



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

METASEZGİSEL ALGORİTMALAR KULLANILARAK SINAV ÇİZELGELEME

Gökhan SEYFİ

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalını

Ocak-2018
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Gökhan Seyfi. tarafından hazırlanan "METASEZGİSEL ALGORİTMALAR KULLANILARAK SINAV ÇİZELGELEME" adlı tez çalışması 19/01/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü BİLGİSAYAR MÜHENDİSLİĞİ Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Doç. Dr. Halife KODAZ	.....
Danışman Doç. Dr. Mustafa Servet KIRAN	.....
Üye Yrd. Doç. Dr. Mehmet HACIBEYOĞLU	.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mustafa YILMAZ.
FBE Müdürü

Bu tez çalışması ..... tarafından ..... nolu proje ile desteklenmiştir.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.



Gökhan SEYFİ

Tarih: 19/01/2018

ÖZET

YÜKSEK LİSANS TEZİ

METASEZGİSEL ALGORİTMALAR KULLANILARAK SINAV ÇİZELGELEME

Gökhan SEYFİ

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mustafa Servet KIRAN

2018, 84

Jüri
Doç. Dr. Mustafa Servet KIRAN
Doç. Dr. Halife KODAZ
Yrd. Doç. Dr. Mehmet HACIBEYOĞLU

Sınav çizelgeleme problemi belirli sayıdaki sınavları zaman dilimlerine çeşitli esnek ve sert kısıtları göz önünde bulundurarak atama işlemidir. Bu işlem oldukça zaman alıcı, karmaşık ve zor bir görevdir ve NP zor problem olarak sınıflandırılmaktadır.

Gerçek dünyada genelde sınav çizelgeleri el ile yapılmaktadır. Bu süreç oldukça zaman alıcı ve hatalara açıktır. Ayrıca elde edilen sonuçlar genelde tatmin edici olmamaktadır. Bu tez kapsamında problemin çözümü için üç farklı evrimsel hesaplama algoritması kullanılmıştır. Bunlar genetik algoritma (GA), parçacık sürüsü optimizasyonu (PSO) ve yapay arı kolonisi algoritması (YAKA)'dır.

Bu çalışmada GA, PSO-S, YAKA-S algoritmalarının zaman çizelgelemesi probleminin optimal çözümünün elde edilmesinde performansları incelenmiştir. Bu amaçla Selçuk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'ne ait dersleri içeren veri kümeleri kullanılmıştır. Elde edilen sonuçlar incelendiğinde önerilen yöntemler sınav çizelgeleme problemlerinin çözümünde kullanılabilir ve elle hazırlanan yöntemlerden çözüm kalitesi ve kısıt tatmini açısından daha iyi sonuçlar üretmişlerdir.

Anahtar Kelimeler: Genetik Algoritma, Sınav Çizelgeleme, Parçacık Sürü Optimizasyonu Algoritması, Yapay Arı Kolonisi Algoritması

ABSTRACT

MS THESIS
EXAM TIMETABLING USING METAHEURISTIC ALGORITHMS

Gökhan SEYFİ

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE IN COMPUTER ENGINEERING

Advisor: Assoc. Prof. Dr. Mustafa Servet KIRAN

2018, 84

Jury
Assoc. Prof. Dr. Mustafa Servet KIRAN
Assoc. Prof. Dr. Halife KODAZ
Assist. Prof. Dr. Mehmet HACIBEYOĞLU

The aim of examination scheduling problem is to assign the exams to number of time periods (time slots) by taking into account some hard or soft constraints. This is a time consuming, complicated and hard task, and this problem is categorized as NP-hard.

Mostly, the examination timetables are manually made in the real world. This process is time consuming and can cause some faults and obtained results are not satisfactory in general. In this study, we propose three methods to solve this problem by utilizing genetic algorithm (GA), particle swarm optimization (PSO) algorithm and artificial bee colony (ABC) algorithm.

This thesis investigates and compares the performances of GA, PSO-E and ABC-E algorithms for an optimal solution of a timetable problem. The test set is taken from courses of Computer Engineering Department at Selcuk University. The obtained results show that the proposed algorithms can be used for solving this problem and obtained solutions by these methods are better than manually prepared timetables in terms of solution quality and constraints satisfaction.

