



SIVAS CUMHURİYET ÜNİVERSİTESİ
Sosyal Bilimler Enstitüsü
Yönetim Bilişim Sistemleri Ana Bilim Dalı

**SEZGİSEL ALGORİTMALAR YARDIMIYLA DERS PROGRAMI
OPTİMİZASYONU**

Yüksek Lisans Tezi

Ahmet YURTSAL

Sivas
Mayıs 2019

SİVAS CUMHURİYET ÜNİVERSİTESİ
Sosyal Bilimler Enstitüsü
Yönetim Bilişim Sistemleri Ana Bilim Dalı

**SEZGİSEL ALGORİTMALAR YARDIMIYLA DERS PROGRAMI
OPTİMİZASYONU**

Yüksek Lisans Tezi

Ahmet YURTSAL

Tez Danışmanı
Doç. Dr. Oğuz KAYNAR

Sivas
Mayıs 2019

KABUL VE ONAY

Üniversite : Sivas Cumhuriyet Üniversitesi
Enstitü : Sosyal Bilimler Enstitüsü
Ana Bilim Dalı : Yönetim Bilişim Sistemleri Ana Bilim Dalı
Tezin Başlığı : Sezgisel Algoritmalar Yardımıyla Ders Programı Optimizasyonu
Savunma Tarihi : 24.05.2019
Danışmanı : Doç. Dr. Oğuz KAYNAR

Unvanı - Adı Soyadı

İmza

Jüri Başkanı : Doç. Dr. Oğuz KAYNAR



Üye : Doç. Dr. Metin ZONTUL



Üye : Doç. Dr. Mehmet Ali ALAN



Oy Birliği

Oy Çokluğu

Ahmet YURTSAL tarafından hazırlanan “Sezgisel Algoritmalar Yardımıyla Ders Programı Optimizasyonu” başlıklı tez, kabul edilmiştir./....../.....

Prof. Dr. Ahmet ŞENGÖNÜL

Enstitü Müdürü

ETİK İLKELERE UYGUNLUK BEYANI

Sivas Cumhuriyet Üniversitesi Sosyal Bilimler Enstitüsü bünyesinde hazırladığım bu Yüksek Lisans tezinin bizzat tarafımdan ve kendi sözcüklerimle yazılmış orijinal bir çalışma olduğunu ve bu tezde;

1. Çeşitli yazarların çalışmalarından faydalandığımda bu çalışmaların ilgili bölümlerini doğru ve net biçimde göstererek yazarlara açık biçimde atıfta bulunduğumu;
2. Yazdığım metinlerin tamamı ya da sadece bir kısmı, daha önce herhangi bir yerde yayımlanmışsa bunu da açıkça ifade ederek gösterdiğimi;
3. Başkalarına ait alıntılanan tüm verileri (tablo, grafik, şekil vb. de dahil olmak üzere) atıflarla belirttiğimi;
4. Başka yazarların kendi kelimeleriyle alıntıladığım metinlerini, tırnak içerisinde veya farklı dizerek verdiğim yine başka yazarlara ait olup fakat kendi sözcüklerimle ifade ettiğim hususları da istisnasız olarak kaynak göstererek belirttiğimi,

beyan ve bu etik ilkeleri ihlal etmiş olmam halinde bütün sonuçlarına katlanacağımı kabul ederim.

21.05/2019

Ahmet YURTSAL



TEŐEKKÜR

Tez alıőmam boyunca bilgi ve tecrübelerini paylaşarak bana yol gösteren deęerli danıőman hocam sayın Do. Dr. Oęuz KAYNAR'a, alıőma boyunca desteklerini esirgemeyen Yasin GÖRMEZ'e ve Ayőegöl ALBAYRAK'a ve her zaman bana moral ve destek veren canım aileme teőekkürlerimi sunarım.



İÇİNDEKİLER

İÇİNDEKİLER	i
KISALTMALAR	iii
TABLolar LİSTESİ	v
ŞEKİLLER LİSTESİ	vii
ÖZET	ix
ABSTRACT	xi
I. BÖLÜM: GİRİŞ	1
1.1. Literatür Araştırması	4
II. BÖLÜM:DERS ÇİZELGELEME PROBLEMİ	13
2.1. Kısıtlar	14
2.1.1. Zorunlu kısıtlar	14
2.1.2. Esnek kısıtlar	15
III. BÖLÜM:SEZGİSEL ALGORİTMALAR	17
3.1. Genetik Algoritma (Genetic Algorithm)	17
3.1.1. Genetik Algoritma Kavramları	19
3.1.2 Genetik Algoritma Çalışma Prensipleri.....	23
3.1.3 Genetik Algoritma Operatörleri.....	24
3.2 Parçacık Sürü Optimizasyonu (Particle Swarm Optimization).....	33
3.3 Yapay Arı Kolonisi Algoritması (Artificial Bee Colony Algorithm)	37
IV. BÖLÜM: UYGULAMA	43
4.1. Problemin kodlaması.....	43
4.2. Uygunluk değerinin hesaplanması	46
4.3. Başlangıç Popülasyonun Oluşturulması.....	49
4.4. Seçilim Operatörünün Uygulanması	50
4.5. Çaprazlama Operatörünün Uygulanması	50
4.7. Tamir Fonksiyonunun Uygulanması	50
4.8. Mutasyon Operatörünün Uygulanması	51
V.BÖLÜM: SONUÇ VE ÖNERİLER	55
KAYNAKLAR	67
EKLER	75

Ek 1. Yönetim Bilişim Sistemleri Bölümü 1. Öğretim Ders Programı	75
Ek 2. Yönetim Bilişim Sistemleri Bölümü 2. Öğretim Ders Programı	77
ÖZGEÇMİŞ.....	79



KISALTMALAR

GA	: Genetik Algoritma
DÇP	: Ders Çizelgeleme Problemi
TA	: Tabu Araştırma
ACO	: Karınca Kolonisi Optimizasyonu
PSO	: Parçacık Sürü Optimizasyonu
ABC	: Yapay Arı Koloni
YZ	: Yapay Zeka
EA	: Evrimsel Algoritmalar
PMX	: Kısmen Uyumlu Çaprazlama
OX	: Sıralı Çaprazlama
PBX	: Pozisyona Dayalı Çaprazlama
CX	: Çevrim Çaprazlama



TABLolar LİSTESİ

Tablo 1. Rulet Tekerleđi Yöntemi.....	26
Tablo 2. Sıralama Yöntemi	27
Tablo 3. Zorunlu Kısıtlar	47
Tablo 4. Esnek Kısıtlar	47
Tablo 5. GA Yöntemine Ait Deney Tablosu.....	59
Tablo 6. PSO Yöntemine Ait Deney Tablosu	61
Tablo 7. ABC Yöntemine Ait Deney Tablosu	63
Tablo 8. GA, PSO ve ABC Yöntemlerini Kıyaslama Tablosu	64





ŞEKİLLER LİSTESİ

Şekil 1. İkili (Binary) Kodlama	21
Şekil 2. Permütasyon (Sıralı) Kodlama	22
Şekil 3. Değer Kodlama	22
Şekil 4. Ağaç Kodlama	22
Şekil 5. GA'nın Akış Diyagramı	24
Şekil 6. Rulet Tekerleği Pasta Grafiği	26
Şekil 7. Turnuva Seçilim Yöntemi	28
Şekil 8. Tek Noktalı Çaprazlama	29
Şekil 9. Çift Noktalı Çaprazlama	29
Şekil 10. Kısmen Uyumlu Çaprazlama.....	30
Şekil 11. Sıralı Çaprazlama.....	31
Şekil 12. Pozisyona Dayalı Çaprazlama	31
Şekil 13. Çevrim Çaprazlama	32
Şekil 14. Mutasyon Operatörü	33
Şekil 15. Parçacık Sürü Optimizasyonu Akış Diyagramı.....	37
Şekil 16. ABC Algoritması Akış Diyagramı	41
Şekil 17. Kodlama Yapısı	45
Şekil 18. Mutasyon Operatörünün Uygulanması.....	51
Şekil 19. GA yönteminde farklı popülasyon sayılarına göre uygunluk değeri sonuçları.....	56
Şekil 20. GA yönteminin çaprazlama oranına göre uygunluk değeri dağılımı.....	57
Şekil 21. GA yönteminin mutasyon oranına göre uygunluk değeri dağılımı	57
Şekil 22. Rulet Tekerleği ve Turnuva Yöntemlerinin Karşılaştırma Sonuçları.....	58
Şekil 23. PSO yönteminin popülasyon sayısına göre uygunluk değeri dağılımı.....	60
Şekil 24. ABC yönteminde farklı popülasyon sayılarına göre uygunluk değeri sonuçları	62
Şekil 25. ABC yönteminde farklı limit sayılarına göre uygunluk değeri sonuçları ..	62



ÖZET

Günümüzde çoğu eğitim kurumunda hazırlanan ders programı her dönem için yeniden yapılmaktadır. Bu işlemin her dönem tekrardan yapılması ve çoğu kurumda elle hazırlanıyor olması bu olayı zahmetli ve zaman alıcı hale getirmektedir. Bu durum hata yapılma ihtimalini ve ders programının bazı yönlerden yetersiz olma ihtimalini de artırmaktadır. Ayrıca her kurumun farklı gereklilik ve kurallara sahip olması, ders programının hazırlanması noktasında mevcut tek bir çözümün elde edilmesini zorlaştırmaktadır. Bu sebeplerden dolayı Ders Çizelgeleme Problemi üzerine farklı yöntemler kullanılarak çok sayıda çalışma yapılmıştır. Bu tez çalışmasında fakültemiz bölümleri için uygun bir haftalık ders programı çizelgesi oluşturulmaya çalışılmıştır.

Çalışmamızda problemin çözümü noktasında evrimsel hesaplama teknikleri olarak kabul edilen Genetik Algoritma, Parçacık Sürü Optimizasyonu ve Yapay Arı Kolonisi yöntemleri kullanılmış ve üç yöntem için de aynı veriler kullanılarak, mevcut yöntemlerin problemin çözümü üzerindeki performansları analiz edilmiştir. Literatürde evrimsel hesaplama teknikleri, Ders Çizelgeleme Problemi ve farklı problemlerin çözümünde kullanılmış ve tekniklerin kaliteli çözümler ürettiği görülmüştür.

Çalışmamızda öğretim elemanı, öğrenci ve fakülte personeli memnun edecek şekilde bütün kısıtlar dikkate alınmıştır. Çalışmamızda kullandığımız yöntemlerin parametreleri üzerinde farklı oran ve sayıda denemeler yapılarak algoritmalar test edilmiştir. Yapılan deneyler sonucunda elde edilen ders programları kontrol edilerek fakülte için uygun ders programları elde edilmiştir. Ayrıca kullanılan algoritmalar çalışma zamanı ve çözüme yakınsama açısından değerlendirilerek performansları karşılaştırılmıştır.

Anahtar Kelimeler: Genetik Algoritma, Parçacık Sürü Optimizasyonu, Yapay Arı Kolonisi, Ders Çizelgeleme Problemi



ABSTRACT

Nowadays, the course schedule is prepared in most educational institutions for every period. The fact that this process is carried out again every year and hand-made in most institutions makes this event troublesome and time consuming. This situation increases the likelihood of mistakes and the possibility that the curriculum is inadequate. In addition, each institution has different requirements and rules, making it difficult to obtain a single solution for the preparation of the curriculum. For these reasons, many studies have been done by using different methods on the Course Scheduling Problem. In this thesis, an appropriate weekly course schedule is prepared for the departments of our faculty.

In the present study, Genetic Algorithm, Particle Swarm Optimization and Artificial Bee Colony methods, which are accepted as evolutionary computational techniques, were used for the solution of the problem and the performances of the existing methods on the solution of the problem were analyzed by using the same data for all three methods. used in the solution of problems and techniques have been found to produce quality solutions.

In our study, all constraints were taken into consideration in order to satisfy the instructor, students and faculty staff. The algorithms were tested by using different ratio and number of experiments on the parameters of the methods used in our study. As a result of the experiments carried out, the course schedules were checked and appropriate course programs were obtained for the faculty. In addition, the algorithms used were evaluated in terms of runtime and solution convergence and their performances were compared.

Keywords: Genetic Algorithm, Particle Swarm Optimization, Artificial Bee Colony, Course Scheduling Problem



I. BÖLÜM: GİRİŞ

Zaman çizelgeleme, günümüzde zamanın verimli kullanılması ve işlerin önceden belirlenen kurallara göre ilerlemesi için hizmet ve üretim sektöründe ihtiyaç haline gelmiştir. Zaman çizelgesi okullarda ders programı ve sınav programının hazırlanması, otobüs hareket saatlerinin belirlenmesi, hastanelerde hemşirelerin vardiyalarının ayarlanması, spor alanında müsabakaların zamanlanması, bir üretim tesisinde işlerin çizelgelenmesi gibi birçok alanda yaygın olarak kullanılmaktadır.

Bir optimizasyon problemi olan zaman çizelgeleme problemleri kaynakların etkin dağılımı ile ilgilenen problemlerdir. Programlama sürecinde birçok kısıt dikkate alınmaktadır. Kaynaklar genellikle sınırlıdır ve iki görev aynı anda belirli bir kaynağı içermemelidir. Zaman çizelgeleme problemlerinin çoğu NP-zor problemler oldukları için deterministik bir algoritma kullanılarak makul bir sürede çözülemezler (Branimir 2003). Bu problemlerin çözümünde yapay zeka optimizasyon algoritmaları olarak adlandırılan Genetik Algoritma (GA), Tabu Araştırma (TA), Karınca Kolonisi Optimizasyonu (ACO), Parçacık Sürü Optimizasyonu (PSO), Yapay Arı Kolonisi (Artificial Bee Colony, ABC) gibi sezgisel yöntemler yaygın olarak kullanılmakta ve problemlerin çözümünde başarılı sonuçlar vermektedir.

Evrimsel Algoritmalar (EA), bilgisayar bilimlerinde, Yapay Zeka (YZ) çalışmalarının altında geçen bir konudur. Evrimsel bir algoritma, çözüm popülasyonundaki uygun olmayan çözümleri yok eden, uygun olan çözümlerin ise hayatta kalmasına izin veren ve istenilen çözüm bulununcaya kadar devam eden bir çalışma yapısına sahiptir. Zamanla başarılı çözümler en uygun çözümü bulmak için evrim geçirmektedir. Başka bir deyişle, EA karmaşık problemleri çözmek için biyolojik süreçleri taklit eden bilgisayar uygulamalarıdır. EA biyolojide seçim, çoğalma ve mutasyon gibi kavramları kullanır. EA tek bir en iyi çözümü sürdürmeye çalışan klasik yöntemlerin aksine, bir çözüm popülasyonuna sahiptir. Çözüm popülasyonu, algoritmanın belirli bir çözümde takılıp kalmasını önlediği için, evrimsel algoritmalarla daha iyi bir optimizasyon mümkündür.

Çalışmamızda kullandığımız GA yöntemi, optimizasyon problemlerinde sıklıkla kullanılan EA'nın en popüler türüdür. GA biyolojiden ilham alınarak

geliştirilen bir arama tekniğidir. Evrimsel biyolojiden gelen kavramları problemleri çözmek için bir teknik olarak kullanır. Optimizasyon, makine öğrenmesi, uyarlanabilir sistem davranışı ve karmaşıklığı gibi çok sayıda bilimsel ve mühendislik probleminde başarılı bir şekilde kullanılmıştır. GA, potansiyel çözümlerin bulunduğu popülasyona, çaprazlama ve mutasyon işlemleri uygular ve üretilen yeni nesil çözümlerle tekrardan bir çözüm popülasyonu oluşturur. Çözümlerden biri problemi çözerse, süreç tamamlanır. Aksi takdirde, iyi performans gösteren çözümler kullanılarak yeni çözümler üretilir ve bu çözüm geliştirme süreci algoritma optimal çözüme ulaşana kadar veya belirlenen durdurma kriteri sağlanana kadar devam eder (Ryder 2006: 1).

Çalışmamızda kullandığımız diğer bir yöntem ise PSO algoritmasıdır. Söz konusu bu algoritma temelde kuş ve balık sürülerindeki bireylerin birbirini geliştirmesine dayanan optimizasyon yöntemidir. PSO en son geliştirilen evrimsel yöntemlerden biridir. Bu yöntem 1995 yılında Kennedy ve Eberhart tarafından geliştirilen ve popülasyon temelli bir yöntemdir. PSO, kuş ve balık sürüleri gibi doğada etkileşim ve iletişim halinde olan canlıların davranışlarından ilham alınarak geliştirilmiştir. Popülasyona dayalı ve doğası gereği evrimsel olduğu için, bir PSO algoritmasındaki üyeler, grubun liderini yani en iyi performansa sahip olanı takip etme eğilimindedir. PSO, GA ile birçok ortak noktayı paylaşmaktadır. Her iki algoritma da rastgele çözümlerden oluşan bir popülasyon grubuyla başlar, her ikisi de popülasyonu değerlendirmek için uygunluk fonksiyonuna sahiptir. Her ikisi de popülasyonu günceller ve rastgele tekniklerle optimum çözümleri arar. Her iki yöntem de optimum çözümleri garanti etmezler ama optimuma yakın sonuçlar bulmada başarılı yöntemlerdir. PSO, problemin çözümüne kolayca uygulanabilir ve basit bir hesaplama mantığına sahiptir ve PSO'nun bu rahat kullanımının yanında birçok optimizasyon probleminin çözümünde etkili bir yöntem olduğu kanıtlanmıştır (Uysal 2006).

Çalışmamızda problemin çözümü için kullandığımız son yöntem ABC algoritmasıdır. Bu yöntem PSO yöntemine benzer şekilde sürü zekasını temel alan bir optimizasyon yöntemidir. Derviş Karaboğa tarafından geliştirilen bu yöntem doğada sürüler halinde hareket eden arıların yiyecek arama davranışları

modellenecek geliştirilmiştir (Karaboga 2005). Yöntem arılarının besin bulmak için kovan ve çevresinde gezerek, sürekli birbirlerine bilgi aktarımında bulunmalarından ilham alınarak geliştirilmiştir. ABC yöntemi kullanılarak birçok alanda çalışmalar yapılmış ve problemlerin çözümündeki sonuçlara göre araştırmacılar tarafından başarılı bir optimizasyon yöntemi olduğu gözlemlenmiştir (Karaboğa 2017; Özcan 2016).

Bu tez çalışmasında GA, PSO ve ABC yöntemleri ders programı çizelgelemede optimum ders, sınıf, öğretmen dağılımının bulunması için kullanılmıştır. Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Fakültesinin ders programının oluşturulması bir problem olarak ele alınmıştır. Belirlenen kısıtlar fakültenin şuanda kullandığı mevcut ders programı dikkate alınarak oluşturulmuştur. Amacımız fakültenin öğretim elemanı ve öğrencisini memnun etmek ve öğretim elemanı ve öğrenci verimliliğini artırarak eğitim öğretim verimliliğini artıran ders programları elde etmektir. Ayrıca fakülte için her dönem hazırlanması gereken ders programı, geliştirilen bu uygulama ile çok fazla emek harcamadan, makul sürede elde edilecek ve bu sayede günümüz iş dünyasında çok büyük öneme sahip zamanın etkin kullanımı sağlanacaktır.

Bu tez çalışmasında GA, PSO, ABC ve ders programı çizelgeleme problemi üzerine literatürde yapılan çalışmalar kısaca anlatılmış, sonrasında Ders Çizelgeleme Problemi(DÇP)'nden bahsedilmiş ve DÇP'nin çözümünde kullanılan GA, PSO ve ABC yöntemlerinin özellikleri, terminolojisi ve nasıl çalıştığı üzerinde durulmuştur. Son olarak DÇP'nin çözümü aşamasında GA, PSO, ABC yöntemlerinin nasıl kullanıldığından ve bu çalışmada yapılan deneylerin sonuçları değerlendirilmiştir.

1.1. Literatür Araştırması

Erben ve Keppeler (1995) ders çizelgeme problemi için GA yöntemini kullanmıştır. Probleme özgü bir kromozom gösterimi ve bilgi artırımı genetik operatörler geliştirmiştir. Bu operatörler, uygun olmayan zaman çizelgesini fark ederek oluşturmaktan kaçınmaktadır. Üniversiteden alınan gerçek verilerle yapılan testler sonucunda umut verici sonuçlar elde edilmiştir.

Çolak (2015) tez çalışmasında ders çizelgeme probleminin çözümü için sezgisel algoritma yöntemlerinden olan GA yöntemini kullanmıştır. Çalışmasında Süleyman Demirel Üniversitesi Uzaktan Eğitim Meslek Yüksekokuluna ait verileri kullanmıştır. Yöntemin uygulanma kısmında C# programlama dili ve Sql Server kullanılarak nesne tabanlı bir algoritma geliştirilmiştir. GA yönteminin uygulanmasında kromozom yapısı olarak her bir sütunun bir zaman dilimini temsil ettiği ve her bir satırın bir dersliği temsil ettiği iki boyutlu bir matris kullanmıştır. Yapılan çalışmada farklı popülasyon sayısı, çaprazlama oranı, mutasyon oranı ve iterasyon sayıları ile deneyler yapılmış ve uygun sonuçlar elde edilmiştir.

Daban ve Özdemir (2004) YZ yöntemlerinden olan GA'yı kullanarak öğretim elemanları ve öğrenci ihtiyaçlarını karşılayabilmek için etkin bir program geliştirmiştir. Öğrencinin haftanın günlerinde ve günün saatlerinde olan algısını dikkate alarak öğrencinin performansını artırmayı amaçlamıştır.

Taç (2006) GA'yı ders programı çizelgeme problemi üzerine uygulamıştır. İki noktadan çaprazlama operatörü kullanmıştır. Kromozom yapısında kısıtlara uymayan genlere kötü gen işareti verilmiştir. Bu işareti mutasyona uğrayacak geni belirlemek için kullanmıştır. Mutasyon olasılığını %20 olarak seçip 20 nesil sonunda en iyi bireyin uygunluk değerinde değişme olmaz ise mutasyon oranının da %5 artırma yöntemi kullanmıştır.

Akkan ve Güçlü (2018) ders programı problemini iki kriterli optimizasyon problemi olarak modellemişler ve problemin çözümünde standart GA yaklaşımına ek olarak Tırmanma ve Tavla Benzetimi algoritmalarını kullanan çok yönlü melez bir GA kullanmışlar.

Song vd. (2017) Üniversite binaları için enerji verimliliğine dayalı ders çizelgeleme çalışması yapmışlar. Bu çalışmada sınıfların verimli tahsisi, daha fazla enerji tasarrufu sağlamaya katkıda bulunabilir fikrini ortaya atmışlardır. Bu bağlamda derslere belirli dersliklerin tahsisi için temel kaynak olan ders çizelgesini kullanmışlar.

Bağış (1996) tez çalışmasında fakülte ders programı optimizasyonu üzerine çalışmış. GA yöntemini kullanarak yaptığı çalışmada, GA operatörlerinin problemin çözümünde GA'nın performansını nasıl etkilediğini incelemiştir.

Alkan (2002)' nın tez çalışması üniversite ders çizelgeleme problemi üzerine olmuştur. Problemin çözümünde yöntem olarak GA'yı kullanmış ve yeni bir mutasyon operatörü (sıralamalı mutasyon) ve çaprazlama operatörü (sınıf bazlı ve düzenli) kullanmıştır.

Datta vd. (2006) ders çizelgeleme problemi üzerine bir çalışma yapmıştır. Kromozom temsili olarak, her bir sütunun bir zaman dilimini temsil ettiği ve bir satırın bir dersliği temsil ettiği iki boyutlu bir matris kullanmışlar. Matrisin her hücresinin değeri ise dersleri temsil etmektedir. Seçilim operatörü olarak da turnuva seçilim yöntemini kullanmışlar.

Yiğit (2006) meslek liselerinde kullanılması amacıyla GA yöntemini kullanarak ders çizelgesi hazırlamıştır. Zaman çizelgeleme probleminin çözümü için yapılan uygulamada C++ programlama dilini kullanmıştır. Uygulanan program, giriş verilerini girmek ve çıktıyı görüntülemek için güçlü ve etkileşimli bir grafik kullanıcı arayüzü içermektedir. Deney sonuçlarına göre optimal bir zaman çizelgesi elde etmiştir.

Yu ve Sung (2002) üniversite haftalık ders çizelgesinin çözümü için bir GA önermişler. Kromozom yapısı olarak derslik ve zaman periyotları şeklinde 2 boyutlu matris kullanmışlar. Başlangıç popülasyonunu oluştururken karmaşıklığı azaltmak için dersleri rastgele atamak yerine zorunlu kısıtların ihlalini önlemişler. Çaprazlama yöntemi olarak kısmen uyumlu çaprazlama yöntemini kullanmışlar.

Baysal (2011) tez çalışmasında GA yöntemini kullanarak Pamukkale Üniversitesi için otomatik sınav programı oluşturmuştur. Çalışmada veri tabanı

modeli SQL Server üzerinde oluşturulmuş ve kodlama yapısı Visual Studio üzerinde geliştirmiştir.

