için bu çözümün kabul edilebilir olması imkânsızdır. Bu çözüm çaprazlama için kullanılacağından çözüm havuzuna dâhil edilmemektedir.

Yavru Birey 2									
N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
S_1	S_1	S_1	S_1	S_1	S_1	S_2	S_1	S_1	S_1
1	1	2	2	3	2	1	3	3	2
1	2	1	2	1	2	2	1	2	1

Şekil 11. Adım 3'ün çaprazlanma şeması

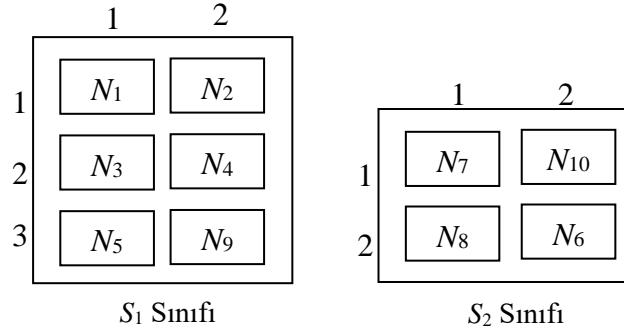
Adım 4.

Aşağıdaki şekilde *Yavru Birey 2*'nin ilk 5 geni ve *Kromozom 2*'nin son 5 geninin çaprazlanması sonucunda oluşan yeni *Yavru Birey 3* görülmektedir. Kontrol fonksiyonu ile genlerin uygunluğu kontrol edildiğinde 0 ceza puanı ile doğru çözüme ulaşıldığı görülmektedir.

Yavru Birey 3									
N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
S_1	S_1	S_1	S_1	S_1	S_2	S_2	S_2	S_1	S_2
1	1	2	2	3	2	1	2	2	1
1	2	1	2	1	2	1	1	3	2

Şekil 12. Adım 3'ün çaprazlanma şeması

Bu aşamada popülasyon ileri aşamalara ulaşmış olmasına rağmen gen yapısı bozuk olan *Yavru Birey 2* ile *Kromozom 2* çaprazlamaya tabi tutulmuştur. Kısaca bir ebeveyn ile bir çocuk arasında çaprazlama yapılmıştır. GA çalışmaları genelde doğada var olan ilişki modelini baz aldığı için bu tarz bir çaprazlama yöntemi doğru sonuca ulaşılmasını engeller. Yapılan uygulamalarda bir sonraki nesle taşınan kromozomlar arasında çaprazlama yapılmıştır. Fakat önerilen modeli en iyi ve en kısa şekilde açıklayabilmek için verilen örnekte kasıtlı olarak bu tür bir hata yapılmıştır. Çaprazlama sonucunda oluşan çözümün sınıf yerleşimi şu şekildedir:



Şekil 13. Sınıf oturma planı

Sınıf oturma planı Şekli-13'deki gibi olur. Öğrencilerin akrabalık, komşuluk ve sınıf arkadaşlığı gibi özelliklerini kontrol ettiğimizde birbirlerini tanıma ihtimali olan hiç kimsenin aynı sınıfta olmadığını görülmektedir.

Çalışmamızda kullanılan GA yönteminin nasıl tasarlandığı ve nasıl uygulandığı örnek olarak verilen bir adet simülasyon verisi üzerinde gösterilmiştir. Önerilen gen yapısında bulunan sınıf sıralarının satır ve sütun numaraları yukarıda verilen senaryoda kullanılmamıştır. Sınava giren kişiler arasındaki tanışıklığın fazla olması ve buna karşılık sınav salonunun az olması durumunda sınıf sıralarının satır ve sütun numaraları işe yaramaktadır. Bu gibi durumlarda tanışıklığı olan kişilerin aynı sınav salonunda bulunmaları gerekecektir. Satır ve sütun numaraları kullanılarak bu kişilerin aynı sınav salonunda birbirlerinden olabileceğince uzak noktalara oturmaları sağlanabilir.

4. KARMAŞIKLIK ANALİZİ

GA modelinin belirlenmesinin ardından inşa edilmesi ile ilgili süreçte kullanıcı tarafından atanması gereken birçok parametre vardır. Olası parametreler şu şekilde listelenebilir:

- Genlerin yapısında hangi faktörlerin bulunacağı,
- Kromozom yapısının nasıl belirleneceği,
- Popülasyonun oluşturulma durumu,
- Kromozomların uygunluk fonksiyonunun ne olacağı,
- Tamir operatörünün nasıl olacağı,
- Çaprazlama işleminin kaç gen ile yapılacağı,
- Mutasyonun hangi aşamada devreye gireceği,
- Gen havuzunun kapasitesinin en fazla kaç kromozon alabileceği.