Keywords: Genetic Algorithms, Examination Scheduling, Particle Swarm Optimization Algorithm, Artificial Bee Colony Algorithm

ÖNSÖZ

Çalışmamda, bana her zaman destek olan aileme, benden yardımlarını hiçbir zaman esirgemeyen çok değerli danışman hocam Doç. Dr. Mustafa Servet KIRAN'a, kendilerinden aldığım fikirlerle bana yol gösteren, desteğini ve bilgisini esirgemeyen değerli hocam Uzman Sedat KORKMAZ'a ve arkadaşım Ahmet Cevahir ÇINAR'a, Konya Selçuk Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim elemanlarına ve sayın jüri üyelerine sonsuz teşekkürler ederim.

Gökhan SEYFİ
Konya-2018

ŞEKİLLER ve ÇİZELGELER

ŞEKİLLER

Şekil 1.1 Zaman çizelgesinin içeriği ve ilişkileri ... 3
Şekil 1.2 Sınav çizelgesinin içeriği ve ilişkileri ... 5
Şekil 3.1 Genetik algoritmanın genel akış diyagramı ... 17
Şekil 3.2 Örnek bir kromozom temsili ... 19
Şekil 3.3 Tam sayı ile örnek bir kromozom temsili ... 19
Şekil 3.4 Tek noktalı çaprazlama ... 21
Şekil 3.5 Çok noktalı çaprazlama ... 22
Şekil 3.6 Tekdüze çaprazlama ... 22
Şekil 3.7 Bit tersleme mutasyonu ... 23
Şekil 3.8 Takas mutasyonu ... 23
Şekil 3.9 Örnek bir parçacık ... 27
Şekil 3.10 Kaba kod PSO ... 27
Şekil 3.11 Bazı komşuluk topolojileri ... 28
Şekil 3.12 Kaba kod YAKA ... 33
Şekil 3.13 Sınav Çizelgeleme için GA Kromozom Yapısı ... 39
Şekil 3.14 Örnek Kromozom ... 39
Şekil 3.15 PSO-S kaba kodu ... 42
Şekil 3.16 Rastgele alt dizi ters takas operatörü ... 43
Şekil 3.17 YAKA-S Kaba kod ... 44
Şekil 4.1 Vize deneyi 1 ... 48
Şekil 4.2 Vize deneyi 2 ... 49
Şekil 4.3 Vize deneyi 3 ... 50
Şekil 4.4 Vize deneyi 4 ... 51
Şekil 4.5 Vize deneyi 5 ... 52
Şekil 4.6 Vize deneyi 6 ... 53
Şekil 4.7 Vize deneyi 7 ... 54
Şekil 4.8 Vize deneyi 8 ... 55
Şekil 4.9 Vize deneyi 9 ... 56
Şekil 4.10 Vize deneyi 10 ... 57
Şekil 4.11 Popülasyon boyutunun vize sınavında yöntemlerin başarısına etkisi ... 59
Şekil 4.12 Final deneyi 1 ... 61
Şekil 4.13 Final deneyi 2 ... 62
Şekil 4.14 Final deneyi 3 ... 63
Şekil 4.15 Final deneyi 4 ... 64
Şekil 4.16 Final deneyi 5 ... 65
Şekil 4.17 Final deneyi 6 ... 66
Şekil 4.18 Final deneyi 7 ... 67
Şekil 4.19 Final deneyi 8 ... 68
Şekil 4.20 Final deneyi 9 ... 69
Şekil 4.21 Final deneyi 10 ... 70
Şekil 4.22 Popülasyon boyutunun final sınavında yöntemlerin başarısına etkisi ... 71