Sigr vd. (2003) GA yöntemini kullanarak zaman çizelgesi üzerine bir çalışma yapmışlar. Algoritmayı, Zagreb'deki Elektrik ve Bilgisayar Mühendisliği Fakültesi'nde küçük ve büyük zaman çizelgesi problemleri üzerinde test etmişler. Program arayüzünü C# ile geliştirmişler. GA'nın temelini, C++ ile STL (Standart Şablon Kütüphanesi) desteği ile geliştirmişler. Zaman çizelgesi olarak gün, zaman aralığı ve derslik şeklinde bir yapı oluşturmuşlar. Seçilim yöntemi olarak rulet tekerleği ve çaprazlama yöntemi olarak tekdüze (Uniform) çaprazlamayı kullanmışlar.

Ghaemi vd. (2007) üniversite ders çizelgeleme problemini çözmek için GA yöntemini kullanmışlar. Problemin çözümü için değiştirilmiş GA ve iş birliği yapan (cooperative) GA yaklaşımlarını kullanmışlar. GA'nın probleme uygulanmasında günler ve zaman periyotları şeklinde bir kromozom yapısı kullanılmış. Seçilim yöntemi olarak rulet tekerleği ve turnuva, çaprazlama yöntemi olarak tek noktalı, tekdüze(uniform), değiştirilmiş tekdüze(modified uniform) yöntemleri uygulanmış. Çalışmada kullanılan GA'nın kodu matlab ortamında oluşturulmuştur.

Küçük (2006) tez çalışmasında GA yöntemini kullanarak hemşire çizelgeleme problemini optimize etme üzerine bir çalışma yapmıştır. Çalışmasında elle hazırlanan problemin hazırlanma süresini azaltmaya ve hemşirelerin uygun çalışma saatlerini bulmayı amaçlamıştır. Problemin çözümü için MATLAB programının GA aracını kullanmıştır.

Aydın (2008) tez çalışmasında GA yöntemini kullanarak bir üniversitenin gerçek verileriyle üniversite ders programı çizelgeleme problemi üzerine çalışmıştır. Problemin çözümü için oluşturduğu yapıda kesin kısıtların ihlaline izin vermeyerek kısa süre içerisinde kaliteli bir çözüm elde etmiştir. Popülasyon büyüklüğü, çaprazlama ve mutasyon oranları gibi GA parametrelerinin en iyi değerleri için denemeler yapmıştır.

Sabar vd. (2012) çalışmalarında, çizelgeleme problemlerini çözmek için bal arısı çiftleşme algoritmasını kullanmışlar. Kullanılan algoritmanın performansı iki problem üzerinde test edilmiştir. Sonuçlar, bal arısı çiftleşme algoritmasının

performansının bilimsel literatürdeki diğer yaklaşımların sonuçlarıyla karşılaştırılabilir olduğunu göstermiştir. Bu da bal arısı çiftleşme algoritmasının eğitimsel çizelgeleme problemlerini çözmeye umut veren bir yaklaşım olduğunu göstermektedir.

Çivril (2009) tez çalışmasını hemşire çizelgeleme problemi üzerine yapmıştır. Çalışmasında bazı hastanelerde hemşire çizelgesini hazırlayan hemşirelerle görüşerek ve daha önceden yapılan çalışma çizelgelerini inceleyerek bazı kısıtlar belirlemiştir. Bu çalışmayı yaparken yöntem olarak GA'yı kullanmış ve programı C# programlama dilinde geliştirmiştir.

Yapıcı (2012) tez çalışmasında, üzerinde kolaylıkla işlemlerin ve değişikliklerin yapılabilirdiği otomatik ders programı hazırlayan bir yazılım geliştirmiştir. Bu yazılımı GA yöntemini kullanarak C# programlama dilinde geliştirmiştir. GA yöntemi, tam genetik algoritma ve yarı genetik algoritma şeklinde kullanılarak test edilmiştir. Tam genetik algoritma rastgele başlangıç popülasyonu oluşturularak test edilmiş ve uygun çözümü bulmada yarı genetik algoritmaya kıyasla daha çok zaman aldığı görülmüştür.

Kalaycı (2008) tezinde, öğrencilerin başarısına odaklı bir sınav çizelgeleme modeli oluşturmayı amaçlamıştır. Bu modeli oluştururken sınavların çakışması veya belirlenen zamanda yapılmaması gibi kesinlikle ihlal edilmemesi gereken kısıtların yanında sınavların çizelgeye dağılımı yapılırken sınav zorluk dereceleri de dikkate alınmıştır. Bu çalışmada problemin çözümü için iki GA modeli kullanılmıştır. Bu modelin ilkinde problem için uygun olmayan çözümlerin uygunluk değerlerine yüksek ceza puanları eklenerek bu çözümlerin popülasyondan yok olması amaçlanmıştır. İkinci modelde ise eğer çözüm zorunlu kısıtları sağlamıyorsa tamir fonksiyonu kullanılarak düzeltme işlemi yapılır. Tamir fonksiyonu sonrası uygun olmayan çözüm yok edilir. Farklı parametrelerle yapılan deneyler sonucunda ikinci modelin yavaş ama daha iyi sonuçlar verdiği gözlemlenmiştir.

Aydın vd. (2008) GA yöntemini kullanarak kısıtlamaya dayalı üniversite ders çizelgeleme problemi üzerine bir çalışma yapmışlar. Yaptıkları çalışmada Atılım Üniversitesi Mühendislik Fakültesi'nin özelliklerine göre esnek bir yazılım

geliştirerek ders programını otomatikleştirmişlerdir. Optimum parametrelere ulaşmak için farklı parametrelerle yapılan deneyler sonucunda uygun çözümlere ulaşılmıştır.

Burke vd. (2007) çalışmalarında, zaman çizelgesinde yaygın olarak kullanılan yapıcı buluşsal bir dizi (grafik renklendirme sezgiselliği) üzerine genel bir hiper-sezgisel yaklaşım araştırmışlar. Hiper-sezgisel çerçevede, sınavlarda ve ders çizelgeleme problemlerinde zaman çizelgesi oluşturmak için kullanılan grafik tahminlerinin permutasyonlarını araştırmak için bir tabu arama yaklaşımı kullanmışlar.

Ulak (2010) tezinde, GA yöntemini kullanarak üniversite ders programlama üzerine çalışma yapmıştır. Problemin çözümünde Grafik Boyama(GC) algoritması ve çok hedefli genetik algoritma yöntemi kullanmıştır. Uygulanan üniversiteye özgü olan kısıtların sağlanması amaçlamıştır. Kromozom yapısı ders sayısı uzunluğu kadar olacak şekilde bir dizi tamsayı olarak tasarlamıştır. Başlangıç popülasyonu rastgele üretilmiş ve seçim operatörü olarak klasik rulet tekerleği yöntemi yerine iki farklı turnuva seçim yöntemi kullanmıştır. Üç farklı çaprazlama operatörü kullanmış ve uygulamada kullandığı mutasyon oranını $1/\text{kromozom uzunluğu}$ şeklinde hesaplamıştır.

Hertz (1991) iki aşamada (TATI / TAG) ders çizelgeleme problemlerini ele almıştır, çalışma yöntemi olarak TA algoritmasını kullanmayı önermiş ve bu yaklaşımın, büyük ölçekli ders çizelgeleme ve test çizelgelemesindeki problemlerin çözümü için uygun olduğunu vurgulamıştır.

Alvarez ve diğ. (2002) ise ders programı çizelgeleme probleminin çözümü için temelinde TA algoritması olan, kullanıcıların hedefleri ve parametreleri geniş bir olasılık yelpazesinden seçebildikleri bir paket program geliştirmişlerdir. Elde ettikleri sonuçlara göre paketin ilk testlerinin tatmin edici sonuçlar verdiğini belirtmişlerdir.

Ders programı çizelgeleme probleminin çözümü için Chen ve Shih (2013), 16 öğretmen, 10 ders ve 10 derslik içeren bir durumu analiz etmek için çalışmalarında standart PSO ve yerel arama sezgisellerini kullanmışlardır. Çalışmalarında öğretmenlerin beklentilerinin ve sınıfların yeterliliğinin yanı sıra çeşitli kısıtları karşılamak için en uygun ders programını amaçlamışlardır. Elde ettikleri sonuçlara göre yerel arama eklendiğinde öğretmenlerin ve sınıfların uygunluk değeri, yerel

arama eklenmeden elde edilen uygunluk değerinden daha iyi olduğunu ve Yerel Arama yöntemi ile birlikte Standart PSO kullanımının en iyi çözümü elde ettiğini gözlemlemişlerdir.

Shiau (2011), ders programı çizelgeleme problemi konusunda yaptığı çalışmada, hem öğretim üyelerinin hem de öğrencilerin zaman dilimi tercihlerini dikkate almıştır. Çalışmasında, en küçük problemde 17 öğretim üyesinin tercihi alınarak 8 ders ataması yapılırken, en büyük problemde ise 49 öğretim üyesinin tercihleri dikkate alınarak 28 dersin ataması yapmıştır. Bu boyuttaki bir çizelgeleme problemi için PSO algoritması ile problemin çözümünde başarılı sonuçlar elde etmiştir.

Chu ve diğ. (2006), yaptıkları çalışmada sınav çizelgeleme için PSO algoritması geliştirmişlerdir. Geliştirilen algoritma 9 sınav 50 öğrenci içeren bir duruma uygulanmıştır. Ele aldıkları problemde sınavlar 3 gün içinde gerçekleştirilmiştir ve her gün ikisi sabah ikisi de öğleden sonra olmak üzere 4 zaman dilimini içermektedir. Elde ettikleri deneysel sonuçlara göre, PSO'nun zaman çizelgesi programlamasının ayrık problemini çözmek için etkili bir yöntem olabileceğini ve zaman çizelgeleme problemi için herhangi bir çatışmanın olmadığını belirtmişlerdir.

Tassopoulos ve Beligiannis (2012), Yunan Liselerinde karşılaşılan ders çizelgeleme problemi için PSO'ya dayalı hibrit uyarlanabilir algoritma geliştirmiş ve uygulamışlardır. Önerilen PSO algoritmasının etkinliğini ve üstün performansını gösterebilmek için literatürde yer alan Sütun Oluşturma Yaklaşımı (Column Generation Approach), GA, Evrimsel Algoritma, Kısıt Programla yaklaşımı ile karşılaştırmışlardır. Deneysel sonuçlara göre çalışmalarında önerdikleri PSO algoritmasının diğer dört etkili algoritmaya kıyasla daha iyi sonuçlar elde ettiğini ifade etmişlerdir.

Irene vd. (2009), ise çalışmalarında PSO algoritmasının üniversite ders çizelgeleme probleminin nasıl çözeceği üzerinde odaklanmıştır. PSO algoritmasını üç farklı veri setinde test etmişlerdir (İlk veri seti 183 ders, 21 sınıf ve 45 zaman diliminden oluşmaktadır. İkinci veri seti 74 ders, 22 odadan ve 45 zaman diliminden oluşmaktadır. Son olarak, üçüncü veri setinde 226 ders 16 oda ve 45 zaman aralığı

bulunmaktadır). Parçacık sürü optimizasyon algoritması tabanlı ders çizelgelemesinin performansını, GA- Kısıta Dayalı Akıl Yürütme tekniği ile karşılaştırmışlardır. GA- Kısıta Dayalı Akıl Yürütme tekniğinin üniversite ders çizelgeleme problemini çözmede etkili olmasına rağmen, PSO tabanlı algoritmaya kıyasla daha fazla hesaplama zamanı harcadığını gözlemlemişlerdir. Elde ettikleri sonuçlara göre PSO algoritmasının üniversite ders çizelgeleme problemleri için alternatif bir çözüm olabileceğini belirtmişlerdir.

Uysal (2006) tez çalışmasında iki kriterli permütasyonlu akış tipi üretim çizelgeleme problemi için GA ve PSO yöntemlerini karşılaştırmıştır. Çalışmasında, 50 iş – 20 makine gibi büyük çaplı problemler için, toplam yapım zamanı ve en büyük pozitif gecikme zamanı kriterlerini birlikte dikkate almıştır. Bu amaçla, PSO ve GA geliştirmiş ve standart test problemlerine uygulamıştır. Yalnızca PSO ve GA'nın yalın sürümleri değil aynı zamanda Değişken Komşuluk Arama isimli bir yerel arama yöntemiyle melezlenmiş olan şekilleri de denenmiş ve iki algoritmanın nispi performansları karşılaştırmıştır. Sonuç olarak, PSO, maksimum gecikme kriterinin ağırlığının daha yüksek olduğu durumlarda daha iyi performans gösterdiği, GA'nın toplam yapım zamanı kriterinde daha başarılı olduğu görülmüştür.

Wu vd. (2015) çalışmalarında PSO algoritması kullanılarak hemşire çizelgeleme problemi geliştirmişlerdir. Geliştirilen modelde hemşirelerin çalışma planlarının adaletli ve dengeli bir şekilde yapılması ile hemşirelerin hizmet kalitesinin artırılması amaçlanmıştır. Elde ettikleri deneysel sonuçlara göre PSO algoritmasının optimum çözümler ürettiğini belirtmişlerdir.

Rasip vd. (2014) çalışmalarında hastanenin talep kapsamını ve hemşire tercihlerini göz önüne alarak hemşire görevlendirme dengesini sağlamak için PSO algoritmasını kullanmıştır. Yöntem Malezya'daki devlet hastanelerin gerçek verileri üzerinde test edilmiş ve elde edilen sonuçlar PSO algoritmasının karmaşık çizelgeleme problemleri üzerinde iyi bir performansa sahip olduğunu belirtmişlerdir.

Altamirano vd. (2012) çalışmalarında Fransız devlet hastanesindeki anestezi hemşirelerinin atama problemlerinin çözümü için PSO kullanmışlardır ve yöntemlerini tam sayılı programlama kullanılarak elde edilen önceki sonuçlar ile

karşılaştırmışlardır ve araştırmacıların tasarladıkları PSO algoritmasının performansının daha iyi olduğunu belirtmişlerdir.

Limlawan vd. (2011) çalışmalarında Thai Airways'in gerçek verileri üzerinde hava alanında çalışan ekibin çizelgeleme problemi için PSO algoritmasını kullanmışlardır. Elde edilen sonuçlara göre parçacık sayısının en iyi 10 olduğunu ve algoritmanın problemi yaklaşık 15 saniyede çözdüğünü belirterek, yöntemin etkili olduğunu ifade etmişlerdir. Limlawan vd. (2014) bir diğer çalışmalarında ise havayolu ekip çizelgeleme probleminin çözümü için hibrit PSO ve iyileştirici sezgisel teknikler (improvement heuristic- IH) önermişlerdir. Önerdikleri yöntemi Thai Airways'in gerçek verileri üzerinde test etmişler ve PSO ve IH kombinasyonu ile, algoritmanın çoğu durumda çözümün kalitesini% 20'den daha fazla artırabildiğini ve IH ile PSO kombinasyonunun 9 durumdan 6'sında daha iyi performans gösterdiğini belirtmişlerdir.

Günther ve Nissen (2010) ise çalışmalarında lojistik servis sağlayan bir işletmenin çalışanları için personel çizelgeleme oluşturmak amacıyla PSO yöntemini kullanmışlardır. Elde ettikleri sonuçlara göre Parçacık Sürü Optimizasyon yöntemi, kullanılan diğer yöntemlerden daha iyi performans gösterdiğini belirtmişlerdir.

Küçüksille ve Tokmak (2011) çalışmalarında, Süleyman Demirel Üniversitesi Gelendost Meslek Yüksekokulu dönem derslerini ve ders görevlendirmelerini ve ihtiyaçlarını dikkate alarak C# programlama dilinde, ABC algoritması kullanarak bir program geliştirmişlerdir. Çalışmalarında ilgili kurumun ihtiyacına göre optimum çizelgenin oluşturulması için sert ve esnek kısıtlar belirlemişlerdir. Geliştirdikleri algoritma ile sert kısıtlar üzerinde %100 başarı elde ederken, esnek kısıtlarda %92,5 oranında başarı elde etmişlerdir.

Bolaji vd. (2011) ikinci Uluslararası Zaman Çizelgesi yarışması (ITC-2007) tarafından hazırlanan veri setini kullanarak ABC algoritmasının performansını test etmişlerdir. Ders programı çizelgeleme problemi için gözcü arı fazında komşuluk yapıları kullanılarak geliştirilmiş bir ABC algoritması önermişlerdir. Önerdikleri geliştirilmiş ABC algoritmasının, iyi kalitede sonuçlar verebilmesine rağmen diğer yöntemlerle elde edilen en iyi sonuçlarla karşılaştırılamayacağını belirtmişlerdir.

Alzaqebah ve Abdullah (2011), sınav çizelgeleme problemlerinin zorluk derecesinin NP-Zor düzeyinde olan problemlerden biri olduğunu öne sürmüştür. Bu nedenle çalışmalarında, ABC algoritması kullanmıştır. Çalışmada geliştirilen çözüm yöntemi, Carter vd. (1996) tarafından sunulan, çeşitli eğitim kurumlarından alınan 13 veri seti ve ikinci Uluslararası Zaman Çizelgesi yarışması (ITC-2007) tarafından hazırlanan veri seti üzerinde test edilmiş ve umut verici sonuçlara erişilmiştir. Alzaqebah ve Abdullah (2011) bir diğer çalışmalarında ise farklı seçim yöntemleri (disruptive, tournament ve rank seçim yöntemleri) kullanarak ABC algoritmasının performansını karşılaştırmışlardır. Deneysel sonuçlara göre disruptive seçim yöntemi ile gerçekleştirilen ABC algoritmasının, diğer seçim yöntemlerine kıyasla daha iyi sonuçlar elde etmişlerdir. Ayrıca hibrit bir ABC algoritmasının kullanılması algoritmanın performansını artırabileceğini vurgulamışlardır.

Bolaji vd. (2015) çalışmalarında, 2010 Birinci Uluslararası Hemşire Görevlendirme Yarışması (INRC2010) tarafından sunulan veri seti üzerinde ABC Algoritması kullanmışlardır. Önerdikleri algoritma tarafından üretilen sonuçları aynı veri seti üzerinde çalışılan mevcut bazı tekniklerle karşılaştırmışlardır ve ABC algoritmasının umut verici olduğunu fakat algoritmanın daha da iyileştirilmesi için daha fazla araştırma yapılması gerektiğini vurgulamışlardır.

Awadallah vd. (2015) ise çalışmalarında hemşire çizelgeleme problemi için Tepe Tırmanma algoritması kullanarak hibrit bir ABC algoritması önermişlerdir. Önerilen hibrit yöntemin performansını 2010 Birinci Uluslararası Hemşire Görevlendirme Yarışması (INRC2010) tarafından sunulan veri seti kullanılarak değerlendirilmiştir. Çalışmanın deneysel sonuçları Tepe Tırmanma algoritmasının ABC algoritmasının performansı üzerinde büyük bir etkisi olduğunu belirtmişlerdir.

II. BÖLÜM: DERS ÇİZELGELEME PROBLEMİ

Ders programı üniversitelerin her dönem yapması gereken bir görevdir. Son zamanlarda, derslik ve öğretim görevlisi artış hızı düşükken, üniversitedeki lisans ve lisansüstü eğitimlerin sayısındaki artış nedeniyle ders programı hazırlanırken zamanın ve kaynakların etkin kullanımı önemli hale gelmiştir (Deris 1999: 241).

DÇP'nin çözümü için bir dizi ders ve bir dizi derslik, şube ve öğretim elemanı verileri kullanılır. Ders programı hazırlanırken her ders, bir derslik, şube ve öğretim elemanı talep etmektedir. Amaç, bu dersleri uygun zaman periyotlarına atamaktır. Bunu yaparken hiçbir derslik, şube ve öğretim elemanı aynı anda birden fazla derse atanmamalıdır (Branimir 2003: 519).

Planlanan çok sayıda ders ve zaman çizelgesine uygulanan çok çeşitli kısıtlamalar olası çözümlerin bulunmasını zorlaştırmaktadır. Bu da ders programı çizelgesinin oluşturulmasını son derece kapsamlı bir görev haline getirir (Burke 2002: 266).

DÇP, NP-zor problemleri sınıfındadır. Bunun anlamı, bu tür problemleri çözmek için gereken zaman ve iş miktarının, problem büyüklüğüne göre katlanarak artmasıdır. Bu problemleri daha zor ve zaman alıcı hale getirir. Bu nedenle, optimizasyon teknikleri onları çözmek ve kesin çözümler yerine en uygun veya en uyguna yakın çözümleri üretmek için kullanılır (Aydın 2008: 3).

Her üniversitenin istekleri, ihtiyaçları, derslikleri, öğretim elemanları ve ders özellikleri gibi ders programının hazırlanması için gerekli olan bilgilerin çoğu birbirinden farklıdır. Üniversitelerin bu şekilde farklılıklarının olması her üniversite için farklı ders programının hazırlanmasını gerektirir.

Bu tez çalışmasında Cumhuriyet Üniversitesi İktisadi İdari Bilimler Fakültesine ait veriler kullanılmıştır. Bu verilere göre fakülte de 9 bölüm bulunmaktadır. Bu bölümlerden sadece Bankacılık ve Finans Bölümü'nün 3'ü birinci öğretim ve 3'ü ikinci öğretim olmak üzere bahar yarıyılında toplam 6 şubesi bulunmaktadır. Kalan 8 bölümün her birinin bahar yarıyılında 8 şubesi bulunmaktadır. Fakültenin 36 adet farklı öğrenci kapasitesine sahip normal dersliği,

4 adet de laboratuvarı vardır. Problemin çözümünde kullanılan veri setleri aşağıdaki gibidir.

$$\text{Dersler}(D) = \{ d_1, d_2, d_3, \dots, d_a \}$$

$$\text{Derslikler}(C) = \{ c_1, c_2, c_3, \dots, c_b \}$$

$$\text{Şubeler}(S) = \{ s_1, s_2, s_3, \dots, s_c \}$$

$$\text{Öğretim Elemanı}(O) = \{ o_1, o_2, o_3, \dots, o_d \}$$

$$\text{Zaman Periyotları}(Z) = \{ z_1, z_2, z_3, \dots, z_e \}$$

2.1. Kısıtlar

Ders çizelgeleme probleminde, yerine getirilmesi gereken bazı kısıtlar mevcuttur. Bu kısıtlar zorunlu ve esnek olarak ikiye ayrılır. Ders çizelgeme probleminde zorunlu ve esnek kısıtlar belirlenirken dikkat edilmesi gereken nokta uygun bir ders programının hazırlanması için zorunlu kısıtların kesinlikle sağlanması, esnek kısıtların ise ders programın kalitesini artırmak için mümkün olduğunca sağlanmasıdır. Esnek kısıtların yeteri kadar sağlanmaması durumunda öğrenci ve öğretim elemanını memnun etmeyen bir ders programı ortaya çıkacaktır. Bu iki kısıt tipini özetlemek gerekirse, zorunlu kısıtlar modelin mutlaka karşılaması gereken, tavize izin verilmeyen kısıtlardır, diğer taraftan esnek kısıtlar ise öğretim elemanlarının istekleri doğrultusunda ve öğrenci memnuniyetini artırmak amacı ile belirlenen kısıtlardır. Bu çalışmada belirlenen kısıtlar uygulamanın yapıldığı fakültenin özelliklerine göre hazırlanmıştır.

2.1.1. Zorunlu kısıtlar

Kullanılabilir bir ders programının elde edilebilmesi için bu kısıtların hepsinin karşılanması gerekmektedir. Aşağıda fakültemiz için belirlenen zorunlu kısıtlar sıralanmıştır.

- 1) Öğretim elemanı herhangi bir zaman diliminde birden fazla derste bulunamaz.
- 2) Öğrenci grupları(şubeler) aynı anda birden fazla derste bulunamaz.
- 3) Bir dersliğe aynı zaman diliminde birden fazla ders atanamaz.

4) Laboratuvar gibi özel derslik gerektiren derslere uygun derslikler atanmalıdır.

5) Haftalık ders saati dört olan bir ders iki saat şeklinde bölünerek, farklı günlere atanmalıdır.

6) Dersler, yapılması istenilen dersliklere ve öğrenci sayılarına göre dersliklere uygun bir şekilde atanmalıdır.

7) Bir şubenin 1. Öğretim(gündüz) ve 2. Öğretim(gece) dersi aynı güne atanmalıdır.

8) Fakültedeki tüm bölümlerin 1. Sınıf ve 2. Sınıflarında uzaktan eğitim dersleri bulunmaktadır. Bu dersler için önceden zaman periyotları belirlenmiştir. 1.Sınıflar(şubeler) ve 2.Sınıflar(şubeler) için uzaktan eğitim derslerinin yapılacağı saatlere başka ders atanmamalıdır.

9) Bazı öğretim elemanları gece derslerinin 17.00 'dan önce olmasını istememektedir. Bu öğretim elemanlarına gece için beş saatten fazla ders atanmamalıdır.

2.1.2. Esnek kısıtlar

Bu kısıtlar, mümkün olduğu kadar yerine getirilmesi gereken zorunlu kısıtlara göre daha az önemli kısıtlardır, yeteri kadar sağlanmasa bile zorunlu kısıtlar sağlandığı takdirde kullanılabilir bir ders programı elde edilecektir. Ancak elde edilen bu ders programı fakülte için tatmin edici bir çözüm olmayacaktır. Dönem boyunca kullanılacak olan ders programının esnek kısıtlarının sağlanması, hem programı güzel ve kullanışlı hale getirir hem de dönemin öğrenciler ve öğretim elemanları için daha keyifli ve verimli geçmesini sağlar. Aşağıda fakültemiz için belirlenen esnek kısıtlar sıralanmıştır.