Hiper parametre sayısının fazlalığı olumsuz bir durum gibi algılansa da programın salt olarak uygulanabilirliği ve işlevselliği çok büyük bir avantaj olarak değerlendirilmelidir. Ayrıca geliştirilen bu modelin en kısa zamanda en iyi oturma planını verebilmesi için programcı seçilen her bir parametre için en uygun değer aralıklarını vermelidir.

Genetik algoritmalar, istenilen hedefe ulaşmak için en uygun imkânları seçen tahmini ve stokastik (*stochastic*) bir süreç içerisinde çalışır. Bu tanım GA'nın kaotik bir karmaşıklığa ve yapıya ait olduğu anlamına gelmez. GA uygulamalarında en iyi sonucun bulunmasına ilişkin net bir zaman tahmininde bulunmak zordur. Ayrıca hesaplama süresinin çok kısa veya beklenenden uzun çıkma durumu söz konusudur [10]. Bu durum Rus ruleti ile benzeşmektedir. Yani tüm olası durumlar içerisinde istenilen çözüme her bir iterasyonda ulaşılabilme ihtimali vardır [11].

Ayrıca önerilen bu modelde zaman karmaşıklığı (*time complexity*) net bir şekilde ifade edilememektedir. Fakat yapılan makale taramasında bu konu ile alakalı değişik yaklaşımlar sunulmuştur [12]. Bilinen en yaygın karmaşıklık formülü büyük O notasyonuna göre şu şekildedir:

$$O(P.G.(O(Fitness)).(P_c.O(crossover)) + (P_m.O(mutation))) \quad (3)$$

Bu formülde P popülasyon, G gen, P_c çaprazlama fonksiyonu ve P_m de mutasyon fonksiyonudur. Karmaşıklık ebetteki genlere ve her bir gen için yapılan hesaplamalara bağlıdır. Burada P , G , P_c ve P_m sabit değerler olduğu için karmaşıklığı şu şekilde sadeleştirmek mümkündür:

$$O(O(Fitness).O(crossover) + O(mutation)) \quad (4)$$

Diğer taraftan sınav oturma düzeninin rastgele yapılması durumunda karmaşıklık analizi incelenecek olursa karşımıza polinomsal olmayan bir durum çıkacaktır. Oluşturulan oturma düzeninin uygunluğu kontrol edildiğinde yazılım tarafından yeni bir oturma düzenine ihtiyaç olduğu söylenecektir. İki veya daha fazla öğrenci arasında tespit edilen yakınlık sonrasında oluşturulacak yeni oturma planında da benzer bir durumun çıkması söz konusudur. Bu durum, tüm olasılıkların tek tek incelenmesini gerektirecektir. N öğrenciye ait oturma olasılıklarının tamamı incelendiğinde permütasyon problemi karşımıza çıkmaktadır. Doğal olarak $O(N!)$ zaman karmaşıklığı içerisinde en uygun oturma, tüm olasılıklar içerisinde çıkarılacaktır. Burada belirtmek isteriz ki zaman karmaşıklığı içerisinde bir parametrede bulunan faktöriyel işareti algoritmanın polinomsal olmadığını (NP , *Non-Polynomial*) göstermektedir [13].

5. SONUÇLAR VE SONRAKİ ÇALIŞMALAR

Önerilen bu sınav salonu yerleştirme modeli Visual Basic'te hazırlanan bir yazılımla test edilmiştir. 2,2 GHz hızındaki, çift çekirdekli, 64 bit işlemcili ve 2 GB RAM'li bir bilgisayar kullanarak değişik senaryolara bağlı olarak elde edilen deneysel sonuçlar Çizelge 1'de verilmiştir. Çizelgedeki N değeri, sınav salonundaki oturak sayısını yani sınava katılan öğrenci sayısını göstermektedir. Sınav salonu sayısı ise $N/20$ olacak şekilde ayarlanmıştır yani her bir salon 20 kişilik öğrenci kapasitesine sahiptir. Ayrıca sınava katılanlar arasındaki birbirini tanıyan çift sayısı $N/10$ olacak şekilde rastgele olarak ayarlanmıştır. Her bir N değerine bağlı olarak denemeler 5'er defa gerçekleştirilmiş ve hesaplama süresinin en iyi, en

kötü ve ortalama değerleri bulunmuştur. Görüldüğü üzere N değeri arttıkça hesaplama süresi de doğru orantılı olarak artmaktadır. Beş yüz bin kişinin girdiği sınavlarda bile en iyi oturma düzeni yaklaşık olarak bir buçuk saat gibi kısa bir zaman diliminde yapılabilmektedir.