ÇİZELGELER

Çizelge 1.1 Eğitimde karşılaşılan çizelgeleme problemleri 4
Çizelge 1.2 Örnek sınav çizelgesi 5
Çizelge 4.1 Veri kümesi özellikleri 45
Çizelge 4.2 Sınav çizelgelemede kullanılan parametreler 46
Çizelge 4.3 Ders listesi 47
Çizelge 4.4 Vize deney 1 sonuçlarının kıyaslanması 49
Çizelge 4.5 Vize deney 2 sonuçlarının kıyaslanması 50
Çizelge 4.6 Vize deney 3 sonuçlarının kıyaslanması 51
Çizelge 4.7 Vize deney 4 sonuçlarının kıyaslanması 52
Çizelge 4.8 Vize deney 5 sonuçlarının kıyaslanması 53
Çizelge 4.9 Vize deney 6 sonuçlarının kıyaslanması 54
Çizelge 4.10 Vize deney 7 sonuçlarının kıyaslanması 55
Çizelge 4.11 Vize deney 8 sonuçlarının kıyaslanması 56
Çizelge 4.12 Vize deney 9 sonuçlarının kıyaslanması 57
Çizelge 4.13 Vize deney 10 sonuçlarının kıyaslanması 58
Çizelge 4.14 Vize deney sonuçlarının ortalamalarının toplu kıyaslanması 58
Çizelge 4.15 GA ile elde edilen vize çizelgesi 60
Çizelge 4.16 Final deney 1 sonuçlarının kıyaslanması 61
Çizelge 4.17 Final deney 2 sonuçlarının kıyaslanması 62
Çizelge 4.18 Final deney 3 sonuçlarının kıyaslanması 63
Çizelge 4.19 Final deney 4 sonuçlarının kıyaslanması 64
Çizelge 4.20 Final deney 5 sonuçlarının kıyaslanması 65
Çizelge 4.21 Final deney 6 sonuçlarının kıyaslanması 66
Çizelge 4.22 Final deney 7 sonuçlarının kıyaslanması 67
Çizelge 4.23 Final deney 8 sonuçlarının kıyaslanması 68
Çizelge 4.24 Final deney 9 sonuçlarının kıyaslanması 69
Çizelge 4.25 Final deney 10 sonuçlarının kıyaslanması 70
Çizelge 4.26 Final deney sonuçlarının ortalamalarının toplu kıyaslanması 71
Çizelge 4.27 GA ile elde edilen final çizelgesi birinci hafta 72
Çizelge 4.28 GA ile elde edilen final çizelgesi ikinci hafta 72

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

ŞEKİLLER ve ÇİZELGELER ..... **vii**

İÇİNDEKİLER ..... **ix**

SİMGELER VE KISALTMALAR ..... **xi**

1. GİRİŞ **1**

1.1. Çizelgeleme Problemleri 3
1.2. Sınav Çizelgeleme Problemi 5

2. KAYNAK ARAŞTIRMASI **6**

2.1. Graf Tabanlı Teknikler 6
2.2. Kısıt Tatmini Teknikleri 6
2.3. Yerel Arama Tabanlı Teknikler 7
2.4. Popülasyon Tabanlı Algoritmalar 8
2.4.1. Evrimsel hesaplama tabanlı 8
2.4.2 Sürü zekası tabanlı optimizasyon algoritmaları 11
2.5. Çoklu Kriterli Teknikler 14
2.6. Üst Sezgiseller(Hyper Heuristics) 14

3. MATERYAL VE YÖNTEM **16**

3.1. Genetik Algoritmalar 16
3.1.1. Kromozom gösterimi ve amaç fonksiyonu 18
3.1.2. Değişkenlerin tanımlanması ve kodlama 19
3.1.3. Popülasyonun İlklendirilmesi 20
3.1.4. Doğal seçim 20
3.1. 5. Çaprazlama 21
3.1. 6. Mutasyon 22
3.1. 7. Sonlandırma koşulu 23
3.2. Parçacık Sürü Optimizasyonu (PSO) 24
3.2.1. PSO algoritması 25
3.3. Yapay Arı Kolonisi Algoritması 29
3.3.1. Doğada bal arılarının davranışları 29
3.3.2. YAKA'nın algoritmik yapısı 30
3.3.4. YAKA'nın ana adımları 32
3.3.5. YAKA kontrol parametreleri 33
3.4. Sınav Çizelgelemesi Problemi 33
3.4.1. Parametreler ve matematiksel gösterim 35

3.4.2. Sert kısıtlar 36
3.4.3. Esnek kısıtlar 37
3.5. Genetik Algoritmanın Sınav Çizelgeleme Problemine Uygulanması 38
3.5.1. GA parametrelerinin tanımlanması ve kodlama 38
3.5.2. Başlangıç popülasyonu , doğal seçim ve uygunluk fonksiyonu 39
3.5.3. Çaprazlama 40
3.5.4. Mutasyon 40
3.5.5. Gelecek nesil ve yakınsama 40
3.6. PSO Algoritmasının Sınav Çizelgeleme Problemine Uygulanması 40
3.6.1. Sınav programı için PSO-S algoritması elemanları 41
3.7. YAKA'nın Sınav Çizelgeleme Problemine Uygulanması 42
3.7.1. Sınav programı için YAKA-S elemanları 42

4. ARAŞTIRMA BULGULARI VE TARTIŞMA 45

4.1. Deney Setinin Özellikleri 45
4.2. Kullanılan Parametreler 46
4.2.1. Vize sınavı deneyleri 47
4.2.1. Final sınavı deneyleri 60
4.2.1. Deney sonuçlarının yorumlanması 72