1) Herhangi bir öğretim elemanın istemediği bir zaman dilimine ders atanmamalıdır.

2) Bir dersin ilk ve son saati aynı derslikte ve aynı gün olmalıdır.

3) Şubelerin gün içerisindeki dersleri arasında boşluk olmamalıdır.

4) Şubeler için haftalık ders saati günlere dengeli bir şekilde atanmalıdır.

5) İsteyen öğretim elemanları için günlük ders saati dengeli bir şekilde ayarlanmalıdır.

6) İsteyen öğretim elemanlarının dersleri gün olarak art arda olmalıdır.



III. BÖLÜM: SEZGİSEL ALGORİTMALAR

Sezgisel algoritmalar, herhangi bir amacı gerçekleştirmek veya hedefe varmak için çeşitli alternatif hareketlerden etkili olanlara karar vermek amacı ile tanımlanan kriterler veya bilgisayar yöntemleridir. Bu algoritmalar, çözüm uzayında optimal çözüme yakınsaması ispat edilemeyen algoritmalar olarak da adlandırılır. Bu tür algoritmalar yakınsama özelliğine sahiptir, ama kesin çözümü garanti etmezler ve sadece kesin çözüm yakınındaki bir çözümü garanti edebilirler. Sezgisel algoritmalara gerek duyulmasının sebeplerinden bazıları aşağıdaki gibi sıralanabilir (Karaboğa 2017: 15).

- Optimizasyon problemi, kesin çözümü bulma işleminin tanımlanamadığı bir yapıya sahip olabilir.
- Anlaşılabilirlik açısından sezgisel algoritmalar karar verici için daha basit olabilir.
- Sezgisel algoritmalar, öğrenme amaçlı ve kesin çözümü bulma işleminin bir parçası olarak kullanılabilir.
- Matematik formülleriyle yapılan tanımlamalarda genellikle gerçek dünya problemlerinin en zor tarafları (hangi amaçlar ve hangi sınırlamalar kullanılmalı, hangi alternatifler test edilmeli, problem verisi nasıl toplanmalı) ihmal edilir. Model parametrelerini belirleme aşamasında kullanılan verinin hatalı olması, sezgisel yaklaşımının üretebileceği alt optimal çözümden daha büyük hatalara sebep olabilir.

Sezgisel algoritmalar sınıfına giren GA, PSO ve ABC yöntemleri bu çalışmada kullanılmış, üç yöntemin genel yapısı ve özellikleri bu bölümde anlatılmıştır.

3.1. Genetik Algoritma (Genetic Algorithm)

GA ilk olarak Michigan Üniversitesi'nde psikoloji ve bilgisayar bilimi uzmanı olan John Holland tarafından 1970'li yıllarda bulunmuş ve daha sonra çalışma arkadaşları ve öğrencileri ile birlikte geliştirilmiştir. Holland 1975 yılında yaptığı çalışmaları "Adaptation in Natural and Artificial Systems" adlı kitabında bir araya getirmiştir. Makine öğrenmesi üzerine çalışmalar yapan Holland, Darwin'in

evrim teorisinden etkilenecek bunu bilgisayar ortamında kullanmayı düşünmüştür. Tek bir mekanik yapının öğrenme yeteneğini geliştirmek yerine, bu tür yapılardan oluşan topluluğun çiftleşme, çoğalma ve değişim vb. gibi genetik süreçlerden geçerek başarılı yeni bireyler (öğrenebilen) oluşturulabildiğini göstermiştir (Şahin 2009: 29). Holland'ın bu çalışmalarından sonra onun doktora öğrencisi olan inşaat mühendisi David E. Goldberg, 1989 yılında alanında klasik olarak kabul edilen “Genetic Algorithms in Search, Optimization & Machine Learning” adlı kitabını yayınlamıştır. Goldberg kitabında yaptığı çalışmaları ile genetik algoritmanın daha iyi anlaşılmasını sağlamış ve birçok alanda uygulanabileceğini göstermiştir. Günümüze kadar bu yöntemle ilgili çeşitli çalışmalar yapılmış ve birçok alana uygulanmıştır.

Goldberg (1989) GA'yı, doğal seleksiyon ve doğal genetik mekaniğine dayanan arama algoritması olarak tanımlamıştır. Bir evrimsel hesaplama yöntemi olan GA, genetik ve doğal seleksiyonu temel alan sezgisel bir arama algoritmasıdır. Bu yöntem evrim teorisini temel alarak bir dizi çözüm ile başlayarak yeni nesil çözümler üreterek arama yapar. Canlıların doğada daha başarılı canlılara dönüşme biçiminden esinlenir (Mathew 2012: 1). GA bu yönden doğadaki canlıların geçirdiği süreci örnek alarak Darwin'in en iyi olan hayatta kalır prensibini temel almaktadır. GA problemlere raslantısal olarak çözüm aradıkları için sezgisel bir yaklaşıma sahiptir. Sezgisel yöntemler, klasik yöntemlerin kesin çözüm bulmakta zorlandığı bazı problemler için kısa süre içerisinde kabul edilebilir yaklaşık bir çözüm üreten tekniklerdir. GA, zor arama alanlarında klasik yöntemlerin gösterdiği eksikliklerin çoğundan kaçınan global bir optimizasyon tekniğidir (Grefenstette 1986: 122). Goldberg' e göre GA klasik optimizasyon ve araştırma yöntemlerinden dört farklı şekilde ayrılır (1989: 7).

- GA parametre setinin kodlanmasıyla çalışır, parametrelerin kendisiyle çalışmaz.
- GA, olasılığa dayanan kurallar kullanmayı tercih eder, deterministik kurallar kullanmaz.
- GA popülasyonun her noktasında arama yapar, tek bir noktadan arama yapmaz.

- GA amaç fonksiyonu bilgisini kullanır, türev araçları veya diğer yardımcı bilgiler kullanmaz.

Geleneksel optimizasyon yaklaşımları bazı problem çözümlerinde başarısız olurken, GA'nın bu farklı yaklaşımı sayesinde daha uygun çözümler bulunmaktadır. Elbette GA her sorunu çözenin en iyi yolu değildir. Örneğin, klasik metotlar birkaç değişkenli analitik fonksiyonların çözümünü bulmada daha uyumludur. Bu gibi durumlarda hesaplamaya dayalı yöntemler GA'dan daha iyi performans gösterir. Buna ek olarak aşırı zor olmayan problemler için, diğer yöntemler çözümü GA'dan daha hızlı bulabilir (Haupt 2004: 23). Bundan dolayı GA'nın aşağıdaki durumlarda kullanılması daha avantajlı olacaktır.

- Arama uzayının büyük olduğu durumlarda,
- Problemin belirli bir matematiksel modelle ifade edilemediği durumlarda,
- Problemin karışık(çözümün zor) olduğu durumlarda,
- Klasik yöntemlerin istenilen sonucu vermediği durumlarda, GA'yı kullanmak daha etkili olur.

Bir genetik algoritmanın yapısı şaşırtıcı derecede basittir, çözümleri kopyalamaktan ve kısmi çözümleri değiştirmekten daha karmaşık hiçbir şey içermemektedir. İşlem basitliği ve etki gücü GA yaklaşımının başlıca çekiciliğidir (Goldberg 1989: 10). GA'nın bu avantajlı yapısından dolayı yapay zeka uygulamaları, mühendislik tasarımı, yol bulma problemleri, çizelgeleme problemleri, robot uygulamaları, görüntü ve ses tanıma problemleri, en iyileme problemleri, sosyal ve ekonomik planlama, ağ tasarım problemleri gibi bir çok alanda tercih edilmiş ve başarılı sonuçlar elde edilmiştir (Emel, Taşkın 2002; Yiğit 2011).

3.1.1. Genetik Algoritma Kavramları

Gen:

Problemin olası bir çözümüne sahip olan kromozom yapısını oluşturan özelliklerin her birine gen denir. Yapısında her biri kendi başına birer bilgi taşıyan en küçük birimlerdir. Bu yapılar bir araya gelerek kromozomları oluşturur (Elmas 2007:

388). Gen yapısı uygulanan probleme göre farklılık gösterebilir. Genin yapısı GA'nın performansı açısından algoritma içerisinde dikkat edilmesi gereken bir kısımdır. Her probleme uygun olabilecek bir gen yapısı söylemek zordur. Problemin tipine göre ikili tabanda (binary), onluk tabanda, gray, tam sayı, gerçel sayı gibi farklı gösterimlerle ifade edilebilir.

Kromozom(Birey):

Birden fazla genin bir araya gelerek oluşturduğu, problemin çözümüne ait bilginin tamamını içeren yapılara kromozom denir (Küçük 2016: 35). Açık bir şekilde ifade etmek gerekirse; problemin çözümüne ait en küçük özelliği taşıyan genler birleşerek kromozomu oluşturur, kromozomlar da birleşerek popülasyonu oluşturur. Bu kromozomların her biri muhtemel bir çözümü temsil eder. Bu yapılar çözüm niteliklerini doğrudan açık tipte değil, kodlanmış tipte depolar (Streichert 2002: 6). Her biri bir çözümü temsil ettiğinden GA'nın işleyişi için oldukça önemli bir birimdir. Bu yüzden bilgisayar ortamında iyi ifade edilmesi gerekir (Taç 2006: 10).

Popülasyon:

Popülasyon, her biri problemin olası bir çözümü olan bireylerin bir araya gelerek oluşturduğu topluluğa denir. Popülasyonun birey (kromozom) sayısı problemin özelliğine göre uygulamayı tasarlayan tarafından belirlenir (Elmas 2007: 388). Popülasyonun birey sayısının belirlenmesi uygun çözümün bulunması açısından oldukça önemlidir. GA'nın birey sayısının az olması uygun çözümün bulunamaması, çok olması ise çözüm süresinin uzaması gibi problemlere neden olabilir. Belirli bir probleme yönelik bir dizi çözümü temsil eden bireyler, GA çalıştırılmadan önce hazırlanır. En baştaki bu bireylere GA'nın başlangıç popülasyonu denir. Başlangıç popülasyonu rastgele hazırlanır ve genellikle problem için örnek bir çözümü temsil eden tek bir kromozomdan elde edilir (Abidin 2013: 12). Fakat ilk popülasyonun rastgele oluşturulması yerine tanımlanan bazı kısıtlara dikkat ederek hazırlanması sayesinde daha iyi adayların oluştuğu ve algoritmaya zaman kazandırdığı görülmüştür (Taç 2006: 11).

Uygunluk Fonksiyonu:

Uygunluk fonksiyonu popülasyon içerisindeki her çözümü, problemin kısıtlarına göre değerlendirerek her çözüm için bir uygunluk değeri belirler. Bir kromozomun uygunluk değeri, o kromozomun elindeki problemi ne kadar iyi çözdüğüne bağlıdır (Mitchell 1998: 7). Bu işlev, probleme özeldir ve standart bir hesaplama formülüne sahip değildir (Abdelhalim, Ghada 2016: 1400). Kromozomun uygunluk değeri yüksek olursa çoğalmak için seçilme şansında yüksek olur. Uygunluk değeri yüksek olan bireyler ne kadar çok bir araya gelirse, oluşan yeni bireyler o kadar kaliteli bir yapıya sahip olurlar (Kuruca 2009:6).

Kodlama:

Genetik algoritmanın önemli bir özelliği, problemi tanımlayan değişkenlerin kodlanmasıdır. Kodlamanın yapılması genetik algoritmanın uygulanmasının ilk adımını oluşturmaktadır (Taşkın, Emel 2009: 33). Bir genetik algoritmanın başarısı, sorunun uygun gösterimine(kodlama) bağlıdır. Her kodlama tipi her problem için uygun olmayabilir. Problemin türüne göre ikili, sıralı, değer ve ağaç kodlama türlerinden biri kullanılabilir. Bu dört kodlama tipinin kullanım şekli aşağıda gösterilmiştir.

İkili kodlama, en yaygın kodlama yöntemidir. Bu kodlamada her kromozom ikili (binary) {0,1} sayılarından oluşmaktadır. Bu sayıların her biri çözüme ait bazı özellikleri göstermektedir (Çakar 2009: 50). Bu kodlama için örnek kromozomlar aşağıdaki gibidir.

	101110111110101
Kromozom2 :	100010100010110

Şekil 1. İkili (Binary) Kodlama

Permütasyon (sıralı) kodlamada bir araya gelen rakamların sıralanmasıyla oluşmaktadır (Çakar 2009: 51). Her kromozom bir numaralar dizisidir. Permütasyon (sıralı) kodlama çizelgeleme, şebeke tasarımları, araç rotalama, gezgin satıcı gibi

sıralama problemlerine uygun bir kodlama tipidir (Dilaver 2015: 17). Bu kodlamanın gösterim biçimi aşağıdaki gibidir.

	1 4 5 7 8 3 2 6 9
Kromozom B :	7 3 9 2 1 6 8 4 5

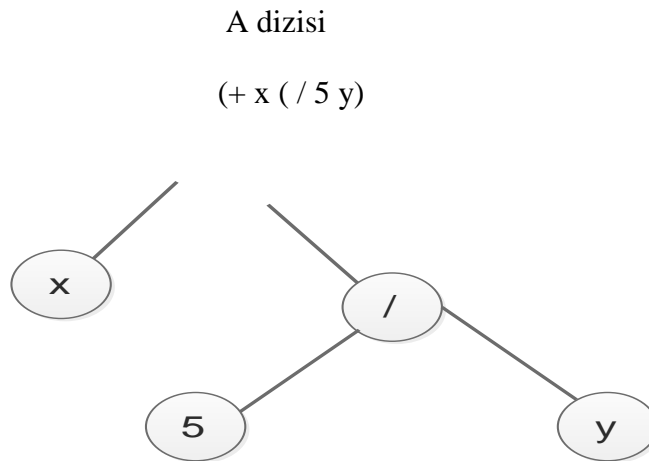
Şekil 2. Permütasyon (Sıralı) Kodlama

Değer kodlaması, gerçek sayıların veya karmaşık değerlerin kullanıldığı kodlama yöntemidir. İkili kodlamanın bu tipteki problemlerde kullanılması oldukça zordur. Değer kodlamasında, her kromozom art arda sıralanan değerlerden oluşmaktadır. Değer kodlaması bazı problem tipleri için oldukça kullanışlıdır. Fakat değer kodlama yönteminin kullanılması durumunda problemdeki mutasyon ve çaprazlama operatörleri de probleme özgü olması gerekmektedir (Taşkın, Emel 2009: 35). Bu kodlamanın gösterim biçimi aşağıdaki gibidir.

	4.4253 5.5234 1.8221 0.1011 2.9137
Kromozom B	UBNIOEIJBRJTIOKSLERPJFZ
Kromozom C	(sağ), (ileri), (sol), (ileri), (sol)

Şekil 3. Değer Kodlama

Ağaç kodlama yöntemi genellikle genetik programlamada kullanılır. Yöntemin her dizisi nesnelere oluşan bir ağaçla ifade edilir. Programların evrime tabi tutulduğu durumlarda oldukça uygun bir kodlama yöntemidir (Taşkın, Emel 2009: 35).



Şekil 4. Ağaç Kodlama

3.1.2 Genetik Algoritma Çalışma Prensibi

Genetik algoritmalar genel olarak, bir problemi çözerken o probleme ait aday çözümleri çoğaltarak, değiştirerek ve değerlendirerek o problemi çözer (Fang 1994: 30). GA çalıştırılmadan önce her biri probleme ait bir çözümü temsil eden bir dizi kromozom hazırlanır. En baştaki bu kromozomlara GA'nın başlangıç popülasyonu denir. Başlangıç popülasyonu genellikle rastgele hazırlanır ve problem için örnek çözümleri temsil eden bireylerden oluşur (Abidin 2013: 12). Başlangıç popülasyonu hazırlandıktan sonra bu popülasyondaki bireylerin uygunluk değerleri hesaplanır. Daha sonra uygunluk değerleri hesaplanan bu bireylere GA operatörleri olan çoğalma, çaprazlama ve mutasyon işlemleri uygulanır.

Mitchell'e göre basit GA adımları aşağıdaki gibidir (1998: 8-9).

1) Bir popülasyondaki rastgele oluşturulmuş kromozomlarla(probleme ait aday çözümler) çözüme başlanır.

2) Popülasyondaki her kromozomun uygunluk değeri hesaplanır.

3) İstenilen yavru adedi elde edilinceye kadar aşağıdaki adımlar tekrarlanır.

a) Mevcut popülasyondan bir çift kromozom seçilir, bireyin seçilme olasılığı uygunluk değerinin artışına göre artar. Seçilim değiştirilerek yapılır, yani aynı kromozomun ebeveyn olmak için birden fazla seçileceği anlamına gelir.

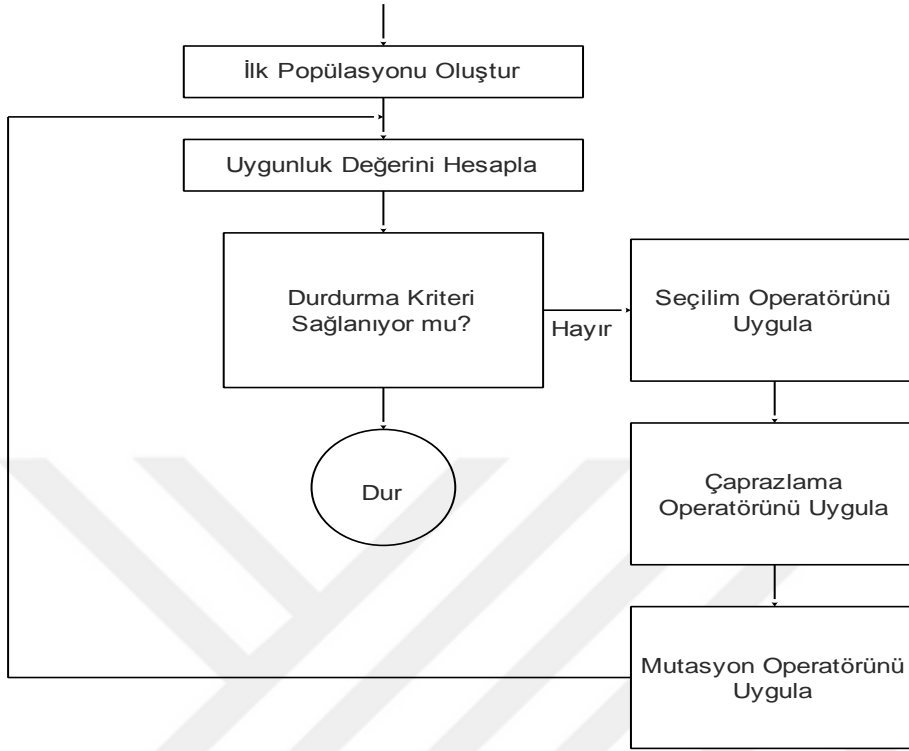
b) İki yavru oluşturmak için uygulamada belirlenen çaprazlama yöntemine göre seçilen çiftler çaprazlanır.

c) Oluşan iki yavru mutasyon oranına göre mutasyona uğratılır ve ortaya çıkan kromozomlar yeni popülasyona yerleştirir.

4) Mevcut popülasyon yeni popülasyonla değiştirir.

5) 2. adıma git.

Bu işlem uygun bir çözüm bulunana kadar veya önceden tanımlanmış bir tekrarlama ulaşınca kadar devam eder (Ulak 2010: 40).



Şekil 5. GA'nın Akış Diyagramı

Bu sürecin her tekrarlanması bir nesil olarak adlandırılır. Nesil üzerinde yapılan bu tekrarlama işlemi genellikle her tekrarlama farklı sonuçlar üreteceği için, her aşama çalışmanın performansı için büyük rol oynar. Burada popülasyon büyüklüğü ve mutasyon ve çaprazlama olasılığı gibi dikkat edilmesi gereken çok sayıda ayrıntı vardır ve algoritmanın başarısı bu ayrıntılara büyük ölçüde bağlıdır (Mitchell 1998: 9).

3.1.3 Genetik Algoritma Operatörleri

GA'nın çalışması, karar değişkenlerini temsil eden rastgele dizilerin bir popülasyonu ile başlar. Popülasyona daha sonra GA'nın üç ana operatörü olan çoğalma, çaprazlama ve mutasyon operatörleri yeni bir popülasyon oluşturmak için uygulanır (Mathew 2012: 5).

3.1.3.1 Seçilim Operatörü

Bir kodlamaya karar verdikten sonra, genetik algoritmanın ikinci aşaması seçilimin nasıl yapılacağıdır. Yani gelecek nesildeki bireyleri oluşturmak için popülasyondaki bireylerin nasıl seçileceği ve kaç tane bireyin oluşturulacağıdır (Mitchell 1998: 124). Seçilim operatörü, çaprazlanacak olan bireyleri seçmekle görevlidir (Pezzella, Morgantia 2007: 3208). Seçilim operatörü, bir popülasyonda iyi bireyler seçer ve çaprazlama havuzu oluşturur. Böylelikle üreme operasyonunda, başarılı yapıdaki bireylerin kopyaları daha sık üretilir. Genellikle bir popülasyon da uygulanan ilk operatördür. GA literatüründe bir dizi seçilim operatörü mevcut, ancak bunların hepsinde ana fikir, programcının seçtiği yöntemle göre bireylerin mevcut popülasyondan alınması ve bunların kopyasını eşleştirme havuzuna olasılıklı bir şekilde eklenmesidir.

GA yönteminde, çaprazlama havuzuna dahil edilecek kromozomları seçmek için kullanılan GA literatüründe iyi bilinen üç seçilim yöntemi bulunmaktadır. Bunlar; Rulet Tekerleği Yöntemi, Turnuva Yöntemi ve Sıralama Yöntemidir (Pezzella, Morgantia 2007: 3208).

Rulet Tekerleği Yöntemi:

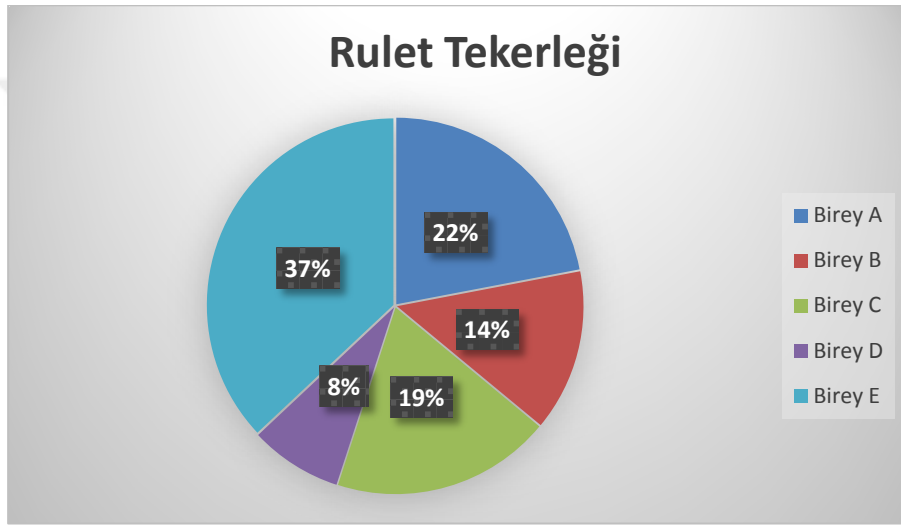
Rulet tekerleği seçilimi, orana göre seçilim stratejisini uygulayan yaygın bir tekniktir (Maulik, Sanghamitra 2000: 1458).

Rulet tekerleğine göre, daha büyük uygunluk değerine sahip olan bireylerin hayatta kalma olasılığı daha yüksektir. Seçme işlemi uygunluk değerinin hesaplanması ile başlar (Goldberg 1989: 6). Popülasyondaki bütün bireylerin uygunluk değeri hesaplanır ve bu uygunluk değerleri toplanarak popülasyonun toplam uygunluk değeri bulunur. Her bireyin uygunluk değeri tek tek toplam uygunluk değerine bölünerek her birinin seçilme olasılıkları elde edilir.

Kromozomların seçilim olasılıkları, bir pasta grafik yüzdesi gibi bir rulet çarkına yerleştirilir ve çark döndürülür. Kişi, bir rulet masasındaki iğnenin gösterdiği noktaya göre seçilir. Pastada daha büyük yüzdeli olanın seçilmesi daha olasıdır (Abidin 2013: 13).