Çizelge 1. Algoritmanın N değerine göre hesaplama süreleri

N Öğrenci Sayısı	Sınav Salonu Sayısı ($N/20$)	Tanışık çift sayısı ($N/10$)	Hesaplama Süresi (sn)		
			En iyi	Ortalama	En kötü
100	5	10	0 sn	0 sn	0,01 sn
1.000	50	100	1,2 sn	1,92 sn	4,53 sn
10.000	500	1.000	8,4 sn	27,65 sn	58,1 sn
100.000	5.000	10.000	2 dk 35 sn	8 dk 25 sn	32 dk 31 sn
500.000	25.000	50.000	45 dk 8 sn	1 saat 55 dk	7 saat 13dk

Aynı sınıf sayısında farklı hesaplama süresi değerlerinin çıkması daha önce belirtildiği gibi GA'nın yapısından kaynaklanmaktadır. Yazılımın koşturulması sürecinde kromozomlar rastgele (*random*) oluşturulduğu için nihai çözümünün ne zaman bulunacağı net olarak bilinmemektedir.

Sınav salonlarının düzenlenmesi ile alakalı literatürde yapılan benzer çalışmalar vardır. Fakat bu çalışmalar üniversitelerdeki birden fazla dersin çakışma olmadan aynı anda nasıl yapılacağı üzerinedir. Bu tarz çalışmalar için hazırlanmış ortak kullanıma açık veri setleri vardır. Tek oturumlu ve bir çok katılımcının olduğu sınavlar için hazır veri setleri bulunmadığı için bu aşamada sentetik veri setleri kullanılmıştır.

On binlerce insanın girdiği sınavlarda katılımcıların sınav yeri için ilçe tercihi alınmaktadır. Tam sayı programlama yöntemiyle tercihler doğrultusunda rastgele yerleştirme işlemi bir saniyenin altında elbette ki yapılabilir. Bu çalışmayı karşılaştırmak için aynı sentetik veri setleri üzerinde başka optimizasyon algoritmaları da kullanılabilir. Karınca kolonisi, Yapay Bağışıklık, Tabu Arama, Parçacık Sürü Algoritması, Arı Kolonisi ve literatürde önerilen yeni eniyileme yöntemleri tercih edilebilir.

Prototip olarak sunulan bu çalışma için yeni bir gen modeli tasarlanabilir. Sınava giren birey ve sınav salonunun özellikleri ile ilgili yeni özellikler gen modelinin belirlenmesinde kullanılarak daha detaylı hesaplamaların ve yerleştirme işlemleri yapılması sağlanabilir.

Ayrıca bu çalışmada tanışıklık durumu için 3 temel kıstas göz önüne alınmıştır. Bunlar sınıf arkadaşlığı, akrabalık ve aynı sokakta ya da semtte yaşama durumlarıdır. Sınava giren kişilerin birbirlerini tanıma durumları bazı kurumlardan elde edilen verilerle detaylandırılabilir. Örneğin Nüfuz müdürlüğünden alınan verilerle sınava girenlerin soyisim benzerlikleri olmadan akrabalık (kuzen, teyze, dayı vs.) durumları tespit edilebilir. Ayrıca aynı kurumdan alınan verilerle sınava katılan kişilerin birbirlerine yakın oturup oturmadıkları saptanabilir. SGK kayıtlarından yararlanarak da aynı iş yerinde çalışanlar tespit edilebilir. Ayrıca Milli Eğitim Bakanlığında alınan verilerle mezun olunan okul bilgileri ve gittikleri dersane bilgileri ile tanışıklık durumu anlaşılabilir. Bahsedilen bu olasılıklar nüfus yoğunluğu bakımında küçük yerleşim yerlerinde özellikle dikkate alınması gereken bir durumdur.

6. DEĞERLENDİRME

Bu çalışma, Türkiye’de geniş kitlelere yapılan merkezi sınavlardaki kopya çekme ihtimalini en aza indirmeyi amaçlanmıştır. Birbirini tanıyan kimselerin aynı sınav salonunda sınava girmelerini engelleyen Genetik Algoritma tabanlı özgün bir eniyileme modeli önerilmiştir. Yapılan pratik uygulamalarda kısa bir hesaplama zamanı içerisinde geniş kitlelere uygulanabilen sınav senaryolarında başarılı sonuçlar elde edilmiştir. Hem teorik hem de pratik bir yapıda olan bu çalışma, bir yazılım uygulamasından çok önerilen bir erken önlem alma modelidir.