5. SONUÇ VE TARTIŞMA 75

6. KAYNAKLAR 77

7. ÖZGEÇMİŞ 84

SİMGELER VE KISALTMALAR

Simgeler

r	0 ile 1 arasında rastgele üretilen sayı
p_c	Çaprazlama olasılığı
c1	Bilişsel davranış katsayısı
c2	Sosyal davranış katsayısı
X	Parçacığın konumu
V	Parçacığın hızı
i	Parçacık, birey, temsilci numarası
j	Parçacık, birey, temsilci boyutu
p	Seçme olasılığı

Kısaltmalar

GA	Genetik algoritma
GD	Büyük Tufan Algoritması
YAKA	Yapay Arı Kolonisi Algoritması
PSO	Parçacık Sürü Optimizasyonu
YAKA-S	Yapay Arı Kolonisi Algoritması Sınav
PSO-S	Parçacık Sürü Optimizasyonu Sınav
FESP	Final sınavı çizelgeleme paketi
NP	Deterministik olmayan polinomsal
GRASP	Açgözlü Rasgele Uyumlu Arama Prosedürleri
GPU	Grafik İşleme Ünitesi
MA	Memetik Algoritma
P	Gen
N	Kromozom Boyutu
F	Uygunluk Değeri (Hedef Fonksiyon)
$N_{tutulacak}$	Eşleşmeden sonra popülasyonda tutulacak birey sayısı
N_{deg}	boyut sayısı
K	Rastgele seçilen birey adedi
D	Boyut
SN	Sürüdeki arı adedi
fes	Fonksiyon değerlendirme sayısı

1. GİRİŞ

Çizelgeleme problemi, uyulması zorunlu olan ve isteğe bağlı olan kısıtların sağlanarak gerçekleştirilecek olan görevlerin kaynaklara atanması işlemidir (Kalaycı 2008). Çizelgeleme eğitim, ulaşım, sağlık, spor gibi birçok alanda karşımıza çıkmaktadır.

İyi bir çizelgeleme hazırlanması oluşturulan çizelgelerde kullanılan kaynakların ve kişilerin daha etkili ve başarılı kullanımı açısından büyük önem arz etmektedir. Örneğin sağlık kurumlarında karşımıza çıkan hemşire çizelgelemesinde, hemşirelerin görevleri sırasındaki verimlilikleri hazırlanan çizelgenin kalitesiyle doğru orantılıdır. Kaliteli çizelgeler öğrencilerin girdikleri dersler de ve sınavlarda en iyi sonuçları elde etmeleri açısından büyük öneme sahiptir.

Üniversitelerde ders programı veya sınav programı hazırlanması da bir çizelgeleme problemi olarak karşımıza çıkmaktadır. Ders programında derslerin uygun şartlar altında sınıflara zaman kısıtı dikkate alınarak atanması beklenirken, sınav programında da benzer bir durum söz konusudur. Sınav programı hazırlanması ders programı hazırlanmasına göre daha zor bir problem olarak karşımıza çıkmaktadır çünkü ders programında atanacak dersler kredilerine göre birden fazla güne bölünebilirken sınav programındaki dersler için bu durum söz konusu değildir. Böylelikle araştırma uzayının da önemli bir miktar küçüldüğü düşünülebilir. Genellikle sınav programları bölüm öğretim elemanlarınca hazırlanmaktadır, çok aşamalı kararların iç içe geçmiş bir şekilde bulunmasından dolayı hataya açık bir süreç yürütülmektedir ve genellikle elde edilen çizelgeler de optimum çözümden uzak olmaktadır.

Çizelgeleme problemleri ayrık problemlerdir ve genellikle NP, NP-Zor veya NP-tam problem kategorilerinden birinde incelenir (Lovelace 2010). Giriş sayısının küçük olduğu durumlarda bu problemler için optimum çözüm bulunabilmesine rağmen genellikle görece büyük boyutlu problemler için optimum çözümün elde edilmesi girişe bağlı üstel bir çalışma zamanı gerektirmektedir. Bu da optimum çözümün girişe bağlı polinom adım sayısında elde edilemeyeceğini ifade eder. Ayrıca çözüm uzayı kaba güç ve dolaylı arama yöntemlerinin uygulanabilmesi için oldukça büyük olabilmektedir ve bu yöntemler zamansal olarak verimsiz olarak kabul edilmektedir. Zaman çizelgeleme problemleri çizelgeleme görevinin detaylarının çokluğundan ötürü karmaşıktır. Bu yapılarından dolayı çizelgeleme problemlerinin çözümü için evrimsel hesaplama veya sürü zekâsı temelli algoritmaların popülaritesinin oldukça arttığı görülmektedir. Bu

yöntemler optimum çözümü garanti edemezler fakat makul zamanlarda optimum veya optimuma yakın bir çözümü garanti edebilirler (Karaboğa 2014).