Tablo 1. Rulet Tekerleđi Yöntemi

Bireyler	Uygunluk değeri	Seçilme olasılığı	Seçilme olasılığı (%)
Birey A	750	$750/3360 = 0.22$	22
Birey B	470	$470/3360 = 0.14$	14
Birey C	650	$650/3360 = 0.19$	19
Birey D	240	$240/3360 = 0.08$	8
Birey E	1250	$1250/3360 = 0.37$	37
TOPLAM	3360	1	100



Şekil 6. Rulet Tekerleđi Pasta Grafiđi

Sıralama Yöntemi:

Sıralama seçim yöntemi hızlı bir şekilde bazı uygunluk değerleri yüksek bireylere odaklanmayı önlemek için kullanılan alternatif bir yöntemdir. Baker (1985) tarafından önerilen yöntemde, popülasyondaki bireyler uygunluđa göre sıralanır ve her bireyin beklenen değeri uygunluk değerinden ziyade sırasına bađlıdır. Ama bireylerin uygunluk değerleri arasındaki farkın önemini kaybetmesi dezavantaja neden olabilir. Bazı durumlarda, bir bireyin en yakın rakibinden çok daha iyi olduğunu bilmek önemlidir (Mitchell 1998: 127). Rulet tekerleđi yönteminde popülasyondaki bir bireyin uygunluk değeri diđer bireylerden çok büyük olması durumunda uygunluk değeri yüksek olan birey rulet tekerleđinde çok büyük bir alanı

kapsar. Bu durumda uygunluk değeri küçük olan bireyin seçilme ihtimali oldukça düşer (Karcı 2002: 16).

Sıralama yönteminde bireyler uygunluk değerlerine göre sıralanır. Bireylerin içerisinde en kötü uygunluk değerine sahip olan 1 değerini en kötü ikinci uygunluk değerine sahip olan birey 2 değerini şeklinde ilerler son olarak en iyi uygunluk değerine sahip olan birey n değerini alır. Bireylere yapılan bu sıralama atamasına göre seçilme olasılıkları belirlenir. Popülasyondaki birey sayısı arttıkça en iyi birey ile en kötü birey arasındaki seçilme ihtimali oranı o derece artacaktır (Çolak 2015: 17). Bu yöntemin bir örneği aşağıdaki gösterilmektedir.

Tablo 2. Sıralama Yöntemi

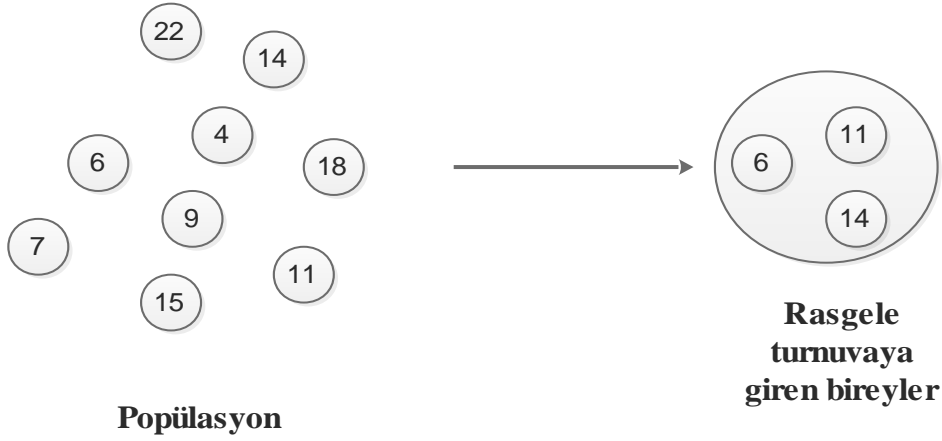
Bireyler	Uygunluk Değeri	Sıra	Seçilme İhtimali
Birey A	70	4	4/15=%26,66
Birey B	42	3	3/15=%20
Birey C	85	5	5/15=%33
Birey D	24	2	2/15=%13,3
Birey E	12	1	1/15=%6,66

Turnuva Yöntemi :

Sıralama yöntemi, tüm popülasyonu uygunluk değerine göre sıralamayı gerektirir; bu da potansiyel olarak zaman alan bir prosedürdür. Turnuva seçilimi, seçim baskısı açısından sıralama seçilimine benzer, ancak sayısal olarak daha verimli ve paralel uygulanmaya daha uygundur (Mitchell 1998: 127).

Az uyuma sahip bireyleri hızla temizleme ve hızlı bir şekilde optimuma yaklaşma avantajına sahiptir. Turnuva seçim yönteminde turnuvaya giren birey sayısında değişiklik yapılabilir. Ancak her durumda iki birey birbirine karşı yarışır (Yeo, Agyei 1998: 271).

Bu seçimde, popülasyondaki bireyler rastsal olarak seçilir. Seçilen bireyler, gruplanır ve bireyler arasında seçim işlemi yapmak için rekabete sokulur. Bu süreç istenilen birey sayısına ulaşıncaya kadar devam eder. Gruplar uygulamanın durumuna göre iki veya daha fazla bireyden oluşabilir. Turnuvaya sokulan bu bireyler, orijinal popülasyona geri döner ve tekrar seçilebilir (Aydın 2008: 13). Bu yöntemin bir örneği aşağıda gösterilmiştir.



Şekil 7. Turnuva Seçim Yöntemi

Bu örnekte popülasyon içindeki bireylerin üzerindeki sayılar bireyin uygunluk değerlerini temsil etmektedir. Rasgele seçilerek turnuvaya giren 6, 11 ve 14 uygunluk değerlerine sahip bireylerden uygunluk değeri en yüksek olan birey seçilir.

3.1.3.2 Çaprazlama Operatörü

GA performansı çeşitli operatörlere bağlıdır. Çaprazlama operatörü de bunlardan biridir. Bu operatör seçim işlemi sonrası hayatta kalan bireyler arasında çaprazlama işlemi yapar. Bu işlem daha iyi bireyler üretmek amacıyla, rastgele olarak belirlenen iki birey arasında gen alışverişi yaparak yeni bireylerin oluşturulmasıdır. Çaprazlama işlemine katılan iki bireye ebeveyn bireyler, çaprazlama sonucu elde edilen bireylere çocuk bireyler denir. Bu çocuk bireyler yeni nesil ebeveyn bireyleri oluşturur. Literatürde yer alan bazı çaprazlama yöntemleri aşağıda anlatılmıştır.

Tek Noktalı Çaprazlama:

Öncelikle, kromozomlar üzerinde rastgele çaprazlama noktası belirlenir. Sonra, çaprazlama noktasından önceki genetik materyal değişmeden kalırken, çaprazlama noktasından sonraki genetik materyal ebeveynler arasında değiştirilir (Aydın 2008: 14). İki bitlik kromozomların tek noktalı çaprazlama şekli aşağıdaki gibidir.

	1 0 1 1 1 1 0 0 1 0
Ebeveyn 2	1 1 0 1 0 0 1 0 1 1



	1 0 1 1 1 1 1 0 1 1
Çocuk 2	1 1 0 1 0 0 0 0 1 0

Şekil 8. Tek Noktalı Çaprazlama

Çift Noktalı Çaprazlama:

Bu yöntemde iki ayrı çaprazlama noktası belirlenir. Bu noktalar arasında kalan genetik özellikler ebeveynler arasında yer değiştirir.

	1 0 1 1 1 1 1 0 1 1
Ebeveyn 2	1 1 0 1 0 0 0 0 1 0



	1 0 1 1 0 0 1 0 1 1
Çocuk 2	1 1 0 1 1 1 0 0 1 0

Şekil 9. Çift Noktalı Çaprazlama

Kısmen Uyumlu Çaprazlama (Partially Mapped Crossover PMX):

Bu yöntem Goldberg tarafından geliştirilmiş ve ilk olarak gezgin satıcı probleminde kullanılmıştır. Bu çaprazlama yönteminde, iki ebeveyn rastgele seçilir ve rastgele iki çaprazlama noktası oluşturulur (Aydın 2008: 14). Aşağıdaki örnekte gösterildiği gibi; öncelikle belirlenen çaprazlama noktasının arasında kalan değerler yer değiştirilir. Daha sonra * işareti ile boş bırakılan genler diğer kromozoma bakılarak sırayla doldurulur.

	2 4 7 3 5 8 6 9 1
Ebeveyn 2	5 6 2 4 8 1 3 7 9



	2 * 7 4 8 1 6 9 *
Çocuk 2	* 6 2 3 5 8 * 7 9



	2 3 7 4 8 1 6 9 5
Çocuk 2	4 6 2 3 5 8 1 7 9

Şekil 10. Kısmen Uyumlu Çaprazlama

Sıralı Çaprazlama (Order Crossover, OX):

Bu çaprazlama yöntemi 1980'li yıllarda Davis tarafından gezgin satıcı problemleri için geliştirilmiştir (Taşkın, Emel 2009: 53). Bu yöntemde öncelikle iki ebeveyn için belirlenen çaprazlama noktaları arasında kalan değerler çocuk bireylere doğrudan aktarılır.

	2 4 7 3 5 8 6 9 1
Ebeveyn 2	5 6 2 4 8 1 3 7 9



	*** 3 5 8 ***
Çocuk 2	*** 4 8 1 ***

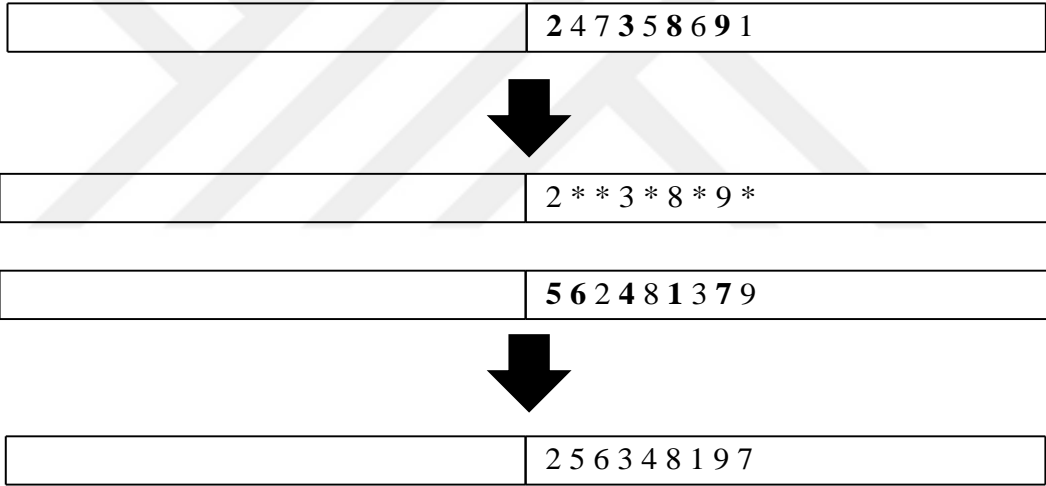
Sonrasında Çocuk 1 için Ebeveyn 2'nin ikinci çaprazlama noktasından başlanarak değerler 3 7 9 5 6 2 4 8 1 şeklinde sıralanır. Bu sıralamadan Çocuk 1'in mevcut değerleri olan 3, 5, 8 çıkarılır ve sıralama aynı kalacak şekilde Çocuk 1'in ikinci çaprazlama noktasından başlanarak bu değerler yazılır. Aynı işlem Ebeveyn 1'den alınarak Çocuk 2 için yapılır.

	2 4 1 3 5 8 7 9 6
Çocuk 2	7 3 5 4 8 1 6 9 2

Şekil 11. Sıralı Çaprazlama

Pozisyona Dayalı Çaprazlama (Position Based Crossover, PBX):

Bu çaprazlama yönteminde Ebeveyn 1 kromozomundan rastgele genler çocuk kromozoma aktarılır. Yapılan bu işlemde sonra çocuk kromozomda boş kalan yerlere sıra ile Ebeveyn 2 kromozomundan alınan genler aktarılır. Aşağıdaki örnekte gösterildiği gibi; ilk adımda çocuk kromozom Ebeveyn 1 kromozomundan 2, 3, 8 ve 9 genlerini alır. Daha sonra sıra ile Ebeveyn 2 kromozomundan 5, 6, 4, 1 ve 7 genlerini alır.



Şekil 12. Pozisyona Dayalı Çaprazlama

Çevrim Çaprazlama (Cycle Crossover, CX):

Bu çaprazlama yönteminde herhangi bir çaprazlama noktası belirlenmez. Bu yöntemin çalışma şeklini aşağıdaki örnek üzerinde anlatacak olursak; öncelikle Ebeveyn 1'in en başında olan değerden işleme başlanmaktadır. Buna göre 2 değeri ilk değerimiz olacaktır. 2 değerinden başlayarak oklar bir düzen şeklinde ilerlemektedir. 2 değerinin bulunduğu konumun Ebeveyn 2'deki karşılığı olan 5 değerine geçilir. Sonra Ebeveyn 2'deki 5 değerinden Ebeveyn 1'deki 5 değerine

geçilir. Yani kısaca ok önce bulunduğu konumun karşısına sonrasında o konumdaki değer diğer ebeveyndeki karşılığına gider. Bu döngü öncesinde gidilen bir konuma tekrar gidinceye kadar devam eder. Bu örnekte son olarak Ebeveyn 1'deki 7 değerinden Ebeveyn 2'deki 2 değerine geçilmiş ve sonrasında Ebeveyn 1'deki 2 değerine geçmesi gerekiyor ama daha önce o konuma gidildiği için döngü tamamlanır. Döngü sonunda okla gidilmeyen değerler iki ebeveyn arasında yer değiştirir.

	2	4	7	3	5	8	6	9	1
Ebeveyn 2	5	6	2	4	8	1	3	7	9

	4	7	3	5	8	6	9	1
--	---	---	---	---	---	---	---	---

	6	2	4	8	1	3	7	9
--	---	---	---	---	---	---	---	---

	2	6	7	4	5	8	3	9	1
Çocuk 2	5	4	2	3	8	1	6	7	9

Şekil 13. Çevrim Çaprazlama

3.1.3.3. Mutasyon Operatörü

Mutasyon bir kromozomun geliştirilmesinin son adımıdır. Mutasyon, bir kromozomun kendisinde değişiklik yapması dışında, çaprazlama operatörüne benzemektedir. Çoğu mutasyon stratejisi rastgele seçim yaklaşımını takip eder (Abdelhalim, Ghada 2016: 1402). Mutasyon operatörü genetik arama sürecine rastgele bir şekilde yeni bilgiler ekler ve sonuç olarak, yerel optimumda sıkışmayı önlemeye yardımcı olur. Çaprazlama operatörünün kullanılması nedeniyle popülasyon homojen olma eğiliminde olabilir, popülasyona çeşitlilik kazandırmak için kullanılan bir operatördür. Bu bakımdan mutasyon operatörüne, rastgele genetik bilgiyi rahatsız etme süreci diyebiliriz (Mathew 2012: 8).

	1 0 1 1 1 1 1 0 1 1
Mutasyondan sonra	1 0 0 1 1 1 1 0 1 1

	2 5 6 3 4 8 1 9 7
Mutasyondan sonra	2 5 9 3 4 8 1 6 7

Şekil 14. Mutasyon Operatörü

3.1.3.4 Diğer Operatörler

Elitizm:

Bireylere uygulanan çaprazlama ve mutasyon işlemlerinden sonra bireylerin yapısında büyük değişimler olur. Bu değişim, problemin çözümü için uygun olan bireylerin yok olmasına sebep olabilir. Bu durumu engellemek için elitizm yöntemi kullanılır. Bu yöntem mevcut popülasyondaki iyi bireyleri belirlenen oranda kopyalayarak bir sonraki popülasyona doğrudan aktarır.

Tamir Fonksiyonu:

Çaprazlama ve mutasyon işlemlerinden sonra ortaya çıkan bireylerin bazılarının yapısında bozulma meydana gelebilir. Böyle bir durumda bireyler problemin çözümünden uzaklaşır. Tamir fonksiyonu kullanılarak kromozomların bozulan gen yapıları tamir edilir.

3.2 Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)

PSO, evrimsel optimizasyon yöntemlerinden biridir. İlk olarak 1995 yılında Kennedy ve Eberhart (1995) tarafından geliştirilen popülasyon tabanlı bir optimizasyon yöntemidir. PSO, kuş sürüsü ve balık eğitimi gibi sosyal etkileşim ve iletişim benzetmesine dayanmaktadır.

Temel olarak sürü zekâsına dayanan bir algoritma olduğu için, bir PSO algoritmasındaki üyeler en iyi performansa sahip olan grup liderini takip etme eğilimindedir.

PSO, GA ile birçok ortak noktayı paylaşmaktadır. Her iki algoritma da sistem rastgele oluşturulmuş bir popülasyon grubuyla başlar ve her ikisi de popülasyonu değerlendirmek için uygunluk değerlerine sahiptir. Her iki algoritma da optimum değer için jenerasyonları güncelleyerek rastgele tekniklerle arama yapar. Fakat diğer evrimsel yöntemlerin tersine PSO'da çaprazlama ve mutasyon gibi filtreleme operatörleri yoktur.

PSO algoritması kolayca uygulanabilir ve işlemci hızı gereksinimi düşük olduğu için hesaplama maliyeti azdır (Eberhart vd. 1996). Buna ek olarak PSO algoritması incelenmekte olan amaç fonksiyonunun gradyan bilgisine ihtiyaç duymaz sadece temel matematiksel operatörleri kullanır. PSO algoritmasının birçok genel optimizasyon problemi için etkili bir yöntem olduğu kanıtlanmıştır hatta bazı durumlarda diğer evrimsel hesaplama yöntemlerin karşılaştığı zorluklarla karşılaşmamaktadır (Kennedy, Eberhart 1995).

Evrimsel hesaplama tekniklerinde re-kombinasyon, mutasyon ve seçim operatörleri olmak üzere üç ana operatör söz konusudur. PSO algoritmasında doğrudan re-kombinasyon operatörü bulunmamaktadır. Fakat sürüde bulunan parçacıkların pozisyonunun, sürünün en iyi pozisyona sahip olan parçasına doğru rasgele hızlanması, evrimsel hesaplama tekniklerindeki re-kombinasyon yöntemine benzemektedir (Eberhart, Shi 1998; Rechenberg 1994; Schwefel 1995).

Bir PSO algoritmasında her bir üyeye parçacık denir ve parçacıklardan oluşan popülasyona da sürü denir. PSO'da bulunan parçacıkların, her birinin ayrı bir hızı vardır. Her bir parçacık kendi hızını, komşularının tecrübesinden yararlanarak sürüdeki en iyi parçacığa göre yeniler. Böylece sürüde bulunan bireylerin pozisyonu, bir öncekine göre daha iyi duruma gelir ve bu süreç hedefe ulaşmaya kadar devam eder (Kennedy, Eberhart 1995).

PSO algoritmasındaki konum güncelleme işlemi, bir çeşit GA yöntemindeki mutasyona benzemektedir. Söz konusu bu mutasyon benzeri işlem, hem PSO algoritması için hem de GA için çok yönlüdür.

PSO algoritması, evrimsel yöntemlerin aksine en uygun olanın hayatta kalması kavramını kullanmayan tek evrimsel algoritmadır. Doğrudan seçim işlevini kullanmaz. Böylece daha düşük uygunluk derecesine parçacıklar optimizasyon

sırasında hayatta kalabilir ve durum uzayının herhangi bir noktasına hareket edebilir (Eberhart, Shi 1998).

PSO, sipariş miktarı belirleme, çizelgeleme problemleri, güç ve voltaj kontrolü, tedarik seçimi ve sıralama problemleri (Kennedy, Eberhart 1995) gibi birçok problemin optimizasyonunda başarı ile kullanılmıştır. Temel olarak PSO algoritması aşağıda açıklanmıştır:

Adım 1: Popülasyonun oluşturulması; başlangıç sürüsü, rasgele üretilen başlangıç pozisyonları ve hızları ile birlikte oluşturulur.

Adım 2: Uygunluk değerlerinin hesaplanması; sürü içerisindeki tüm parçacıkların uygunluk değerleri hesaplanır.

Adım 3: En iyi üyenin bulunması; Her bir parçacık için mevcut jenerasyondan yerel en iyi (lbest) bulunur.

Adım 4: Global en iyinin bulunması; Mevcut jenerasyondaki yerel eniyiler içerisinde global en iyi (gbest) seçilir.

Adım 5: Pozisyon ve hızların yenilenmesi; pozisyon ve hızlar aşağıdaki gibi yenilenir.

$$V_{id} = W * V_{id} + c_1 * r_1 * (P_{id} - X_{id}) + c_2 * r_2 * (P_{gd} - X_{id})$$
$$X_{id} = X_{id} + V_{id}$$

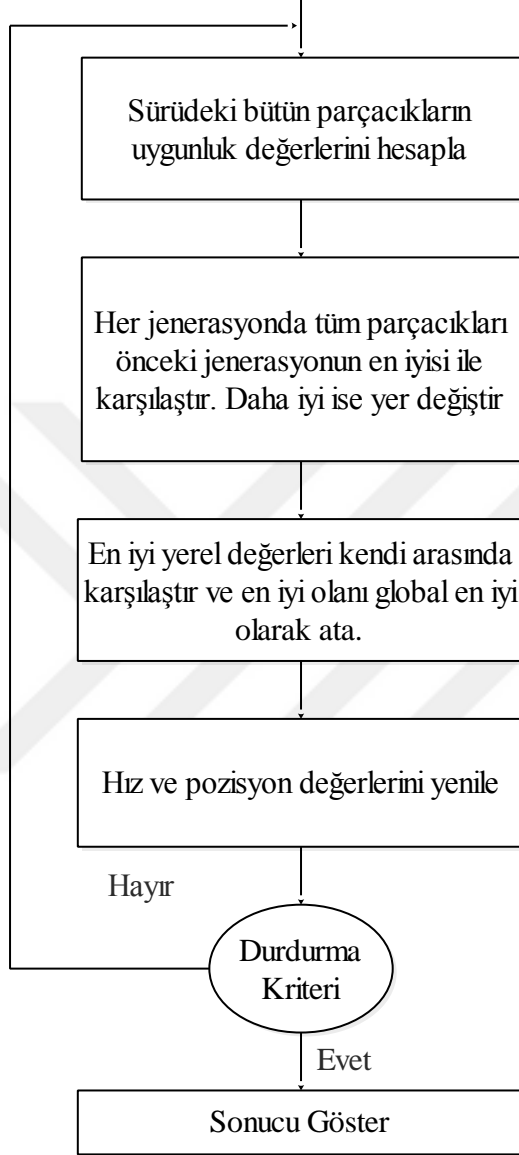
Burada X_{id} pozisyon ve V_{id} hız değerlerini verirken, r_1 ve r_2 değerleri [0,1] aralığında rasgele üretilmiş sayılardır. W atalet ağırlık değeri ve c_1 , c_2 pozitif sabit değerlerdir, genellikle 2'ye yakın değerler kabul edilir.

Adım 6: Durdurma kriterleri sağlanıncaya kadar adımları tekrar et; Durdurma kriteri sağlanıncaya kadar 2, 3, 4, 5 adımları tekrar edilir. Şekil 15'de PSO algoritmasının akış diyagramı gösterilmiştir.

PSO algoritmasının yerel (local) ve global bazda olmak üzere iki çeşidi geliştirilmiştir. Global bazda konum bilgisi paylaşım yönteminde her bir parçacık en iyi konumda bulunan parçacığa doğru ilerlemektedir, söz konusu parçacık "gbest" olarak adlandırılmaktadır. Diğer taraftan "lbest" olarak adlandırılan konum bilgisi paylaşım yönteminde ise paylaşım yalnızca parçacıkların komşuları ile yerel (local) bazda gerçekleşmektedir (Eberhart vd. 2001).

PSO algoritmasının kombinasyonel optimizasyon problemleri üzerindeki uygulamaları hala kısıtlı olmasına rağmen, PSO algoritması basit konsepti ve düşük işlem maliyeti ile avantajlı bir yöntemdir. Ayrıca, PSO'nun sürekli fonksiyonları optimize etmek için güçlü bir yöntem olduğu söylenebilir.





Şekil 15. Parçacık Sürü Optimizasyonu Akış Diyagramı

3.3 Yapay Arı Kolonisi Algoritması (Artificial Bee Colony Algorithm)

ABC algoritması, bal arısı kolonilerinin kendilerine has olarak sürdürdükleri zeki tutumlarını örnek alarak, arıların besin ararlarken kullanmış oldukları yöntemlerden esinlenerek meydana getirilmiş bir optimizasyon algoritması olarak

kabul görmektedir. Sürünün kendine özgü zekasına dayanan söz konusu algoritma, doğada sürü olarak hareket edebilen arıların besin bulmalarında sergilemiş oldukları tutumları temel alarak problemlere çözüm aramaktadır.

Arı sürülerinin zeki davranış tutumları ve besin arama süreci içerisindeki tutumlarını modelleyen Karaboğa, ABC algoritmasını geliştirmiştir (Karaboğa 2005). ABC algoritmasında yapay arılar, işçi arılar, izleyici arılar ve kâşif arılar olmak üzere üç gruba ayrılır. Bir besin kaynağından yararlanan arı işçi arı olarak sınıflandırılır. İşçi arılar, kovanda bekleyen ve işçi arıların danslarını izleyen, izleyici arılar ile bilgi paylaşırlar. İzleyici arılar daha sonra paylaşılan bilgi ile o yiyecek kaynağının kalitesine orantılı bir gıda kaynağı seçer. Böylece iyi besin kaynakları kötü olanlardan daha fazla arı çekecektir. Kâşif arılar, kovanın yakınında rastgele yeni besin kaynakları ararlar. Kâşif ya da izleyici arı bir besin kaynağı bulduğunda, bulunan besin kaynağı kullanılır. Bir besin kaynağı tamamen kullanıldığında, ilgili tüm işçi arılar o pozisyondan ayrılır ve tekrar kâşif arı olurlar. Böylece, kâşif arılar “keşfetme” işini yürütürken, izleyici arılar “tüketim” işini yapar. Algoritmada, bir besin kaynağı, optimizasyon problemine olası bir çözüme karşılık gelir ve bir yiyecek kaynağının nektar miktarı, ilgili çözümün uygunluk değerine karşılık gelir (Akay, Karaboğa 2012) .