KAYNAKLAR

- [1] Anadolu Ajansı, *ÖSS’de Skandal: 5 Kardeş Arka Arkaya*, Milliyet Gazetesi, 19 06 2005. [<http://www.milliyet.com.tr/2005/06/19/son/sontur18.html>], Erişim Tarihi: 21.12.2015].
- [2] Shin-ike K. ve Iima H., A Method For Determining Classroom Seating Arrangements By Using A Genetic Algorithm, *SICE Annual Conference (SICE)*, Tokyo, 2011, s. 161-166.
- [3] Krauss B., Lee J., Newman D., Optimizing the Assignment of Students to Classes in an Elementary School, *INFORMS Transactions on Education*, Cilt 14, 2013, s. 39-44.
- [4] Ross P., Hart E. ve Corne D., Some Observations About Ga-Based Exam Timetabling, *Practice and Theory of Automated Timetabling II*, Toronto, Springer Berlin Heidelberg, 2006, s. 115-129.
- [5] Qu E., Burke E., Mccollum B., Merlot L. ve Lee S., A Survey of Search Methodologies and Automated System Development for Examination Timetabling, *J. of Scheduling*, Kluwer Academic Publishers, Cilt 12, No. 1, Şubat 2009, s. 55-89.
- [6] Baeck T. ve Fogel D.B., *Evolutionary Computation 1: Basic Algorithms and Operators*, New York: CRC Press, ISBN: 978-0750306645, 2000.
- [7] Sivanandam S.N. ve Deepa S.N., *Introduction to Genetic Algorithms.*, New York: Springer Science & Business Media, ISBN: 978-3-540-73189-4, 2008.
- [8] Elmas Ç., *Yapay Zeka Uygulamaları*, s. 388-298, İstanbul: Seçkin Yayınevi, 2010.
- [9] Haupt R.L. ve Haupt S.E., *Practicle Genetic Algorithms*, New York: Winley, ISBN: 978-0-471-45565-3, 2004.
- [10] Mitchell M., *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*, New York: The MIT Press, ISBN: 9780262133166, 1998.
- [11] Nebiyev V., *Yapay Zeka Problemler - Yöntemler - Algoritmalar*, İstanbul: Seçkin Yayıncılık, 2013, s.100-110.
- [12] Rylander R., Computational Complexity and Genetic Algorithms, Doktora Tezi, University of Idaho, ABD, 2001, s. 1-110.
- [13] Dasgupta D. ve Michalewicz Z., *Evolutionary Algorithms in Engineering Aslications*, New York: Springer, ISBN 978-3-662-03423-1, 1997.

ÖZGEÇMİŞ/CV

Faruk BULUT; Dr. (Ph. D.)

Lisans derecesini 1998’te Marmara Üniversitesi Bilgisayar Eğitimi Bölümü’nden, Yüksek Lisans derecesini 2010’de İstanbul Fatih Üniversitesi Bilgisayar Mühendisliği Bölümü’nden, Doktora derecesini ise 2015 yılında

İstanbul Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü'nden aldı. Şu anda İzmir Kâtip Çelebi Üniversitesinde öğretim üyesi olarak çalışmaktadır. Temel çalışma alanları Makine Öğrenmesi, Meta Öğrenme ve Kolektif Metotlar üzerinedir.

He has got his bachelors' degree in the Computer Education Department at Marmara University, Istanbul/Turkey in 1998, his master degree in the Computer Engineering Department at Fatih University, Istanbul/Turkey in 2010, PhD degree in the Computer Engineering Department at Yıldız Technical University, Istanbul/Turkey in 2015. He is an academician in the İzmir Kâtip Çelebi University. His major areas of interests are: Machine Learning, Meta Learning and Ensemble Methods.

Şeyda SUBAŞI; Öğrenci (Student)

Şu an Ankara Orta Doğu Teknik Üniversitesi'nde Bilgisayar Mühendisliği Bölümünde lisans öğrencisidir. TÜBİTAK Ulusal Bilgisayar Proje ve Bilim Olimpiyatlarına birçok kere katılmış ve değişik başarılar elde etmiştir. İlgi alanları Veri Yapıları, Algoritmalar ve Çizge Teorileri üzerinedir.

Now she is an undergraduate student in the Computer Engineering Department at Middle East Technical University in Ankara. She has participated to the TUBİTAK National Project and Science Olympiads in Informatics (Computer) several times and won some medals and awards. Her major areas of interests are on Data Structures, Algorithms and Graph Theories.