Sınav çizelgeleme problemleri araştırmacıların kombinatoryal çözüm yöntemlerini test edebilecekleri zorluk seviyesinde olduğu için çok ilgi çekmiştir. Graf boyama (Malkawi ve ark 2008), sezgisel yöntemler (heuristics) (Fong ve ark 2015), tamsayı programlama (Daskalaki ve Birbas 2005), yapay sinir ağları (Smith ve ark 2003), tabu arama (White ve ark 2004), evrimsel algoritmalar (Burke ve Newall 1999) ve kısıt mantık programlama (Boizumault ve ark 1996) gibi çeşitli yaklaşımlar bu problemlerin çözümü için kullanılmıştır.

Bu çalışma kapsamında ele alınan çizelgeleme problemi, bu problemin bir gerçek dünya problemi olduğundan dolayı standart bir çözüm yolu yoktur ve bu yüzden çözümü ile ilgili standart bir model bulunmamaktadır. Her bir kurumun sahip olduğu kaynakları ve başarım ölçüm kıstasları aynı değildir. Bu yüzden oluşturulan çizelgelerin kalitesinin değerlendirilmesi de kurumdan kuruma ya da kişiden kişiye göre değişmektedir. Öznel bir durumun genel bir yöntemle değerlendirilemeyeceği göz önüne alındığında bu çalışma kapsamında yapılan testler, karşılaştırmalar ve analizler de Selçuk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nün kaynakları, kısıtları, öğrencileri dikkate alınarak yapılmıştır.

Yapılan çalışma bölümdeki öğrencilerin başarısını ve verimliliğini arttırmak açısından önemlidir. Bu çalışma sayesinde öğrenciler daha adil sınav takvimlerine sahip olacaklardır.

Yapılan çalışmanın amacı Selçuk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'ne ait sınav çizelgeleme problemlerinin genetik algoritma (GA), parçacık sürü optimizasyonu (PSO-S) algoritması ve yapay arı kolonisi algoritması (YAKA-S) kullanılarak çözülmesi ve bu 3 yöntemin karşılaştırılmasıdır.

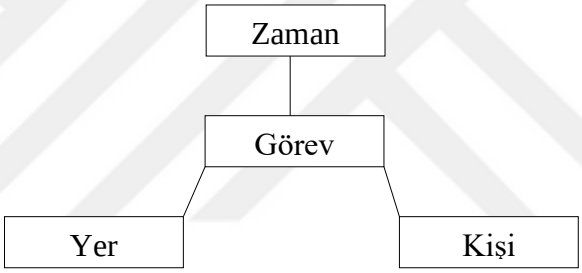
İkinci olarak herhangi bir öğrencinin girdiği sınavların arasındaki boş zaman arttırılarak öğrencilerin ders çalışma zamanları arttırılması amaçlanmıştır

Ana motivasyonumuz ise sürekli optimizasyon problemlerinin çözümünde kullanılan PSO ve YAKA yöntemlerinin ayrık bir problem olan sınav çizelgeleme probleminin çözümünde kullanılmasıdır. Bu amaçla iki algoritma üzerinde çeşitli değişiklikler yapılmış ve problem üzerinde kullanılmıştır. Bunun dışında aynı problem karşılaştırma maksadıyla GA yöntemi ile de çözülmüştür.

Bir diğer motivasyonumuz ise çözüm sürecini yavaşlatan ve karmaşıklaştıran tamir fonksiyonlarını kullanmak yerine alternatif çözümler kullanmaktır. Bu amaçla kullanılan üç yöntem için tamir gerektirmeyen bir çözüm temsili (gösterimi) seçilmiştir.