ABC algoritmasında koloninin ilk yarısı işçi arılardan, diğer yarısı ise izleyici arılardan oluşur. Her bir besin kaynağı için yalnızca bir tane işçi arı olduğu varsayıldığı için işçi arı sayısı, besin kaynağı sayısına eşittir. Bu nedenle, izleyici arı sayısı aynı zamanda söz konusu çözüm sayısına eşittir. ABC algoritması, rastgele oluşturulmuş bir grup besin kaynağı ile başlar. ABC algoritmasının temel prosedürü aşağıdaki şekilde tanımlanabilir (Zhang, Wu 2011).

Adım 1: Başlangıç yiyecek kaynağı bölgelerinin oluşturulur.

Adım 2: Her bir işçi arı bir besin kaynağı üzerinde çalışmaya başlar.

Adım 3: Her izleyici arının, işçi arılar tarafından paylaşılan nektar bilgilerine göre bir besin kaynağı seçer.

Adım 4: Besin kaynaklarını rastgele arayacak olan kâşif arılarını belirleme.

Adım 5: Sonlandırma koşulu sağlanıncaya kadar Adım 2'ye geri dön.

Her adım için ayrıntılı açıklama aşağıda belirtilmiştir.

(1) *Başlatma aşaması.* Besin kaynağı sayısı, başlangıç çözümleri rastgele oluşturulmuş D boyutlu gerçel vektörlerdir. $x_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,D}\}$ aşağıda belirtilen eşitlik ile elde edilen i 'inci besin kaynağını temsil eder.

$$x_{i,d} = x_d^{min} + r \times (x_d^{max} - x_d^{min}), d = 1, \dots, D \quad (1)$$

Burada r , $[0,1]$ aralığındaki rastgele bir sayıdır ve x_d^{min} ve x_d^{max} sırasıyla boyut d için alt ve üst sınırlardır.

(2) *İşçi arı aşaması.* Bu aşamada her bir işçi arı bir çözüm ile eşleştirilir. Yeni bir çözüm bulmak için (yeni besin kaynağı), çözümde (orijinal besin kaynağı) rastgele bir değişiklik uygular. Yeni çözüm v_i , diferansiyel genleşme kullanarak x_i 'den üretilir.

$$v_{i,d} = x_{i,d} + r' \times (x_{i,d} - x_{k,d}) \quad (2)$$

Burada, $d, \{1, \dots, D\}$ kümesinden rastgele seçilir, $k, \{1, \dots, SN\}$ kümesinden rastgele seçilir, $k \neq i$ ve $r', [-1,1]$ aralığında rastgele bir sayıdır.

v_i elde edildiğinde, değerlendirilir ve x_i ile karşılaştırılır. v_i 'nin uygunluk değeri (fitness) x_i 'den daha iyi olması durumunda (yeni besin kaynağının nektar miktarı eskisinden daha yüksek ise) arı eski çözümü unuttur ve yeni çözümü ezberler. Aksi durumda, x_i üzerinde çalışmaya devam eder.

(3) *İzleyici arı aşaması.* Tüm işçi arılar yerel aramalarını tamamladıkların, besin kaynağının nektar bilgilerini her biri olasılıklı bir şekilde bir besin kaynağı seçecek olan izleyici arılar ile paylaşırlar. Bir izleyici arının besin kaynağını (x_i) seçtiği olasılık p_i aşağıdaki gibi hesaplanır.

$$p_i = \frac{f_i}{\sum_{i=1}^{SN} f_i} \quad (3)$$

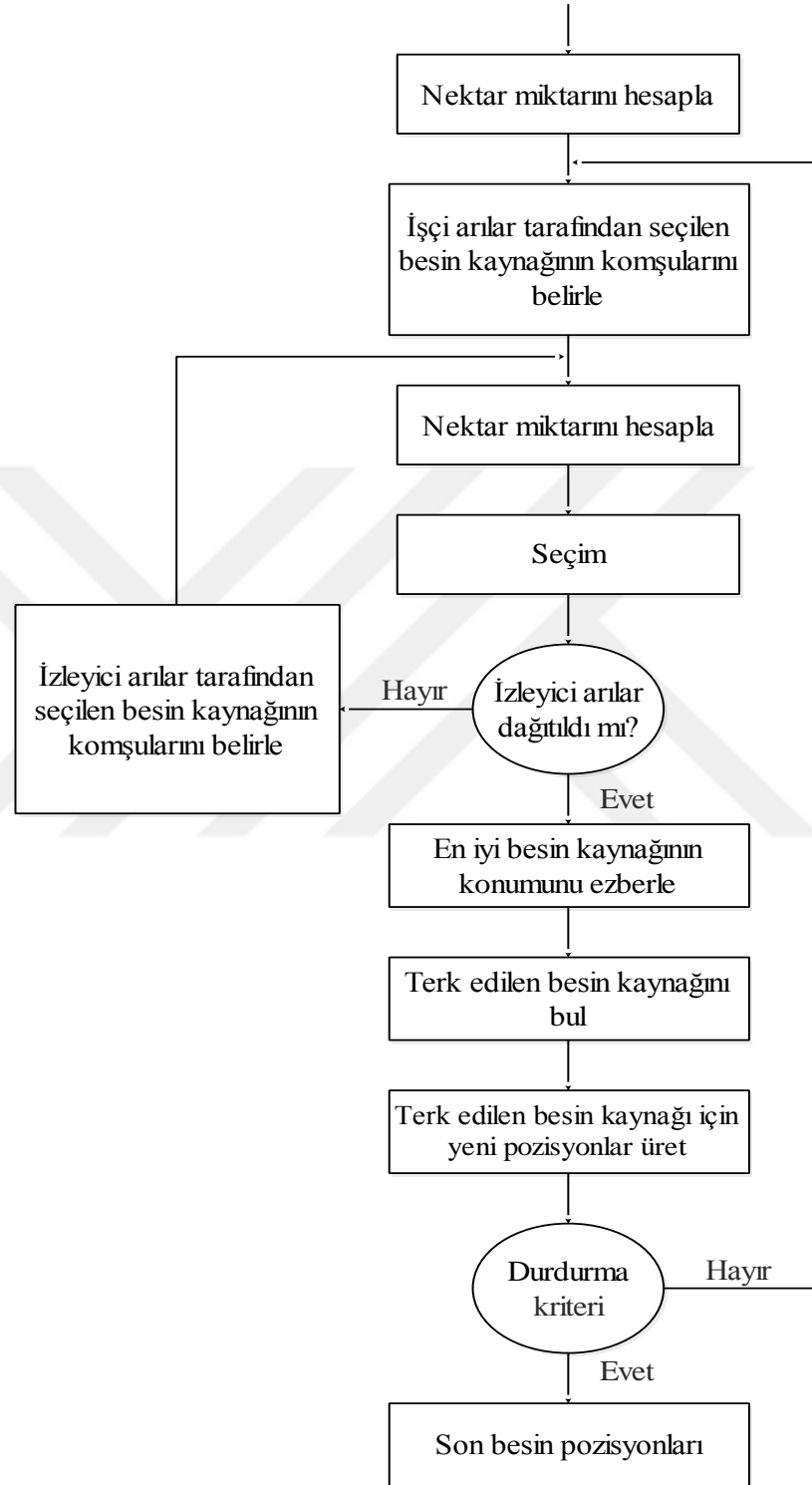
Burada, f_i , x_i 'nin uygunluk deęeridir (fitness), izleyici arılar daha yüksek nektar miktarına sahip besin kaynaklarını seçme eğilimindedir.

İzleyici arı x_i besin kaynağını seçtiğinde, eşitlik 2'ye göre x_i ile ilgili yerel arama yapar. Bir önceki durumda da olduğu gibi, eđer deęiştirilmiş çözüm daha iyi uygunluk deęerine (fitness) sahip ise yeni çözüm x_i 'nin yerine geçecektir.

(4) *Kâşif arı aşaması.* ABC algoritmasında, bir çözümün kalitesi önceden belirlenmiş sayıda test 'den sonra geliştirilmemesi durumunda, besin kaynağının terk edildiđi varsayılır ve işçi arılar, kâşif arı haline gelir. Kaşif arı daha sonra eşitlik (1)'i kullanarak rastgele besin kaynağı ortaya koyar.

Algoritmanın anlaşılmasını kolaylaştırmak için, aşağıdaki şekilde ABC algoritmasının akış diyagramı verilmiştir (Akay, Karaboęa 2012).

pozisyonları



Şekil 16. ABC Algoritması Akış Diyagramı



IV. BÖLÜM: UYGULAMA

Ders çizelgeleme probleminin çözümü için çalışmamızda GA, PSO ve ABC algoritması ayrı ayrı kodlanmıştır. Her ne kadar üç algoritma birbirlerinden farklı gibi gözükseler de her üç algoritmanın da popülasyon temelli olmaları, iteratif bir yapıda var olan çözümleri sürekli iyileştirerek en iyi çözüme yakınsama gibi ortak çalışma prensiplerine sahip olmaları nedeniyle çoğu yönden benzerlik göstermektedirler. Çözümlerin her birisi GA'da kromozom olarak adlandırılırken, PSO'da sürüdeki bireylerin yerlerini, ABC' algoritmasında ise besin kaynağını temsil etmektedir. Benzer şekilde uygunluk fonksiyonu her üç algortmada da çözümün kalitesini göstermektedir. Bu nedenle öncelikle her bir çözümün uygun şekilde kodlanması ve bir çözümün diğer çözümlerden daha iyi olup olmadığını gösteren bir uygunluk fonksiyonunun belirlenmesi gereklidir. Kullanılan algoritmalarından bağımsız olarak ders çizelgeleme problemin kodlanması ve uygunluk fonksiyonlarının belirlenmesi öncelikle ele alınmıştır.

4.1. Problemin kodlaması

Sezgisel algoritmalarda problemin ne şekilde kodlanacağı çok önemli bir aşamadır. Kodlama şekli bir noktada problemin bir çözümünü temsil etmektedir. Kodlama yapısının düzgün bir şekilde ifade edilmesi algoritmanın performansı açısından oldukça önemlidir. Ayrıca kodlama yapısı tasarlanırken kodlamanın boyutu, iyi çözümlerin bulunması açısından dikkat edilmesi gereken noktalardan biridir. Çözüm kodlanırken problemin tipine göre hangi yapının kullanılacağına karar verilir. Literatürde DÇP'nin çözümünde kullandığımız kodlama yapısına benzer kromzom yapıları kullanılmış ve uygun çözümler elde edilmiştir. Bu çalışmada her üç yöntem içinde kullanılan kodlama yapısı, derslikler satır ve zaman dilimleri sütun olacak şekilde oluşturulmuştur. Oluşturulan bu kodlama yapısının boyutu derslik*zaman periyodları şeklinde olacaktır.

A = Toplam kodlama boyutu

Z = Günlük zaman periyodu sayısı

C = Toplam derslik sayısı

G = Ders yapılan gün sayısı

$$A = Z \times C \times G$$

Fakültenin 40 adet dersliđi bulunmaktadır. Günlük zaman dilimi 2 ve ders yapılan gün sayısı 5'tir.

$40 \times 2 \times 5 = 400$ boyutunda bir kod yapısı oluşturulmuştur.

Oluşturulan kodlama yapısında zaman periyodları blok şeklinde düşünülmüştür.

1. Blok: 08.00-12.00 saatleri arası

2. Blok: 13.00-17.00 saatleri arası

1. Blok ve 2. Blok gündüz şubelerinin zaman periyodlarını temsil etmektedir. Gece şubelerinin dersleri gündüz şubelerinin derslerinin aynısı olduğundan gece derslerinin blokları kodlama yapısından çıkarılmıştır. Bulunan en iyi çözüme göre gündüz derslerinin bulunduğu derslikler ve ders saatlerine bakılarak gece şubelerinin ders programı tablosu hazırlanmıştır. Bu bloklar 4 saatlik zaman periyotları şeklinde oluşturulmuş ama bu bloklara 2 saatlik bir ders gelebilir. Örneđin; sabah blođuna 2 saatlik bir ders gelirse bu ders isteđe göre 09.00'da başlatılıp 11.00'da bitirilebilir veya 10.00'da başlatılıp 12.00'da bitirilebilir.

Kullandığımız kodlama yapısında satırların derslikleri, sütunların zaman periyodlarını temsil ettiđini söylemiştik. Oluşan bu matrisin hücreleri ise ders numaralarını tutmaktadır. Ayrıca bir derslikte herhangi bir zaman periyodunda ders yapılmadıđı zaman 0 deđeri ile temsil edilmektedir. Oluşturulan kodlama yapısının ilk hali 40 satır ve 10 sütundan oluşan 0 matrisi şeklindedir. Sonrasında ders numaraları bu kodlama yapısına aktarılmaktadır. Bu ders numaralarının ve 0 deđerlerinin her biri GA'da bir gen olarak düşünülür ve bir araya gelerek kromozom yapısını oluşturur. Bu kodlama yapısı sayesinde bazı kısıtlar ortadan kaldırılmıştır. Aşađıda bu kısıtlardan bahsedilmiş ve çözümlü gösteren kod yapısının bir örneđi Şekil 17'de gösterilmiştir.

Bir dersliđe aynı zaman diliminde birden fazla dersin atanmaması, uygun bir ders programının oluşması için kesinlikle sağlanması gerekir. Oluşturulan kodlama yapısı sayesinde bir derslikte herhangi bir zaman diliminde birden fazla dersin atanması ortadan kaldırılmıştır.

Gündüz şubelerinin derslerinin aynı günün akşamında gece şubeleri içinde yapılacağı dikkate alınarak bu kodlama yapısı oluşturulmuştur. Böylece aynı şubelerin gündüz ve gece derslerinin aynı zaman periyotlarına paralel şekilde atanması sağlanmıştır. Örneğin; herhangi bir gündüz şubesinin pazartesi günü 09.00'da başlayıp 12.00'da biten bir dersi, o şubenin gece şubesi içinde 17.00'da başlayıp 20.00'da bitecek şekilde olmuştur.

Bir dersin ilk ve son saatinin aynı derslikte veya aynı gün olmaması durumu için çözümü gösteren kod yapısına her ders için tek bir ders numarası atanmış ve zaman dilimleri bloklar halinde oluşturulmuştur. Böylece bu kısıt ile ilgili sorun ortadan kalkmıştır.

Zaman dilimleri bloklar halinde oluşturulmuştur. Böylece şubelerin gün içerisindeki dersleri arasında boşluk olması problemi ortadan kaldırılmıştır.

Kodlama matrisinin yapısı bir gün içerisinde en fazla iki ders yapılabilecek şekilde tasarlanmıştır. Bu sayede şubelerin günlük ders saatinin dengeli bir şekilde olması sağlanmıştır.

Derslikler \ Zaman periyotları										
1		14	33	21	44	0	0	131	19	42
2	0	33	8	38	46	52	4	0	40	0
3	0	76	0	0	0	0	91	0	1	220
4	11	21	46	0	189	0	55	0	93	0
5	61	0	33	6	0	71	0	111	0	0
6	49	0	0	0	2	0	0	0	18	88
7	22	8	0	219	0	0	64	0	199	0
...	0	127	0	139	0	67	0	66	7	0
...	79	0	0	77	0	0	124	0	0	3
...	51	0	94	9	0	86	92	0	99	0
40	0	28	99	0	5	0	101	0	14	0

Şekil 17. Kodlama Yapısı

4.2. Uygunluk deęerinin hesaplanması

Uygunluk deęeri bir çözümlerin olası dięer çözümlere oranla ne derecede iyi ve uygun olduęunun bir göstergesidir. Uygunluk deęeri, zaman çizelgeleme probleminin çözümünde kullanılan zorunlu ve esnek kısıtların ne kadar ihlal edildięini göstergesidir (Abdullah 2008: 258). Problemlerin çözümünde bireylerin uygunluk deęerlerini hesaplayan bir uygunluk fonksiyonu vardır. Uygunluk fonksiyonu, her bireyin uygunluk deęerini zorunlu ve esnek kısıtlara göre ayrı ayrı hesaplamaktadır.

Zaman çizelgeleme problemlerinde bireyin uygunluk deęeri, bireyin ihlal ettięi kısıtların ceza puanları toplanarak hesaplanır. Bundan dolayı bireyin uygunluk deęeri ne kadar küçükse çözüm için o kadar uygun olacaktır. Aşağıda bir bireyin uygunluk deęerinin hesaplanması ve zorunlu ve esnek kısıtlara verilen ceza puanları tablo şeklinde gösterilmiştir. Bu ceza puanları fakültenin her dönem hazırladıęı ders programı dikkate alınarak verilmiştir. Fakülte ders programı için kesinlikle sağlanması gereken kısıtlara 100 ceza puanı verilmiştir. Dięer kısıtlar için belirlenen ceza puanları önem dereceleri dikkate alınarak belirlenmiştir. Belirlenen bu ceza puanlarının amacı, algoritmanın problemi çözümü aşamasında hangi kısıtlara önem vermesi gerektięi ve iki kısıt arasında kaldığında hangisini tercih etmesi gerektięi konusunda algoritmayı yönlendirmektir.

$$f(b) = \sum_{j=1}^n C_j$$

b = Birey

f(b) = Bireyin uygunluk deęeri

n = Kısıt Sayısı

C_j = j. Kısıtın Ceza Puanı

Tablo 3. Zorunlu Kısıtlar

Zorunlu Kısıtlar	Açıklama	Ceza Puanı
Z1	Bir öğretim elemanının aynı zaman diliminde birden fazla derse atanması	100
Z2	Bir şubenin aynı zaman diliminde birden fazla derse atanması	100
Z3	4 saatlik bir dersin bütün saatlerinin aynı gün yapılması	100
Z4	Derslerin istenilen dersliklere atanmaması	100
Z5	Dersler, dersin öğrenci sayısına göre dersliklere uygun bir şekilde atanmaması	100
Z6	1.Sınıflar için uzaktan eğitim derslerinin yapılacağı saatlere başka ders atanması	100
Z7	Bir öğretim elemanı 2.Öğretim(gece) derslerinin 16.00'da başlamışını istemiyorsa o öğretim elemanına gece için üç saatlik iki ders atanması	100

Tablo 4. Esnek Kısıtlar

Esnek Kısıtlar	Açıklama	Ceza Puanı
E1	Bir öğretim elemanının istemediği bir zaman dilimine dersinin atanması	30
E2	Öğretim elemanı istekleri dikkate alınarak, bir öğretim elemanının belirlediği günlük ders saati sayısından fazla derse atanması	20
E3	Öğretim elemanı istekleri doğrultusunda bir öğretim elemanının derslerinin gün olarak art arda olmaması	10

Yukarıdaki tablolarda bulunan kısıtların her biri uygunluk fonksiyonun hesaplanmasında kullanılmıştır. Bu kısıtların ayrıntılı açıklaması ve kodlama mantıkları aşağıda anlatılmıştır.

Bir öğretim elemanının aynı zamanda birden fazla derste bulunması imkânsızdır. Bu kısıtın tamamen sağlanması gerekir. Bu kısıt ile zaman dilimlerini temsil eden sütunların tek tek kontrolü yapılmıştır. Bir öğretim elemanının aynı sütun içerisinde birden fazla dersi varsa çakışmaya neden olacaktır. Bulunan her çakışma için uygunluk değerine ceza puanı eklenmiştir.

Bir şubenin aynı zamanda birden fazla derste bulunması imkânsızdır. Bu kısıt aynı sütunda bir şubenin birden fazla dersinin olup olmadığını kontrol eder. Sütun

içerisinde aynı şubenin birden fazla dersi varsa bu durum cezalandırılır. Bireyin ceza puanı ne kadar yüksek olursa seçilme olasılığı aynı oranda düşecektir.

Dersin laboratuvar ihtiyacı olup olmadığına bakılarak uygun bir dersliğe atanması amaçlanır. Fakültede ki her ders daha önceden belirlenmiş dersliklerde yapılmaktadır. Bu veriler excel dosyasına eklenmiştir ve dersin istenilen dersliğe atanmaması kısıtı ile kontrol edilmiştir. Böylece özel derslik gerektiren derslerin uygun dersliğe atanması sağlanmıştır.

Ders verileri için excel dosyası hazırlanırken her derse bir numara verilmiştir. dört saatlik dersler, iki saatlik parçalara ayrılarak iki farklı numara verilerek algoritma tarafından farklı dersler gibi yorumlanmaları sağlanmıştır. Dört saatlik bir dersin bütün saatlerinin aynı gün yapılmasını engellemek için bu iki ders parçasının aynı gün olması durumu cezalandırılmaktadır. Fakat herhangi bir öğretim elemanın istediği doğrultusunda dört saatlik bir ders aynı gün yapılabilir.

Dersin istenilen dersliğe atanmaması kısıtı, her dersin atanması gereken derslikleri kontrol etmektedir. Hangi dersin hangi dersliğe atanması konusunda fakülte için önceden hazırlanmış olan ders programı verileri kullanılmıştır. Fakültenin ders programında her şube için bir derslik belirlenmiştir. Bu kısıtın kod yapısı kromozomun her satırını kontrol ederek o satırda yani o derslikte olmaması gereken her ders için uygunluk değerine ceza puanı eklemektedir.

1. sınıflar için İngilizce ve Türk Dili, 2. sınıflar için İnkılap Tarihi dersleri sanal sınıf ortamında yapılmaktadır. Bu dersler için 1. sınıflar için Cuma bütün gün 2. Sınıflar için Cuma günü sabah ayrılmıştır. Bu belirlenen zamanlarda sınıfların normal dersi olursa uygunluk değerine ceza puanı eklenmektedir.

Bazı öğretim elemanları, öğretim elemanları istekleri verileri toplanırken 2.Öğretim (gece) derslerinin 16.00'da başlayabileceğini belirtmişlerdir. Diğer hocalar ise sadece 17.00'da başlamasını istemişlerdir. Böyle bir durumda 17.00'da başlamasını isteyen öğretim elemanın dersi 16.00'a atanırsa uygunluk değerine ceza puanı eklenmektedir. Gündüz için oluşturulan ders programı akşam içinde kullanılacağı için bu durumu engellemek için derslerinin 16.00'da başlamasını istemeyen öğretim elamanları için gündüz üç saatlik iki ders atanmamalıdır.

Öğretim elemanları istekleri alınırken öğretim elemanlarından dersinin olmasını istemediği zaman dilimleri belirtmesi istenilmiştir. Sonrasında bu istenilmeyen zaman dilimleri bir excel tablosuna aktarılmıştır. Bu kısıt ile öğretim elemanlarının istemedikleri zaman dilimlerine dersleri atanmış ise uygunluk değerine ceza puanı eklenmektedir.

Öğretim elemanları istekleri alınırken öğretim elemanlarından günlük en fazla kaç saat dersi olması isteği de alınmıştır. Bu verilere göre belirtilen ders saatini geçen durumlar için uygunluk değerine ceza puanı eklenmektedir.

Öğretim elemanları istekleri alınırken bazı öğretim elemanları derslerinin peş peşe günlerde olmasını istemiştir. Bu isteği olan öğretim elemanlarının dersleri arasında gün farkı varsa yani bir dersi Salı diğer dersi Perşembe veya Cuma gibi bir durum olursa uygunluk değerine ceza puanı eklenmektedir.

4.3. Başlangıç Popülasyonun Oluşturulması

Sezgisel algoritmalar yardımıyla ders çizelgeleme problemini optimize etmenin ilk aşaması bir başlangıç popülasyonu oluşturulmasıdır. Bu popülasyon üzerinde çeşitli operatörler uygulanarak kısıtları sağlayan yeni çözümler oluşturulmaya çalışılır. Literatürde başlangıç popülasyonunun oluşturulması ile ilgili olarak daha önceden kullanılmış olan uygun bir çözümün popülasyona eklenmesi veya popülasyonun herhangi bir kısıtı sağlayacak şekilde oluşturulması gibi yöntemler kullanılmıştır. Çalışmamızda yapılan deneyler sonucunda başlangıç popülasyonumuzu, bazı kısıtları sağlayacak şekilde oluşturduk. Dersleri kodlama yapısına atarken dersin istenilen dersliğe yani kodlama yapısı içerisinde istenilen satıra atılmasını, öğretim elemanlarının derslerinin istedikleri günlere atılmasını yani kodlama yapısı içerisinde istenilen sütuna atılmasını ve uzaktan eğitim dersleri için ayrılan zaman dilimlerine 1. Sınıf ve 2. Sınıf derslerinin atılmamasını sağlayarak başlangıç popülasyonu için bu üç kısıtı aynı anda sağlayacak bir kod yapısı geliştirilmiştir. Bu şekilde oluşturulan başlangıç popülasyonu uygun çözüme ulaşmada ve problemin hızlı şekilde çözülmesinde programa yardımcı olmuştur.

4.4. Seçilim Operatörünün Uygulanması

Uygunluk fonksiyonu ile popülasyondaki bireylerin uygunluk değeri hesaplandıktan sonra bu değerlere göre bir seçim işlemi yapılır. Seçim işleminde iyi çözümlerin kötü çözümlere göre hayatta kalma olasılıkları daha yüksek olur. Seçim operatörü çözüm süreci boyunca daha kaliteli bireylerin oluşmasını sağlar. Bu operatör, yüksek uygunluk değerine sahip çözümleri tercih ederek algoritmaları optimum çözüme yönlendirir. İkinci bölümde temel seçim yöntemlerinden ayrıntılı olarak bahsetmiştik. Bu çalışmada Turnuva Seçim Yöntemi ve Rulet Tekerleği Seçim Yöntemi kullanılarak farklı parametre değerleri ile denemeler yapılmıştır.

4.5. Çaprazlama Operatörünün Uygulanması

Bireyler popülasyondan seçildikten sonra seçilen bireyler arasında çaprazlama işlemi uygulanır. Çaprazlama işleminin genel tanımı, rastgele seçilen iki birey arasında bilgi alışverişi yaparak yeni birey veya bireyler üretme sürecidir. Bu bilgi alışverişi ile kaliteli bireylerin oluşması amaçlanır. Literatürde ders çizelgeleme problemi için çeşitli çaprazlama operatörleri kullanılmıştır. Bu çalışmada farklı çaprazlama operatörleri üzerinde denemeler yapılmıştır. Denemeler sonucunda kodlama yapımıza en uygun, en iyi değerleri veren çaprazlama operatörünün Goldberg tarafından önerilen Kısmen Uyumlu Çaprazlama yöntemi olduğu görülmüştür. Bu çaprazlama yönteminde rastgele iki nokta belirlenerek bu noktalar arasında kalan genler bireyler arasında yer değiştirilir. Boş kalan genlerin yerleri diğer bireyin gen konumlarına bakılarak doldurulur. Bu çalışmada rastgele belirlenen tek noktadan çaprazlama yapılarak diğer işlemler uygulanmıştır ve iyi sonuçlar elde edildiği gözlemlenmiştir.

4.6. Tamir Fonksiyonunun Uygulanması

Çaprazlama operatörünün uygulanmasından sonrasında bireylerin yapısında bozulmalar yaşanmaktadır. Bazı ders numaralarının kaybolması, bazı ders numaralarının ise birden fazla atanması durumu olmuştur. Bu durumun tamiri için uygun bir kod yapısı geliştirilmiştir. Bu kod yapısı kaybolan ders numaralarını tespit ederek uygun yerlere atamaktadır. Ayrıca çaprazlama sonrası çözümün yapısında

birden fazla olan ders numaralarını tespit ederek birini bırakıp diğerlerinin yerine 0 değeri atamaktadır.

4.7. Mutasyon Operatörünün Uygulanması

Mutasyon, çaprazlama işleminden sonra bireylere ayrı ayrı uygulanır. Bu operatörün kullanılmasının sebebi popülasyon içerisinde çeşitliliği korumaktır. Eğer sadece çaprazlama operatörü kullanılsa popülasyon hızlı bir şekilde benzer bireylerden oluşacaktır. Bu hızlı yakınsama probleminden kurtulmak için mutasyon operatörü ile yeni çözümler üreterek popülasyonun çeşitliliği sağlanır. Mutasyon işlemi uygulanırken iyi çözüm performansı sağlamak için mutasyon oranına dikkat edilmelidir. Mutasyon işlemi rasgele seçilen çözüm içerisinde rastgele iki noktanın konumları yer değiştirerek uygulanır. Bu çalışmada mutasyon işlemi uygulanırken yarı akıllı bir mutasyon operatörü kullanılmıştır. Kodlama yapısı dikkate alınarak ilk nokta rastgele seçilmiş ikinci nokta ise ilk noktayla aynı satırdan (aynı derslikten) olacak şekilde seçilerek çözüme daha hızlı yakınsama amaçlanmıştır. Şekil 18’de mutasyon operatörünün uygulanma şekli gösterilmiştir.

Mutasyon Öncesi

14	0	193	0	37	44	211	12	35	2
----	---	-----	---	----	----	-----	----	----	---



Mutasyon sonrası

14	0	35	0	37	44	211	12	193	2
----	---	----	---	----	----	-----	----	-----	---

Şekil 18. Mutasyon Operatörünün Uygulanması

Çalışmamızda GA'nın kodlama tipi olarak Değer Kodlama kullanılmıştır. Bu kodlama tipine uygun olarak da kromozom yapısı oluşturulmuş ve uygun başlangıç popülasyonu, seçim operatörü, çaprazlama operatörü ve mutasyon operatörü belirlenmiştir. Çalışmamızda oluşturulan kromozom yapılarının her biri bir ders programını temsil etmektedir. GA problemin çözümünde kromozomlardan oluşan bir başlangıç popülasyonu oluşturmaktadır. Algoritma başlamadan önce belirlenen popülasyon sayısı kadar başlangıç popülasyonunun kromozom sayısı olur ve

algoritma çözüm arama sürecini bitirene kadar bu sayı sabit kalmaktadır. Başlangıç popülasyonu oluşturulduktan sonra çözüm arama süreci sonlanana kadar GA yöntemi ile kromozomlar üzerine uygunluk fonksiyonu ve 3 temel GA operatörü uygulanmıştır. Uygunluk fonksiyonu ile kromozomların yani ders programlarının her iterasyonda fakülte için ne kadar uygun oldukları değerlendirilmiş ve yaptığı kısıt ihlallerine göre ceza puanı verilmiştir. Bu değerlendirmeden sonra seçim operatörü ile hangi kromozomlara çaprazlama ve mutasyon operatörü uygulanacağına karar verilmiştir. Son olarak önceden belirlenen oranlara göre seçim operatörü ile seçilen kromozomlara, çaprazlama ve mutasyon operatörleri uygulanmıştır. Bu süreç optimal çözüme, belirlenen iterasyona veya durdurma kriterine ulaşıncaya kadar tekrarlanmıştır.

Ders programı çizelgelemede öğretim elemanları, dersler, şubeler, derslikler ve zaman periyotları olmak üzere beş ana faktör vardır. Bu beş faktörün kombinasyonu PSO yönteminde parçacık pozisyonu olarak tanımlanır ve her parçacık bir çözümü temsil eder. Amaç, optimal parçacık pozisyonunu (optimal ders programı çizelgesi) elde etmektir.

PSO yönteminde popülasyonda bulunan parçacıkların her biri bir pozisyona sahiptir. Bu pozisyonlar parçacığın problemin çözümüne ne kadar uygun olduğunu göstermektedir. Parçacığın pozisyonu her iterasyonda güncellenmektedir. PSO algoritması her parçacığın mevcut iterasyona kadar olan en iyi pozisyonunu (lbest) ve popülasyondaki tüm parçacıkların mevcut iterasyona kadar olan en iyi pozisyonunu (gbest) saklamaktadır. Her parçacığın pozisyonunun güncellenmesi için öncelikle lbest ve gbest değerleri ile bir parçacığın hız değeri hesaplanmaktadır. Sonrasında hesaplanan bu hız değeri ile parçacığın çözüm uzayındaki yeni pozisyonu belirlenmektedir. Çalışmamızda parçacık pozisyonlarının güncellenmesi çaprazlama ve mutasyon operatörleri ile yapılmıştır. Popülasyon içerisindeki bütün parçacıkların lbest'i ile popülasyonun gbest'i arasında rastgele belirlenen bir noktadan çaprazlama işlemi uygulanmıştır. Popülasyonun hızlı yakınsamasını engellemek için çaprazlama sonrası elde edilen parçacıklara mutasyon işlemi uygulanmıştır.

DÇP'nin çözümü için kullandığımız üçüncü yöntem bal arılarının nektar arayışlarında gösterdikleri işbirliğinden ilham alınarak geliştirilen ABC

algoritmasıdır. Amaç, en iyi nektarı (fakülte için en uygun ders programını) bulmaktır.

Standart bir ABC algoritması, kullandığımız GA ve PSO yöntemlerinde olduğu gibi başlangıç popülasyonu oluşturarak problemin çözümüne başlamakta ve uygunluk fonksiyonu ile çözümlerin uygunluğunu değerlendirmektedir. Aynı zamanda ABC algoritması, problemin çözümü için sıkı bir işbirliği içinde olan üç arı grubunu (işçi arı, izleyici arı ve kâşif arı) kullanarak probleme çözüm aramaktadır.

Algoritmamızın işçi arı aşamasında, popülasyondaki her çözüme mutasyon işlemi uygulanmıştır. Mutasyon işlemi başka çözümler üretmek için ders çizelgesi üzerinde seçilen iki dersin yer değiştirilmesidir. Bu mutasyon işleminin aynısı GA ve PSO yöntemlerinde de kullanılmıştır. Mutasyon işlemi sonrasında elde edilen yeni çözümün uygunluk değeri önceki çözümden daha iyi ise yeni çözüm hafızada tutulur, eğer çözüm değerinde iyileşme gerçekleşmezse eski çözüm ile devam edilir. Aynı zamanda her çözüm iyileştirilememe sayısı olarak ifade edilen bir sayaca sahiptir. Başlangıç popülasyonunda her çözüme iyileştirilememe sayısı olarak 0 değeri atanır. Eğer mutasyon işlemi sonrası çözümün uygunluk değerinde iyileşme yaşanmazsa iyileştirilememe sayısı bir artırılır. Ne zaman çözümün uygunluk değerinde bir iyileşme yaşanırsa o zaman çözümün sayacı sıfırlanır.

İzleyici arı aşamasında, işçi arı aşamasında olduğu gibi popülasyon içerisindeki çözümlere mutasyon işlemi uygulanmıştır. Mutasyon işleminin uygulanacağı çözümlerin belirlenmesi Rulet Tekerleği Seçilim Yöntemi ile yapılmıştır. Bu seçim tekniği, çözümlerin uygunluk değerine göre seçilme ihtimalini belirlemektedir. Çözümün kalitesi arttıkça mutasyon işlemi için seçilme olasılığı da artmaktadır. Mutasyon işleminden sonra işçi arı aşamasında olduğu gibi çözümün uygunluğunda bir iyileşme olursa çözümün sayacı sıfırlanır aksi durumda çözümün iyileştirilememe sayısı bir artırılır.

ABC algoritmasının parametrelerinden biri limit parametresidir. Bu parametre algoritma çalışmaya başlamadan önce belirlenmektedir. Bir çözümün iyileştirilememe değeri belirlenen limit değerinden büyükse kâşif arı aşaması başlar. Bu aşama iyileştirilememe ve limit değerlerini dikkate alarak hafızada tutulan çözümün silinmesine ve yeni çözümün oluşturulmasına karar vermektedir. Bu

ařamadan sonra popülasyonun en uygun çözümlü hafızaya alınır. İterasyon sayısına veya durdurma kriterine ulařılmadıysa algoritma işçi arı aşamasına geri dönmektedir.



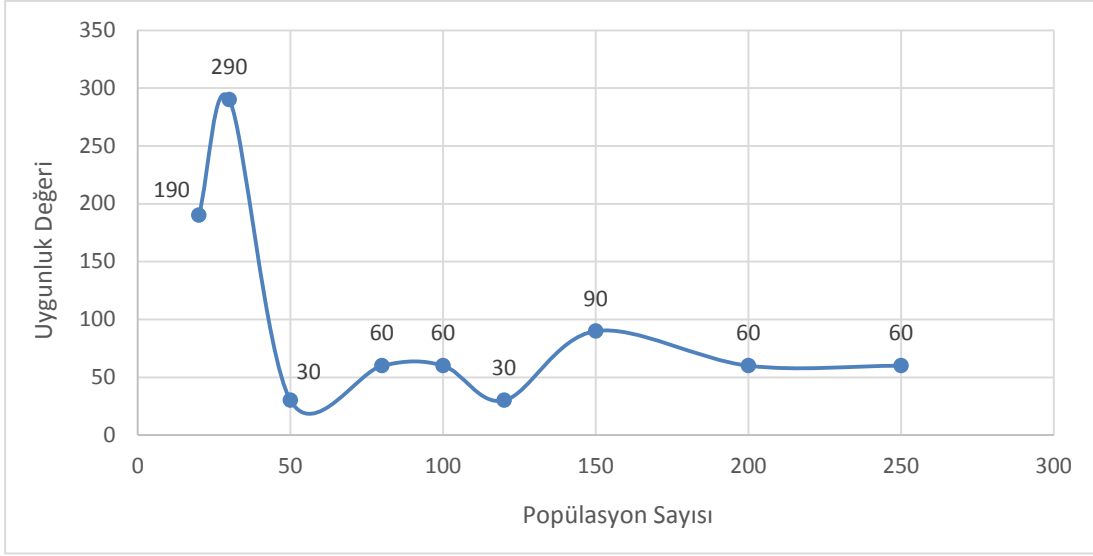
V.BÖLÜM: SONUÇ VE ÖNERİLER

Bu çalışmada deneysel sonuçlar, Intel (R) Core (TM) i5-4200U, 1.60-2.30 GHz işlemciye ve 4 GB RAM'e sahip dizüstü bilgisayar kullanılarak elde edilmiştir. Yöntemlerin probleme uygulanması Matlab programında gerçekleştirilmiştir. Uygulama için hazırlanan yazılımın çözüm algoritmasında standart GA, PSO ve ABC yöntemleri kullanılmıştır. DÇP'nin çözümü için hangi yöntemin daha iyi performans gösterdiğini görmek için bu üç sezgisel yöntem karşılaştırılmıştır.

Tüm algoritmaların düzgün çalışması için ayarlanacak algoritma parametrelerine ihtiyacı vardır. Bu algoritmalar ile optimum çözümün aranması noktasında uygun parametrelerin seçimi oldukça önemlidir. Algoritma parametrelerindeki bir değişiklik, algoritmanın etkinliğini değiştirir. Bu çalışmada tüm parametreler deneysel olarak belirlenmiş ve parametreler üzerinde mümkün olduğu kadar kombinasyon yapılmaya çalışılmıştır. Üç yöntem içinde yeteri kadar deney yapılmış sadece uygun çözüme ulaşan sonuçlar aşağıdaki deney tablolarında verilmiştir.

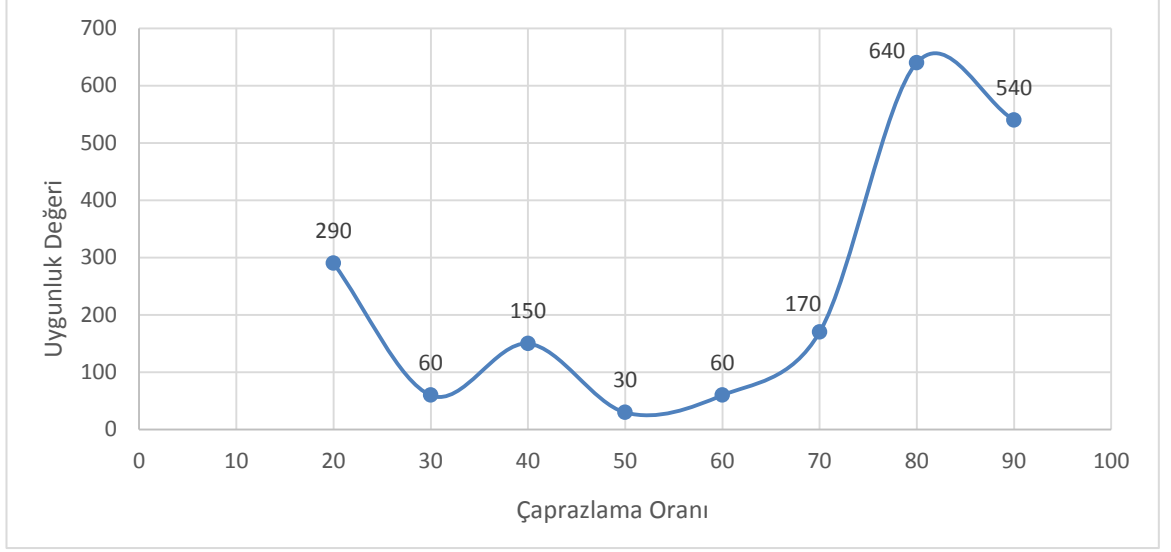
GA yöntemi kullanılarak yapılan deneyler için Rulet Tekerleği Seçilim Yöntemi ve Turnuva Seçilim Yöntemi, Kısmen Uyumlu Çaprazlama(PMX) yöntemi ve kromozom yapısına göre belirlenmiş olan bir mutasyon operatörü kullanılmaya karar verilmiştir. GA operatörleri ve parametreleri belirli değer aralıklarında değiştirilerek uygun çözüm değerlerine ulaşılmaya çalışılmıştır.

Bu çalışmada popülasyon sayısının 50-100 arasında iyi sonuçlar verdiği 50'nin altına inildikçe uygunluk değerinin kötüleştiği ve bütün zorunlu kısıtların sağlanmadığı görülmüş, 100'ün üstüne çıkıldıkça uygunluk değerinde iyileşme yaşanmadığı görülmüştür. Bütün parametreler sabit tutularak farklı popülasyon sayıları ile yapılan deneylerin uygunluk değeri sonuçları Şekil 19'da gösterilmiştir.

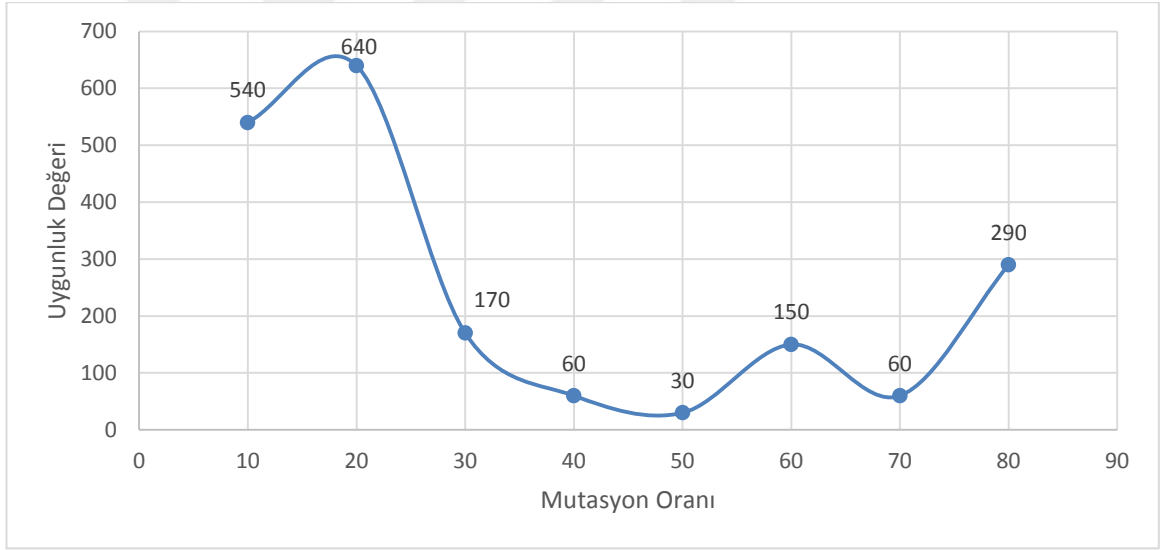


Şekil 19. GA yönteminde farklı popülasyon sayılarına göre uygunluk değeri sonuçları

Farklı parametre değerleri ile çaprazlama oranının %30-%80 arasında, mutasyon oranının ise %20-%70 arasında iyi sonuçlar verdiği görülmüştür. Fakat 50 popülasyon sayısı ve 1000 iterasyon sayısı sabit tutularak, çaprazlama oranı ve mutasyon oranı üzerine yapılan deneyler sonucunda; çaprazlama oranının %30-%60 arasında iyi sonuçlar verdiği %30'un altına inildiğinde veya %60'ın üzerine çıktığında uygunluk değerinin kötüleştiği görülmüş, mutasyon oranının ise %40-%70 arasında iyi sonuçlar verdiği %40'ın altına inildikçe veya %70'in üstüne çıktıkça çözümün uygunluktan uzaklaştığı görülmüştür. Çaprazlama ve mutasyon oranlarının uygunluk değeri üzerine etkileri Şekil 20 ve Şekil 21'de gösterilmiştir.

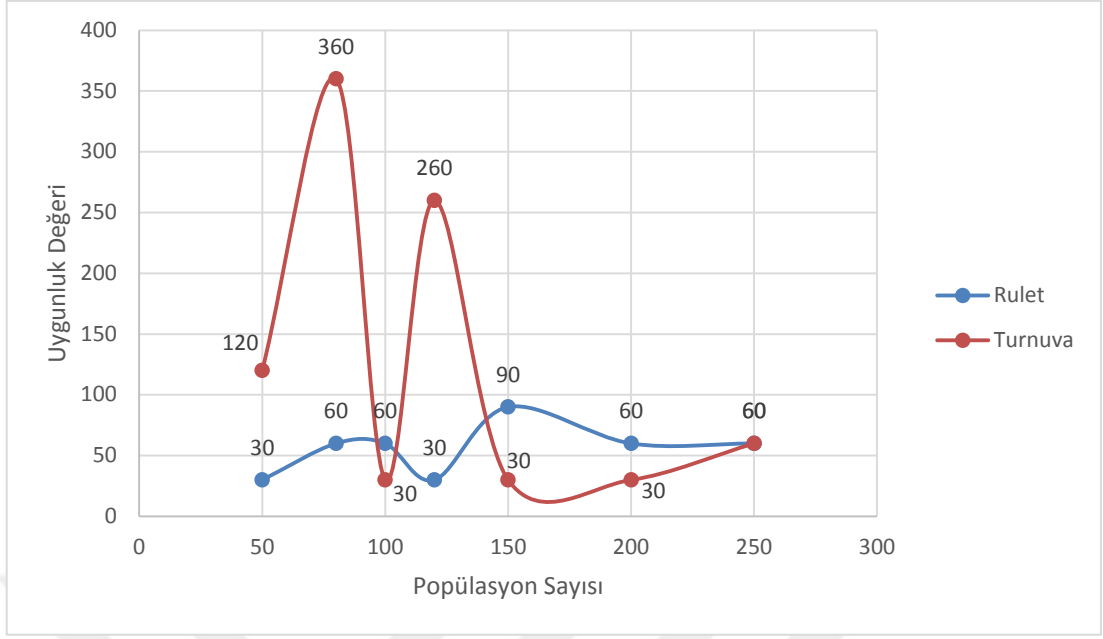


Şekil 20. GA yönteminin çaprazlama oranına göre uygunluk değeri dağılımı



Şekil 21. GA yönteminin mutasyon oranına göre uygunluk değeri dağılımı

Seçilim operatörü olarak Rulet Tekerleği Yöntemi ile yapılan deneyler Turnuva Seçilim Yöntemine göre daha iyi sonuçlar verdiği gözlenmiştir. Tüm parametreler sabit tutularak seçilim yöntemlerine bakıldığında, genellikle Rulet Tekerleği Yöntemi ile daha hızlı çözüme ulaşıldığı görülmüştür. Şekil 22’de diğer parametreler sabit tutularak farklı popülasyon sayıları ile Rulet Tekerleği ve Turnuva yöntemlerinin uygunluk değeri üzerine etkileri gösterilmiştir.



Şekil 22. Rulet Tekerleği ve Turnuva Yöntemlerinin Karşılaştırma Sonuçları

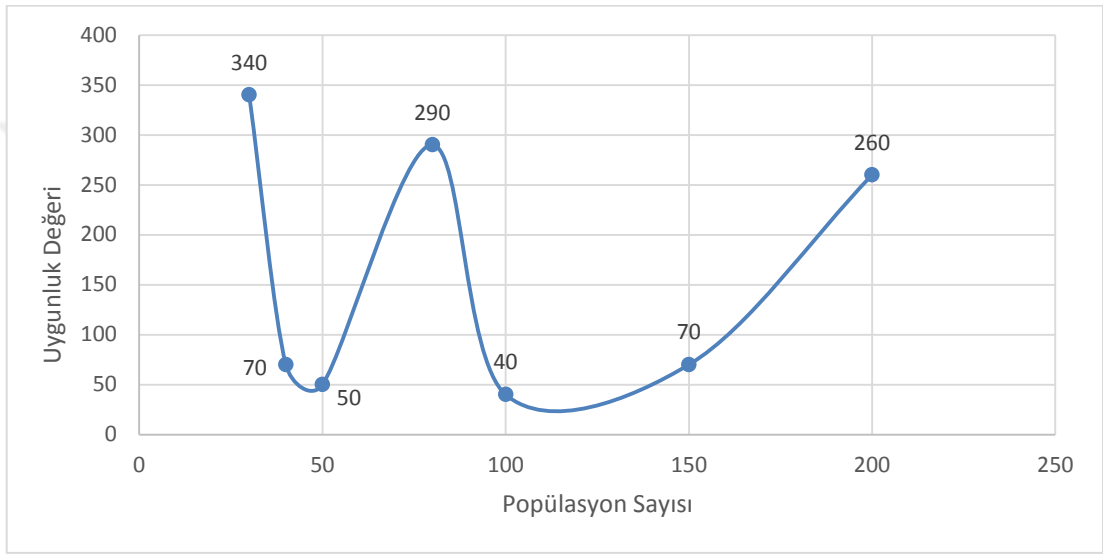
Yapılan deney sonucunda fakülte için uygun ders programları elde edilmiştir. En hızlı ve en uygun şekilde elde edilen ders programı parametre olarak 50 popülasyon sayısı, 1000 iterasyon sayısı, Rulet Tekerleği Seçilim Yöntemi, yaklaşık %50 çaprazlama oranı, %50 mutasyon oranı ve 2 elitizm sayısı ile elde edilmiştir. Deneyler için elitizm sayısının 2 olarak sabit tutulmasına karar verilmiştir. Çaprazlama oranına yaklaşık dememizin sebebi elitizm ile gelen bireylerin bu oranın içerisinde olmasıdır. Tablo 5’de GA yöntemi ile yapılan deneylere ait parametre ve sonuç değerleri bulunmaktadır.

Tablo 5. GA Yöntemine Ait Deney Tablosu

Popülasyon Sayısı	İterasyon Sayısı	Seçim Operatörü	Yaklaşık Çaprazlama Oranı(%)	Mutasyon Oranı(%)	Elitizm Sayısı	Uygunluk Değeri	Zaman (Dk)
20	1000	Rulet	40	60	2	90	4.9240
20	2000	Rulet	80	20	2	160	7.6189
20	3000	Rulet	60	40	2	120	12.1905
50	1000	Rulet	30	70	2	60	7.1777
50	1000	Rulet	40	60	2	150	7.4854
50	1000	Rulet	60	40	2	60	7.3652
50	1000	Rulet	50	50	2	30	8.0109
50	1000	Turnuva	60	40	2	60	7.2749
50	2000	Turnuva	50	50	2	120	14.1237
80	1000	Turnuva	40	60	2	60	11.1836
80	1000	Rulet	50	50	2	60	10.8797
100	1000	Turnuva	50	50	2	30	12.6741
100	1000	Rulet	40	60	2	30	12.9415
100	1000	Turnuva	30	70	2	60	13.7593
100	1000	Rulet	70	30	2	60	12.9866
100	1000	Rulet	50	50	2	60	12.5832
100	2000	Turnuva	70	30	2	80	27.6699
100	1000	Rulet	30	70	2	30	13.7162
100	2000	Turnuva	80	20	2	40	26.0067
120	1000	Rulet	50	50	2	30	14.7453
150	1000	Turnuva	50	50	2	30	18.9699
150	1000	Rulet	50	50	2	90	20.7375
200	1000	Turnuva	40	60	2	60	25.4325
200	1000	Rulet	50	50	2	60	24.0480
200	1000	Turnuva	50	50	2	30	25.1464
250	1000	Turnuva	50	50	2	60	29.9808
250	1000	Rulet	50	50	2	60	30.2645
500	1000	Rulet	50	50	2	90	57.4138
500	2000	Rulet	80	20	2	60	369.5084
500	1000	Turnuva	50	50	2	30	59.9206

PSO yöntemi oldukça sade bir algoritma yapısına sahip olmasına rağmen zor bir problem olan DÇP'nin çözümünde oldukça etkili olmuştur. PSO tabanlı algoritma ABC yöntemine kıyasla uygulanabilir bir çözüm için daha az zaman harcadığını GA yöntemine kıyasla daha fazla zaman harcadığını göstermiştir. Bununla birlikte

bulduğu en uygun ve en hızlı çözüm için harcadığı zamanın diğer iki yöntemden fazla olduğu görülmüştür. Algoritma 40 ve üzeri popülasyon sayısı ve yeterli sayılabilecek 1000 ve 2000 iterasyon sayıları ile uygun çözümler üretmiştir. Fakat 40'ın üstünde bazı popülasyon sayıları ile uygun olmayan sonuçlar bulunmuştur. Uygun olmayan sonuçlar şekillere eklenmiş, çözüm tablosuna eklenmemiştir. PSO yönteminde farklı popülasyon sayıları ile uygunluk değerinin değişimi Şekil 24'de gösterilmiştir. Ayrıca PSO yöntemi kullanılarak yapılan deneyler sonucunda elde edilen uygun çözümler Tablo 7'de spesifik olarak verilmiştir.

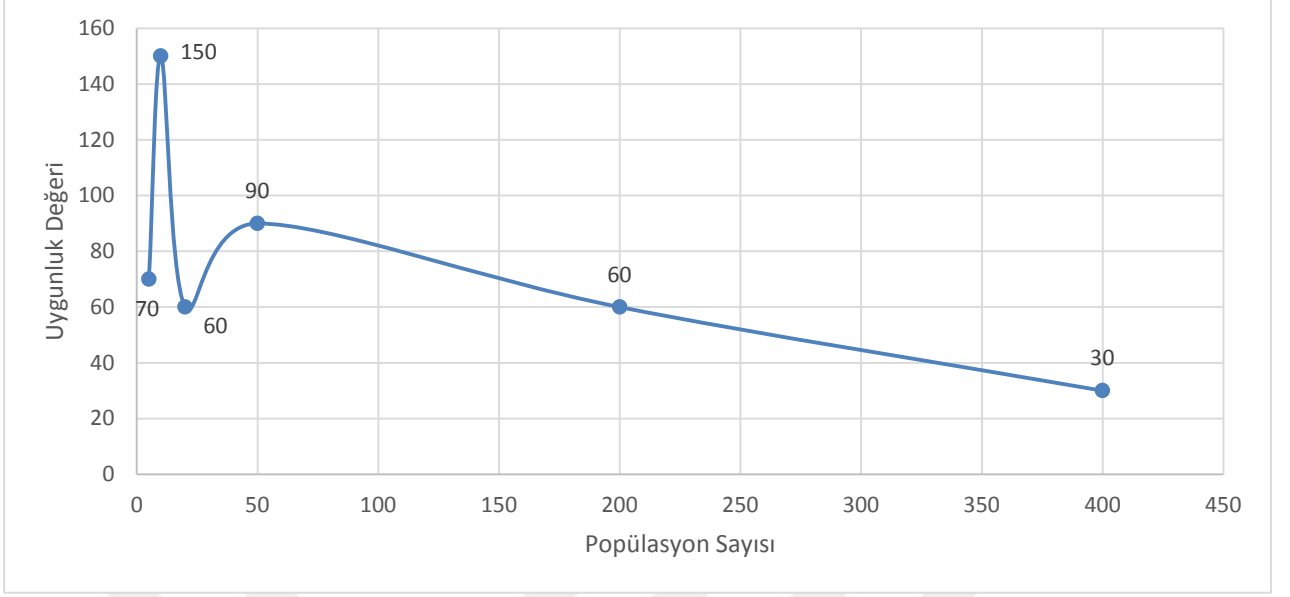


Şekil 23. PSO yönteminin popülasyon sayısına göre uygunluk değeri dağılımı

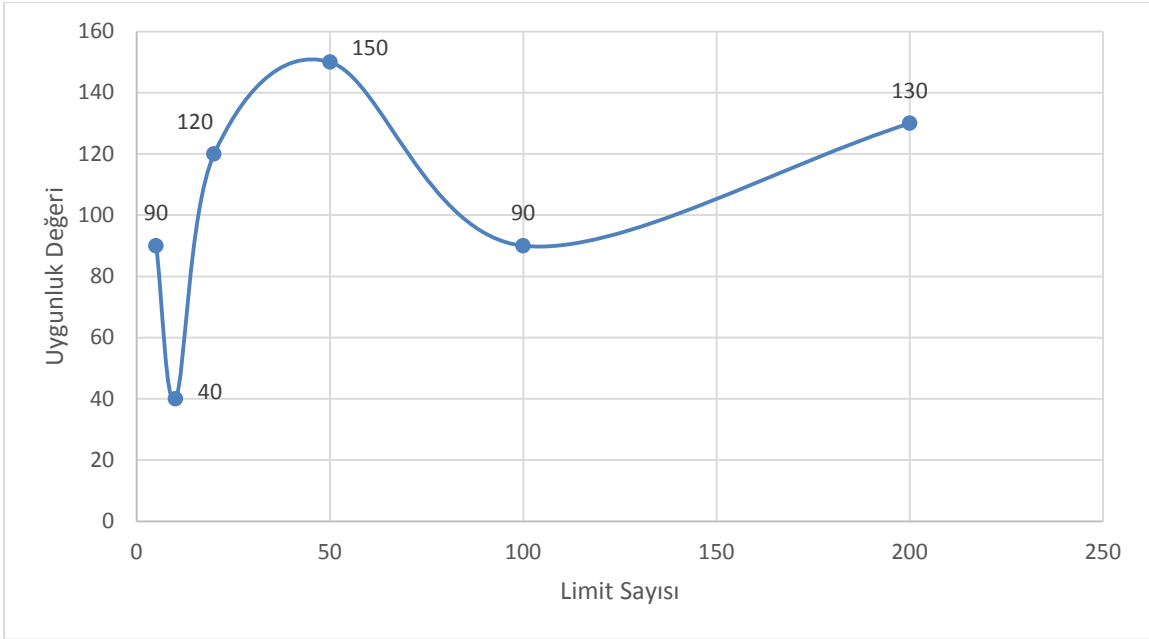
Tablo 6. PSO Yöntemine Ait Deney Tablosu

Popülasyon Sayısı	İterasyon Sayısı	Uygunluk Değeri	Zaman (Dk)
40	2000	60	14.7385
40	1000	70	7.6933
50	1000	50	9.2392
50	2000	30	17.9886
100	1000	40	17.7989
100	2000	100	35.4299
150	1000	70	26.5973
150	500	40	13.4041
300	2000	60	107.7465
500	2000	30	189.8092
750	2000	30	277.9001

ABC algoritmasının performansını değerlendirmek için farklı popülasyon sayısı, iterasyon sayısı ve limit sayısı ile deneyler yapılmıştır. İterasyon sayısı diğer yöntemlerde olduğu gibi genellikle 1000 veya 2000 olarak sabit tutulmuştur. 2000 iterasyon sayısı ve düşük popülasyon sayısı ile hızlı ve başarılı sonuçlar elde edilmiştir. Popülasyon sayısı olarak 5 ve 20 arasında ABC algoritması ile uygun ve hızlı çözümler elde edilmiştir. Bu çalışmada limit sayına 5 ve 500 arasında değerler verilmiştir. Limit sayısının düşük olması problem için uygun çözümün bulunmasını zorlaştırmakta ve problemin çözüm süresini artırmakta, limit sayısının fazla olması durumunda ise algoritmanın daha kaliteli bir çözüm bulma noktasında sorun yaşadığı görülmüştür. Bu çalışmada 10 limit sayısı ile hızlı ve uygun çözümler elde edilmiştir. ABC yönteminde iterasyon sayısı 1000 ve limit değeri 50 olarak sabit tutulmuş, 6 deney ile popülasyon sayısının uygunluk değeri üzerine etkisi gözlemlenmiştir. Bu sonuçlara göre, 0-50 arasındaki popülasyon sayıları ile uygun çözümlere uygun çözümlere ulaşılmıştır. Şekil 24'te bu deneylerin sonuçları verilmiştir. ABC'nin diğer bir parametresi olan limit sayısı üzerine yapılan deneylerde 10 popülasyon sayısı ve 1000 iterasyon sayısı sabit tutulmuştur. Yapılan deneylerin sonuçları Şekil 25'te verilmiştir. Problemin çözümünde kullanılan diğer iki yöntem ile kıyaslandığında ABC algoritmasının daha başarılı olduğu görülmüştür. Tablo 7'de ABC algoritması ile yapılan 22 deneyin parametre ve sonuçları paylaşılmıştır.



Şekil 24. ABC yönteminde farklı popülasyon sayılarına göre uygunluk değeri sonuçları



Şekil 25. ABC yönteminde farklı limit sayılarına göre uygunluk değeri sonuçları

Tablo 7. ABC Yöntemine Ait Deney Tablosu

Popülasyon Sayısı	İterasyon Sayısı	Limit Sayısı	Uygunluk Değeri	Zaman (Dk)
5	2000	10	30	5.8840
5	2000	100	30	5.7062
5	1000	50	70	2.8968
10	1000	5	90	7.5448
10	1000	10	40	6.7891
10	1000	20	120	6.8564
10	1000	50	150	7.0625
10	1000	100	90	5.8318
10	1000	200	130	5.3061
10	2000	100	30	11.9237
10	3000	10	30	21.4441
20	1000	10	90	11.8228
20	2000	10	30	27.9280
20	1000	50	60	11.3658
50	1000	10	60	28.9996
50	1000	50	90	28.3071
50	1000	100	50	27.7390
100	1000	10	30	62.4398
100	1000	50	30	58.7479
200	1000	50	60	114.8632
400	1000	50	30	224.2152
500	1000	100	30	299.3666

Önerilen üç yaklaşımın performansları elde edilen en iyi ve en hızlı çözüme göre karşılaştırıldı. Üç yöntemde makul sürede oldukça başarılı sonuçlar elde etmiştir. Ancak üç yöntem ile elde edilen en iyi ve en hızlı çözümlere göre ABC yöntemi diğer iki yöntemle göre daha hızlı çözüme ulaşmıştır. Söz konusu üç deney Tablo 10’da verilmiştir. Bunun yanı sıra 50 popülasyon sayısı ve 1000 iterasyon sayısı ile algoritmaların en hızlı çözümleri kıyaslanmıştır. Elde edilen bulgulara göre ABC algoritmasının diğer iki yöntemle göre daha yavaş çalıştığı gözlemlenmiştir. GA’nın diğer iki yöntemden daha hızlı problemin çözümünü sonlandırdığı görülmüştür. Elbette bu çözüm süreleri algoritmaların diğer parametreleri ile doğrudan ilişkilidir. Örneğin; ABC algoritmasının limit değeri veya GA’nın çaprazlama oranı çözüm süresini etkiler. Çalışmamızda her yöntem farklı parametre değerleri ile test edilmiş, uygun çözüme ulaşan parametre değerleri tablolara

eklenmiş ve bu üç yöntemin karşılaştırılması da tablolardaki çözümler üzerinden yapılmıştır. Tablo 8’de üç yöntemin kıyaslanan çözümleri verilmiştir.

Tablo 8. GA, PSO ve ABC Yöntemlerini Kıyaslama Tablosu

Yöntem	Popülasyon Sayısı	İterasyon Sayısı	Uygunluk Değeri	Zaman (Dk)
GA	50	1000	30	8.0109
GA	50	1000	60	7.1777
PSO	50	2000	30	17.9886
PSO	50	1000	50	9.2392
ABC	5	2000	30	5.7062
ABC	50	1000	50	27.7390

Bu tez çalışması DÇP’nin çözümü için farklı algoritma türlerinin uygulanmasını tartışmıştır. Deneysel sonuçlar önerilen üç algoritmanın da, belirlenen çeşitli kısıtlara göre öğretim elamanları ve sınıfların gereksinimlerini karşılayan tatmin edici bir ders programı oluşturabileceğini göstermiştir. Son olarak elde edilen en iyi ders programlarından bir tanesi kullanılarak bütün bölümlerin 1. öğretim ve 2. öğretim ders programı tabloları hazırlanmıştır. Bu tablolardan Yönetim Bilişim Sistemleri Bölümü’nün ders programı tablosu ekte paylaşılmıştır.

Bu tez çalışmasında, Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Fakültesi ders programının hazırlanması için kullanılan gerçek veriler üzerinde çalışma yapılmıştır. Kullanılan bu veriler Excel programına kod yapısı düşünülerek aktarılmıştır. Bu Excel dosyasında 220 tane ders, 91 tane öğretim elemanı ve 35 tane şube hakkında bilgiler yer almaktadır. Excel dosyası ve hazırlanan algoritmalar oldukça kullanışlı ve esnek bir yapıya sahiptir. GA, PSO ve ABC yöntemleri kullanılarak geliştirilen algoritmalar farklı fakültelerin ders programlarının hazırlanmasında rahatlıkla kullanılabilir şekilde hazırlanmıştır. Excel dosyası içerisindeki veriler değiştirilse bile program yeni veriler üzerinde rahatlıkla çalışacaktır. Ayrıca Matlab programında hazırlanan algoritmaların parametre değerleri istenildiği gibi ayarlanarak kullanılabilir.

Literatürde kullanılan klasik yöntemlerin çözülmesi zor olan çizelgeleme problemlerinin optimizasyonu aşamasında yetersiz kaldığı görülmektedir. Bu tez çalışmasında sezgisel yöntemler olan GA, PSO ve ABC’nin farklı parametreleri

kullanılarak yapılan deneyler sonucunda zor bir optimizasyon problemi olan DÇP'nin uygun çözümü sunulmuş ve söz konusu yöntemler kıyaslanmıştır. Bu çalışmada yöntemler düzgün bir şekilde kullanarak zor bir problem olan DÇP'nin çözümünde iyi sonuçlar elde edilmiştir. Böylelikle hem bizim fakültemizin hem de farklı fakültelerin kullanabileceği ve bütün bölümlerin ders programını kısa sürede ve tek tabloda hazırlayan algoritmalar geliştirilmiştir. Öğretim elemanları ve fakültenin bütün istekleri dikkate alınarak hazırlanan bu ders programı, lisansüstü programların ve idari görevlerin ayarlanması konusunda da fakülteye kolaylık sağlayacaktır.

İleriki çalışmalarda, üç yöntemin de algoritma yapısında kullandığımız algoritmanın performansı açısından oldukça önemli olan genetik operatörlerin iyileştirilmesine odaklanarak, daha kaliteli ve hızlı çözüm veren algoritmaların geliştirilmesi düşünülmektedir. Bunun yanı sıra bu yapının kullanıcı için faydalı olabilecek bir arayüz çalışmasıyla desteklenmesi düşünülmektedir.

Bu tez çalışmasının literatür kısmında bahsedildiği üzere farklı problemlerin çözümünde sezgisel yöntemler ve sezgisel yöntemlerin birlikte kullanımıyla oluşturulan hibrit yaklaşımlar kullanılmıştır. Gelecekteki çalışmalarda GA, PSO ve ABC yaklaşımları ile birlikte yardımcı yöntemler kullanılarak problemlerin çözüm sürelerini kısaltacak algoritmaların elde edilmesi hedeflenmektedir. Ayrıca daha sonraki yapılacak çalışmalarda, kullandığımız üç yöntemin dışında ACO, TA, Tavla Benzetimi, Diferansiyel Gelişim gibi farklı yöntemlerle hiyerarşik ve hibrit yapılar ele alınarak problemin çözümünde daha etkin neticelerin elde edilmesi gelecek çalışmalara bırakılmıştır.

Gelecekte yapılması planlanan diğer bir çalışma ise, geliştirilmesi düşünülen yeni algoritmalar ve bu çalışmada kullandığımız algoritmalar kullanılarak farklı sektörlerdeki optimizasyon problemlerinin çözümü üzerine olacaktır.



KAYNAKLAR

- ABDELHALİM Esraa A., EL KHAYAT Ghada A. (2016). “A Utilization-Based Genetic Algorithm For Solving The University Timetabling Problem (Uga).” *Alexandria Engineering Journal*, 55(2), 1395-1409.
- ABDULLAH Salwani, TURABİEH Hamza (2008). “Generating University Course Timetable Using Genetic Algorithms And Local Search. In *Convergence And Hybrid.*” *Information Technology, 2008. ICCIT'08. Third International Conference On* (Vol. 1, Pp. 254-260). IEEE.
- ABİDİN Didem (2013). “Curriculum Plan Optimization With Rule Based Genetic Algorithms.” Diss. Doktora Tezi, DEÜ, Fen Bilimleri Enstitüsü.
- AKAY Bahriye, KARABOGA Derviş (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of intelligent manufacturing*, 23(4), 1001-1014.
- AKKAN Can, GÜLCÜ Ayla (2018). “A Bi-Criteria Hybrid Genetic Algorithm With Robustness Objective For The Course Timetabling Problem.” *Computers & Operations Research*, 90, 22-32.
- ALKAN Alpay (2002). “A Solution To Course Scheduling Problem Using Genetic Algorithms.” Yüksek Lisans Tezi, Yeditepe Üniversitesi Bilim ve Mühendislik Enstitüsü.
- ALTAMİRANO Leopoldo vd. (2012). Anesthesiology nurse scheduling using particle swarm optimization. *International Journal of Computational Intelligence Systems*, 5(1), 111-125.
- ALVAREZ Ramon, CRESPO Enric, TAMARİT Jose M. (2002) “Design and implementation of a course scheduling system using tabu search”. *European Journal of Operational Research*, 137(3), 512-523.
- ALZAQEBAH Malek, ABDULLAH Salwani (2011). Artificial bee colony search algorithm for examination timetabling problems. *International Journal of Physical Sciences*, 6(17), 4264-4272.

- ALZAQEBAH Malek, ABDULLAH Salwani (2011). Comparison on the selection strategies in the artificial bee colony algorithm for examination timetabling problems. *Int J Soft Comput Eng*, 1(5), 158-163.
- AWADALLAH Mohammed A., BOLAJİ Asaju La'aro, AL-BETAR Mohammed A. (2015). A hybrid artificial bee colony for a nurse rostering problem. *Applied Soft Computing*, 35, 726-739.
- AYDIN M. Aslı (2008). "Solving University Course Timetabling Problem Using Genetic Algorithm." Yüksek Lisans Tezi, Bahçeşehir Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği.
- AYDİN Cansu Çiğdem, CAGİLTAY Nergiz Ercil, BİROGUL Serdar (2008). "A Genetic Algorithm Based Course Scheduling System." Ulusal Bilisim Kurultayı (BILISIM'08).
- BAĞIŞ Aytekin (1996). "Genetik Algoritma Kullanılarak Ders Programının Optimum Şekilde Düzenlenmesi." Yüksek Lisans Tezi, Erciyes Üniversitesi Fen Bilimler Enstitüsü.
- BAYSAL Ceyda (2011). "Evrimsel Hesaplama Tekniği Kullanılarak Otomatik Sınav Programı Oluşturma." Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı.
- BOLAJİ Asaju La'aro vd. (2011). An improved artificial bee colony for course timetabling. In *Bio-Inspired Computing: Theories and Applications (BIC-TA)*, 2011 Sixth International Conference on (pp. 9-14). IEEE.
- BURKE Edmund K. vd. (2007). "A Graph-Based Hyper-Heuristic For Educational Timetabling Problems. *European Journal Of Operational Research*." 176(1), 177-192.
- BURKE Edmund Kieran, PETROVIĆ Sanja (2002). "Recent Research Directions In Automated Timetabling. *European Journal Of Operational Research*." 140(2), 266-280.
- CHEN Ruey-Maw, SHİH Hsiao-Fang (2013). "Solving university course timetabling problems using constriction particle swarm optimization with local search". *Algorithms*, 6(2), 227-244.

- CHU Shu-Chuan, CHEN Yi-Tin, Ho Jiun-Huei (2006). Timetable scheduling using particle swarm optimization. In Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on (Vol. 3, pp. 324-327). IEEE.
- ÇAKAR Kemal (2009). "Genetik Algoritmalar Yardımıyla Acil Servis İstasyonu Yerleşiminin Optimizasyonu". Doktora Tezi, Atatürk Üniversitesi Sosyal Bilimler Enstitüsü İşletme(Sayısal Yöntemler) Anabilim Dalı.
- ÇİVRİL Hanife (2009). "Hemşire Çizelgeleme Probleminin Genetik Algoritma İle Çözümü". Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Elektronik Bilgisayar Eğitimi Anabilim Dalı.
- ÇOLAK Recep (2015). Sezgisel Algoritmalarla Ders Programı Çizelgeleme Problemi Çözümü. Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı.
- DABAN Fatma, ÖZDEMİR Ersin (2004). "Eğitimde Verimliliği Artıran Ders Programlarının Hazırlanması İçin Genetik Algoritma Kullanımı." Journal Of Educational Sciences & Practices 3.6.
- DATTA Dilip, DEB Kalyanmoy, FONSECA Carlos M. (2006). "Solving Class Timetabling Problem Of IIT Kanpur Using Multi-Objective Evolutionary Algorithm." *Kangal, Report, 2006006*, 1-10.
- DERİS Safaai vd. (1999). "Incorporating Constraint Propagation İn Genetic Algorithm For University Timetable Planning." Engineering Applications Of Artificial Intelligence 12.3: 241-253.
- DİLAVER Dilek (2015). "Genetik Algoritmalar Yardımıyla İş Atölye Çizelgelemesi Üzerine Bir Çalışma." Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Anabilim Dalı.
- EBERHART Russ, SIMPSON Pat, DOBBINS Roy (1996). "Computational Intelligent PC Tools", Boston, MA. (1996) Academic Press Professional.

- EBERHART Russell C., SHI Yuhui (1998). "Comparison between genetic algorithms and particle swarm optimization". In *Evolutionary Programming VII*, Porto, V.W., Saravanan, N., Waagen, D. and Eiben, A.E. (eds) 611–616 Springer.
- EBERHART Russell C., SHI Yuhui, KENNEDY James (2001). "Swarm intelligence", Morgan Kaufmann, San Mateo, CA.
- ELMAS Çetin (2007), *Yapay Zeka Uygulamaları*, Seçkin Yayıncılık, Ankara.
- EMEL Gül Gökay, TAŞKIN Çağatan (2002). Genetik Algoritmalar ve Uygulama Alanları. *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21(1), 129-152.
- ERBEN Wilhelm and KEPPLER Jürgen (1995). "A Genetic Algorithm Solving A Weekly Course-Timetabling Problem." *International Conference On The Practice And Theory Of Automated Timetabling*. Springer, Berlin, Heidelberg.
- FANG Hsiao-Lan (1994). "Genetic Algorithms In Timetabling And Scheduling." Diss. University Of Edinburgh.
- GHAEMI Sehraneh, VAKILI Mohammad Taghi, AGHAGOLZADEH Ali (2007). "Using A Genetic Algorithm Optimizer Tool To Solve University Timetable Scheduling Problem." In *Signal Processing And Its Applications, 2007. ISSPA 2007. 9th International Symposium On*(Pp. 1-4). IEEE.
- GOLDBERG David E. (1989) *Genetic Algorithm In Search, Optimization & Machine Learning*, Addison-Wesley Longman Publishing, Boston, MA, USA.
- GRAFENSTETTE John J. (1986). "Optimization Of Control Parameters For Genetic Algorithms." *IEEE Transactions On Systems, Man, And Cybernetics*, 16(1), 122-128.
- GÜNTHER Maik, NISSEN Volker (2010). Sub-daily staff scheduling for a logistics service provider. *Künstl Intell.*, 24:105-113.
- HAUPT Randy L., ELLEN HAUPT Sue (2004). *Practical Genetic Algorithms*. John Wiley & Sons.

- HERTZ Alain, (1991) Tabu search for large scale timetabling problems. Eur. J. Oper. Res., 54, 39–47.
- IRENE Sheau Fen Ho, DERIS Safaai, ZAITON Mohd Hashim Siti (2009). A study on PSO-based university course timetabling problem. In Advanced Computer Control, 2009. ICACC'09. International Conference on (pp. 648-651). IEEE.
- KALAYCI Can Berk (2008). “Öğrenci Başarısına Odaklı Sınav Çizelgeleme Modeli Ve Yazılım Uygulaması” Yüksek Lisans Tezi Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı.
- KARABOGA Dervis (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- KARABOĞA Derviş (2017). *Yapay Zeka Optimizasyon Algoritmaları*, Nobel Yayıncılık, Ankara.
- KENNEDY James, EBERHART Russell C. (1995). “Particle Swarm Optimization”, Proc. of the IEEE Int. Conference on Neural Networks, 4, 1942-1948.
- KÜÇÜK Ahsen (2016). “Hemşire Çizelgeleme Probleminin Genetik Algoritmalarla Optimizasyonu Ve Bir Uygulama.” Yüksek Lisans Tezi Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Anabilim Dalı, İzmir.
- KÜÇÜKSİLLE Ecir Uğur, TOKMAK Mahmut (2011). Yapay arı kolonisi algoritması kullanarak otomatik ders çizelgeleme. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 15(3).
- LIMLAWAN Varis, KASEMSONTITUM Boontariga, JEENANUNTA Chawalit (2011). Airline crew rostering problem using particle swarm optimization. In Quality and Reliability (ICQR), IEEE International Conference on (pp. 501-505). IEEE.
- LIMLAWAN Varis, KASEMSONTITUM Boontariga, JEENANUNTA Chawalit (2014). A hybrid particle swarm optimization and an improved heuristic algorithm for an airline crew rostering problem. Science Asia, 456-462.
- MATHEW Tom V. (2012). "Genetic Algorithm." Report Submitted At IIT Bombay.

- MAULIK Ujjwal, BANDYOPADHYAY Sanghamitra (2000). "Genetic Algorithm-Based Clustering Technique." *Pattern Recognition*, 33(9), 1455-1465.
- MİTCHELL Melanie (1998). *An Introduction To Genetic Algorithms*. MIT Press.
- ÖZCAN Mümin (2016). "Atölye Tipi Çizelgeleme Problemlerinde Evrimsel Algoritmalar İle Yapay Arı Kolonisi Algoritmasının Bütünleşik Bir Yaklaşımı Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü.
- PEZZELLA F., MORGANTI G., CIASCETTI G. (2008). "A Genetic Algorithm For The Flexible Job-Shop Scheduling Problem." *Computers & Operations Research*, 35(10), 3202-3212.
- RASIP Norhayati Mohd vd. (2014). A guided particle swarm optimization algorithm for nurse scheduling problem. *Appl. Math. Sci*, 8(113), 5625-5632.
- RECHENBERG Ingo, et al. (1994). *Evolution strategy, in computational intelligence: imitating life. Computational intelligence imitating life*. IEEE Press, Piscataway.
- RYDER Jack (2006). "*A Genetic Algorithm For Scheduling*."
- SABAR, Nasser R. vd. (2012). "A Honey-Bee Mating Optimization Algorithm For Educational Timetabling Problems." *European Journal Of Operational Research*, 216(3), 533-543.
- SCHWEFEL Hans-Paul, RUDOLPH Günter (1995). Contemporary evolution strategies. In: *European conference on artificial life*. Springer, Berlin, Heidelberg, p. 891-907.
- SHIAU Der-Fang (2011). "A Hybrid Particle Swarm Optimization for a University Course Scheduling Problem with Flexible Preferences," *Expert Systems with Applications*, vol. 38, p. 235–248.
- SIGL Branimir, GOLUB Marin, MORNAR Vedran (2003). Solving Timetable Scheduling Problem Using Genetic Algorithms. In *Information Technology Interfaces, 2003. ITI 2003. Proceedings Of The 25th International Conference On* (Pp. 519-524). IEEE.

- SONG Kwonsik vd. (2017). Energy Efficiency-Based Course Timetabling For University Buildings. *Energy*, 139, 394-405.
- STREİCHERT Felix (2002). Introduction To Evolutionary Algorithms. *Paper To Be Presented Apr, 4*.
- ŞAHİN Yusuf (2009). Depo Operasyonlarının Planlanması İçin Genetik Algoritma Esaslı Bir Model. MS Thesis. Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü.
- TAC Kadir Can (2006). Genetik Algoritma Kullanılarak Haftalık Ders Programı Zaman Çizelgeleme Yazılımının Geliştirilmesi, Marmara Üniversitesi Fen Bilimleri Enstitüsü Elektronik-Bilgisayar Eğitimi Anabilim Dalı.
- TASSOPOULOS Ioannis X., BELIGIANNIS Grigorios N. (2012). Using particle swarm optimization to solve effectively the school timetabling problem. *Soft Computing*, 16(7), 1229-1252.
- TAŞKIN Çağatan, EMEL Gül Gökay (2009). “Sayısal Yöntemlerde Genetik Algoritmalar”. Bursa.
- TOKMAK Mahmut (2011). Yapay arı kolonisi algoritması ile ders çizelgeleme probleminin çözümü, Doctoral dissertation, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Isparta.
- ULAK Ahmet (2010). “University Course Timetabling Using Multi Objective Genetic Algorithms.” Submitted To The Institute Of Graduate Studies İn Science And Engineering İn Partial Fulfillment Of The Requirements For The Degree Of Master Of Science İn Computer Engineering Yeditepe University.
- UYSAL Özgür (2006). “Comparison Of Genetic Algorithm And Particle Swarm Optimization Algorithm For Bicriteria Permutation Flowshop Scheduling Problem. Doktora Tezi, Marmara Üniversitesi, Endüstri Mühendisliği Programı.
- WU Tai-Hsi, YEH Jinn-Yi, LEE Yueh-Min (2015). A particle swarm optimization approach with refinement procedure for nurse rostering problem. *Computers & Operations Research*, 54, 52-63.

- YANG Shengxiang, JAT Sadaf Naseem (2011). Genetic Algorithms With Guided And Local Search Strategies For University Course Timetabling. *IEEE Transactions On Systems, Man, And Cybernetics, Part C (Applications And Reviews)*, 41(1), 93-106.
- YAPICI Muhammed Mutlu (2012). “Genetik Algoritma Kullanılarak Ders Çizelgeleme Yazılımının Geliştirilmesi.” Yüksek Lisans Tezi, Gazi Üniversitesi Bilişim Enstitüsü.
- YEO Michael Frank, AGYEI Eugene Osei (1998). Optimising Engineering Problems Using Genetic Algorithms. *Engineering Computations*, 15(2), 268-280.
- YİĞİT Vecihi (2011). Genetik Algoritma ile Türkiye Net Elektrik Enerjisi Tüketiminin 2020 Yılına Kadar Tahmini. *Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi*, 3(2), 37-41.
- YİĞİT Tuncay (2006). “Meslek Liseleri Haftalık Ders Çizelgelerinin Genetik Algoritmalar Yardımıyla Oluşturulması.” *Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi Dergisi*, 19, 25-39.
- YU Enzhe, SUNG Ki-Seok (2002). “A Genetic Algorithm For A University Weekly Courses Timetabling Problem.” *International Transactions In Operational Research*, 9(6), 703-717.
- ZHANG Rui, WU Cheng (2011). An artificial bee colony algorithm for the job shop scheduling problem with random processing times. *Entropy*, 13(9), 1708-1729.

EKLER

Ek 1. Yönetim Bilişim Sistemleri Bölümü 1. Öğretim Ders Programı

Günler Saatler	Pazartesi	Salı	Çarşamba	Perşembe	Cuma
09:00-09:50	<p>Borçlar Hukuku- F. Savaş-İİBF- B104-YBS1</p> <p>Görsel Programlama II- O.KAYNAR- İİBF-LABC- YBS2</p> <p>İnternet Programlama II- A.G.YÜKSEK- İİBF-LABD- YBS3</p>	<p>İstatistik II- Necati A. ERİLLİ-İİBF- BZ03-YBS2</p> <p>Yöneylem Araştırması II-E. KARAKIŞ-İİBF- BZ04-YBS3</p>	<p>İşletme Bilimine Giriş- A.R.İNCE- İİBF-B104- YBS1</p>	<p>İktisada Giriş II- Ali R. ATAMAN- İİBF-B104- YBS1</p>	
10:00-10:50	<p>Borçlar Hukuku- F. Savaş-İİBF- B104-YBS1</p> <p>Görsel Programlama II- O.KAYNAR- İİBF-LABC- YBS2</p> <p>İnternet Programlama II- A.G.YÜKSEK- İİBF-LABD- YBS3</p> <p>Sermaye Piyasası- Z. Şenol- İİBF- LABD-YBS4</p>	<p>İstatistik II- Necati A. ERİLLİ-İİBF- BZ03-YBS2</p> <p>Yöneylem Araştırması II-E. KARAKIŞ-İİBF- BZ04-YBS3</p> <p>İnsan Kaynakları Yönetimi- A.R.İNCE-İİBF- BB03-YBS4</p>	<p>İşletme Bilimine Giriş- A.R.İNCE- İİBF-B104- YBS1</p> <p>Sistem Analizi Ve Tasarımı- M.A.ALAN- İİBF-LABC- YBS2</p>	<p>İktisada Giriş II- Ali R. ATAMAN- İİBF-B104- YBS1</p> <p>İş Süreçleri Analizi ve Programlama- M.A.DEVECİ- İİBF-LABA- YBS2</p> <p>Kalite Yönetimi- A.R.İNCE-İİBF- BZ04-YBS3</p> <p>Bilişim Sistemleri Yönetimi ve Etiği- M.F.TUNA- İİBF-BB03- YBS4</p>	<p>İngilizce II-UZEM YD- İİBF-SYYBS-YBS1</p> <p>Sosyal Bilimlerde Yöntem-M. ASLAN- İİBF-BZ04-YBS3</p>
11:00-11:50	<p>Borçlar Hukuku- F. Savaş-İİBF- B104-YBS1</p> <p>Görsel Programlama II- O.KAYNAR- İİBF-LABC- YBS2</p> <p>İnternet Programlama II- A.G.YÜKSEK- İİBF-LABD- YBS3</p> <p>Sermaye Piyasası- Z. Şenol- İİBF- LABD-YBS4</p>	<p>İstatistik II- Necati A. ERİLLİ-İİBF- BZ03-YBS2</p> <p>Yöneylem Araştırması II-E. KARAKIŞ-İİBF- BZ04-YBS3</p> <p>İnsan Kaynakları Yönetimi- A.R.İNCE-İİBF- BB03-YBS4</p>	<p>İşletme Bilimine Giriş- A.R.İNCE- İİBF-B104- YBS1</p> <p>Sistem Analizi Ve Tasarımı- M.A.ALAN- İİBF-LABC- YBS2</p>	<p>İktisada Giriş II- Ali R. ATAMAN- İİBF-B104- YBS1</p> <p>İş Süreçleri Analizi ve Programlama- M.A.DEVECİ- İİBF-LABA- YBS2</p> <p>Kalite Yönetimi- A.R.İNCE-İİBF- BZ04-YBS3</p> <p>Bilişim Sistemleri Yönetimi ve Etiği- M.F.TUNA-</p>	<p>İngilizce II-UZEM YD- İİBF-SYYBS-YBS1</p> <p>Sosyal Bilimlerde Yöntem-M. ASLAN- İİBF-BZ04-YBS3</p>

				İİBF-BB03-YBS4	
13:00-13:50	<p>Matematik II-S.ŞENER-İİBF-B104-YBS1</p> <p>Mali Tablolara Analizi-M.DEMİR-İİBF-BZ04-YBS3</p>	<p>Programlama Teknikleri-E.GÜZELCE-İİBF-LABB-YBS1</p> <p>MuhasebeII-M.ÖZTÜRK-İİBF-BZ03-YBS2</p> <p>Örgütsel Davranış-F.ÇAMLİBEL-İİBF-BZ04-YBS3</p> <p>Veri Madenciliği-M.A.ALAN-İİBF-LABC-YBS4</p>	<p>Programlama Teknikleri-E.GÜZELCE-İİBF-LABB-YBS1</p> <p>Bilgisayar Uygulamalı Maliyet Muhasebesi-M.ÖZTÜRK-İİBF-LABD-YBS4</p>	<p>Pazarlama Yönetimi-M.F.TUNA-İİBF-BZ03-YBS2</p> <p>Veri İletimi ve Bilgisayar Ağları-E.GÜZELCE-İİBF-LABB-YBS3</p> <p>Türk Vergi Sistemi-B.DEMİRGİL-İİBF-BB03-YBS4</p>	<p>Türk Dili II-UZEM TD-İİBF-STYBS-YBS1</p> <p>Veri Tabanı Yönetim Sistemleri II-E.GÜZELCE-İİBF-LABC-YBS2</p> <p>Kurumsal Kaynak Planlama-F.DEMİRKOPARAN-İİBF-LABA-YBS3</p>
14:00-14:50	<p>Matematik II-S.ŞENER-İİBF-B104-YBS1</p> <p>Mali Tablolara Analizi-M.DEMİR-İİBF-BZ04-YBS3</p>	<p>Programlama Teknikleri-E.GÜZELCE-İİBF-LABB-YBS1</p> <p>Örgütsel Davranış-F.ÇAMLİBEL-İİBF-BZ04-YBS3</p> <p>Veri Madenciliği-M.A.ALAN-İİBF-LABC-YBS4</p>	<p>Programlama Teknikleri-E.GÜZELCE-İİBF-LABB-YBS1</p> <p>Bilgisayar Uygulamalı Maliyet Muhasebesi-M.ÖZTÜRK-İİBF-LABD-YBS4</p>	<p>Pazarlama Yönetimi-M.F.TUNA-İİBF-BZ03-YBS2</p> <p>Veri İletimi ve Bilgisayar Ağları-E.GÜZELCE-İİBF-LABB-YBS3</p> <p>Türk Vergi Sistemi-B.DEMİRGİL-İİBF-BB03-YBS4</p>	<p>Türk Dili II-UZEM TD-İİBF-STYBS-YBS1</p> <p>Veri Tabanı Yönetim Sistemleri II-E.GÜZELCE-İİBF-LABC-YBS2</p> <p>Kurumsal Kaynak Planlama-F.DEMİRKOPARAN-İİBF-LABA-YBS3</p>
15:00-15:50	<p>Matematik II-S.ŞENER-İİBF-B104-YBS1</p>	<p>Örgütsel Davranış-F.ÇAMLİBEL-İİBF-BZ04-YBS3</p> <p>Veri Madenciliği-M.A.ALAN-İİBF-LABC-YBS4</p>		<p>Pazarlama Yönetimi-M.F.TUNA-İİBF-BZ03-YBS2</p> <p>Veri İletimi ve Bilgisayar Ağları-E.GÜZELCE-İİBF-LABB-YBS3</p> <p>Türk Vergi Sistemi-B.DEMİRGİL-İİBF-BB03-YBS4</p>	<p>Veri Tabanı Yönetim Sistemleri II-E.GÜZELCE-İİBF-LABC-YBS2</p> <p>Kurumsal Kaynak Planlama-F.DEMİRKOPARAN-İİBF-LABA-YBS3</p>

Ek 2. Yönetim Bilişim Sistemleri Bölümü 2. Öğretim Ders Programı

Günler Saatler	Pazartesi	Salı	Çarşamba	Perşembe	Cuma
16:00-16:50	<p>Borçlar Hukuku- F. Savaş-İİBF- B104-YBS1</p> <p>Görsel Programlama II- O.KAYNAR- İİBF-LABC- YBS2</p> <p>İnternet Programlama II- A.G.YÜKSEK- İİBF-LABD- YBS3</p>	<p>İstatistik II- Necati A. ERİLLİ-İİBF- BZ03-YBS2</p> <p>Yöneylem Araştırması II-E. KARAKIŞ-İİBF- BZ04-YBS3</p>	<p>İşletme Bilimine Giriş- A.R.İNCE- İİBF-B104- YBS1</p>	<p>İktisada Giriş II- Ali R. ATAMAN- İİBF-B104- YBS1</p>	
17:00-17:50	<p>Borçlar Hukuku- F. Savaş-İİBF- B104-YBS1</p> <p>Görsel Programlama II- O.KAYNAR- İİBF-LABC- YBS2</p> <p>İnternet Programlama II- A.G.YÜKSEK- İİBF-LABD- YBS3</p> <p>Sermaye Piyasası- Z. Şenol- İİBF- LABD-YBS4</p>	<p>İstatistik II- Necati A. ERİLLİ-İİBF- BZ03-YBS2</p> <p>Yöneylem Araştırması II-E. KARAKIŞ-İİBF- BZ04-YBS3</p> <p>İnsan Kaynakları Yönetimi- A.R.İNCE-İİBF- BB03-YBS4</p>	<p>İşletme Bilimine Giriş- A.R.İNCE- İİBF-B104- YBS1</p> <p>Sistem Analizi Ve Tasarımı- M.A.ALAN- İİBF-LABC- YBS2</p>	<p>İktisada Giriş II- Ali R. ATAMAN- İİBF-B104- YBS1</p> <p>İş Süreçleri Analizi ve Programlama- M.A.DEVECİ- İİBF-LABA- YBS2</p> <p>Kalite Yönetimi- A.R.İNCE-İİBF- BZ04-YBS3</p> <p>Bilişim Sistemleri Yönetimi ve Etiği- M.F.TUNA- İİBF-BB03- YBS4</p>	<p>İngilizce II-UZEM YD- İİBF-SYYBS-YBS1</p> <p>Sosyal Bilimlerde Yöntem-M. ASLAN- İİBF-BZ04-YBS3</p>
18:00-18:50	<p>Borçlar Hukuku- F. Savaş-İİBF- B104-YBS1</p> <p>Görsel Programlama II- O.KAYNAR- İİBF-LABC- YBS2</p> <p>İnternet Programlama II- A.G.YÜKSEK- İİBF-LABD- YBS3</p> <p>Sermaye Piyasası- Z. Şenol- İİBF- LABD-YBS4</p>	<p>İstatistik II- Necati A. ERİLLİ-İİBF- BZ03-YBS2</p> <p>Yöneylem Araştırması II-E. KARAKIŞ-İİBF- BZ04-YBS3</p> <p>İnsan Kaynakları Yönetimi- A.R.İNCE-İİBF- BB03-YBS4</p>	<p>İşletme Bilimine Giriş- A.R.İNCE- İİBF-B104- YBS1</p> <p>Sistem Analizi Ve Tasarımı- M.A.ALAN- İİBF-LABC- YBS2</p>	<p>İktisada Giriş II- Ali R. ATAMAN- İİBF-B104- YBS1</p> <p>İş Süreçleri Analizi ve Programlama- M.A.DEVECİ- İİBF-LABA- YBS2</p> <p>Kalite Yönetimi- A.R.İNCE-İİBF- BZ04-YBS3</p> <p>Bilişim Sistemleri Yönetimi ve Etiği- M.F.TUNA- İİBF-BB03- YBS4</p>	<p>İngilizce II-UZEM YD- İİBF-SYYBS-YBS1</p> <p>Sosyal Bilimlerde Yöntem-M. ASLAN- İİBF-BZ04-YBS3</p>

<p>19:00-19:50</p>	<p>Matematik II-S.ŞENER-İİBF-B104-YBS1</p> <p>Mali Tablolar Analizi-M.DEMİR-İİBF-BZ04-YBS3</p>	<p>Programlama Teknikleri-E.GÜZELCE-İİBF-LABB-YBS1</p> <p>MuhasebeII-M.ÖZTÜRK-İİBF-BZ03-YBS2</p> <p>Örgütsel Davranış-F.ÇAMLİBEL-İİBF-BZ04-YBS3</p> <p>Veri Madenciliği-M.A.ALAN-İİBF-LABC-YBS4</p>	<p>Programlama Teknikleri-E.GÜZELCE-İİBF-LABB-YBS1</p> <p>Bilgisayar Uygulamalı Maliyet Muhasebesi-M.ÖZTÜRK-İİBF-LABD-YBS4</p>	<p>Pazarlama Yönetimi-M.F.TUNA-İİBF-BZ03-YBS2</p> <p>Veri İletimi ve Bilgisayar Ağları-E.GÜZELCE-İİBF-LABB-YBS3</p> <p>Türk Vergi Sistemi-B.DEMİRGİL-İİBF-BB03-YBS4</p>	<p>Türk Dili II-UZEM TD-İİBF-STYBS-YBS1</p> <p>Veri Tabanı Yönetim Sistemleri II-E.GÜZELCE-İİBF-LABC-YBS2</p> <p>Kurumsal Kaynak Planlama-F.DEMİRKOPARAN-İİBF-LABA-YBS3</p>
<p>20:00-20:50</p>	<p>Matematik II-S.ŞENER-İİBF-B104-YBS1</p> <p>Mali Tablolar Analizi-M.DEMİR-İİBF-BZ04-YBS3</p>	<p>Programlama Teknikleri-E.GÜZELCE-İİBF-LABB-YBS1</p> <p>Örgütsel Davranış-F.ÇAMLİBEL-İİBF-BZ04-YBS3</p> <p>Veri Madenciliği-M.A.ALAN-İİBF-LABC-YBS4</p>	<p>Programlama Teknikleri-E.GÜZELCE-İİBF-LABB-YBS1</p> <p>Bilgisayar Uygulamalı Maliyet Muhasebesi-M.ÖZTÜRK-İİBF-LABD-YBS4</p>	<p>Pazarlama Yönetimi-M.F.TUNA-İİBF-BZ03-YBS2</p> <p>Veri İletimi ve Bilgisayar Ağları-E.GÜZELCE-İİBF-LABB-YBS3</p> <p>Türk Vergi Sistemi-B.DEMİRGİL-İİBF-BB03-YBS4</p>	<p>Türk Dili II-UZEM TD-İİBF-STYBS-YBS1</p> <p>Veri Tabanı Yönetim Sistemleri II-E.GÜZELCE-İİBF-LABC-YBS2</p> <p>Kurumsal Kaynak Planlama-F.DEMİRKOPARAN-İİBF-LABA-YBS3</p>
<p>21:00-21:50</p>	<p>Matematik II-S.ŞENER-İİBF-B104-YBS1</p>	<p>Örgütsel Davranış-F.ÇAMLİBEL-İİBF-BZ04-YBS3</p> <p>Veri Madenciliği-M.A.ALAN-İİBF-LABC-YBS4</p>		<p>Pazarlama Yönetimi-M.F.TUNA-İİBF-BZ03-YBS2</p> <p>Veri İletimi ve Bilgisayar Ağları-E.GÜZELCE-İİBF-LABB-YBS3</p> <p>Türk Vergi Sistemi-B.DEMİRGİL-İİBF-BB03-YBS4</p>	<p>Veri Tabanı Yönetim Sistemleri II-E.GÜZELCE-İİBF-LABC-YBS2</p> <p>Kurumsal Kaynak Planlama-F.DEMİRKOPARAN-İİBF-LABA-YBS3</p>

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Ahmet YURTSAL
Uyruğu : T.C.
Doğum Tarihi ve Yeri : 1991/Sivas
E-posta : ahmet_yurtsal@hotmail.com

EĞİTİM

Derece	Kurum	Mezuniyet Yılı
Lisans	Cumhuriyet Üniversitesi	2015
Yüksek Lisans	Cumhuriyet Üniversitesi	2019

İŞ TECRÜBESİ

Tarih	Kurum	Görev
23.10.2017	Hatay Mustafa Kemal Üniversitesi	Araştırma Görevlisi