1.1. Çizelgeleme Problemleri

Zaman çizelgeleme, çeşitli kısıtlamalara tabi olacak şekilde belirli zaman aralığında verilen kaynakların nesnelere atanması işlemdir (Wren 1995). Daha açık bir şekilde ifade edecek olursak bir zaman çizelgesi çeşitli görevlerin ne zaman, kim veya ne tarafından gerçekleştirileceğini gösteren tablolardır. Bir zaman çizelgesi hazırlarken ilgili bütün kişi ve nesnelerin mümkün olduğunca talepleri karşılanmalıdır. Ayrıca herhangi bir kişiye veya nesneye aynı anda birden fazla görev verilmemelidir. Şekil 1.1'de görülebileceği gibi genel bir zaman çizelgesinin içeriği ve ilişkileri gösterilmiştir.



Şekil 1.1 Zaman çizelgesinin içeriği ve ilişkileri

Çizelgeleme problemleri çok farklı alanlarda karşımıza çıkmaktadır ve birçok araştırmacının ilgisini çekmiştir. Üzerinde çok sayıda araştırma yapılmış popüler bir problemdir. Literatürde çok farklı alanlardaki çizelgeleme problemleri ile ilgili çalışmalar mevcuttur. Zaman çizelgelemenin en yaygın örnekleri; hemşire (Berrada ve ark 1996), doktor (Cayirli ve ark 2006), askeriye (Lee ve ark 2009), ders programları (Lü ve Hao 2010), sınav programları (Mansour ve Timani 2007), havayolları uçuş çizelgeleri (Grönkvist 2006), personel çizelgeleme (Aickelin ve ark 2009) vb. problemlerdir.

Eğitim kurumlarında karşılaşılan çizelgeleme problemleri genellikle üç ana gruba ayrılır. Bu üç gruptaki problemlerin, çatışmasız olma bir başka deyişle bir öğrenci veya gözetmenin aynı anda birden fazla sınava katılamama gibi karakterleri birbirine benzerdir. Ancak bazı önemli farklılıkları vardır. Okul çizelgeleme problemlerinde öğrencilerin derslere tam zamanlı sınıflarda girmesi istenirken, üniversite ders

çizelgeleme problemlerinde bütün haftaya yayılan aynı zamanda öğrencilerin ders seçme işlemlerinin esnek olduğu derslere katılmaları gerekmektedir. Sınav çizelgelemeyle ise bir yarı dönemde iki veya üç kez karşılaşılır, bir veya iki hafta sürer. Bu problemde ise herhangi bir öğrenci için eğer mümkünse öğrenci sınav performansını olumsuz etkilememek adına bir günde birden fazla sınav atamaktan kaçınılmalıdır. Çizelge 1.1'de gösterilenlerin dışında eğitim kurumlarında sınavlara gözetmen atama da bir problem olarak karşımıza çıkmaktadır ve üzerine çeşitli araştırmalar yapılmıştır.

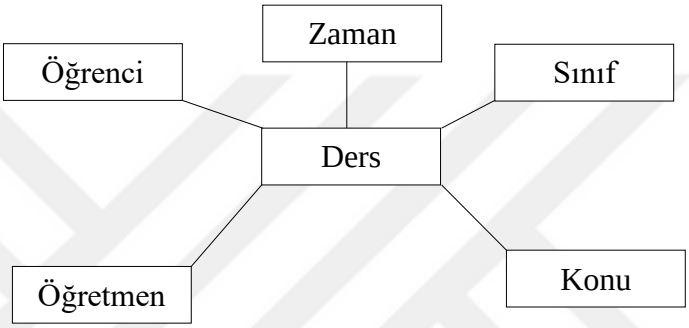
Çizelge 1.1 Eğitimde karşılaşılan çizelgeleme problemleri

Kategori	Tanımlama
Okul Çizelgeleme	Hafta boyunca okul sınıflarının derslere zamansal olarak ataması yapılmalıdır. Herhangi bir öğrenci veya öğretmenin aynı anda iki farklı sınıfta olma ihtiyacı yoktur
Üniversite Ders Çizelgeleme	Hafta boyunca derslerin sınıflara ve zamanlara ataması yapılmalıdır. Sınıf kapasiteleri göz önüne alınarak aynı öğrenci veya öğretmenin aynı anda iki farklı sınıfa atanmaması gerekmektedir.
Üniversite Sınav Çizelgeleme	Sınavlar sınırlı bir süreye çizelgelenmelidir (süreler genellikle bir veya iki haftadır). Bir öğrenci aynı anda iki sınava atanmamalıdır. Çoğu durumda sınıf kapasitesi de aynı anda gözetilmelidir.

Tam olarak belirlenmiş bir algoritmayla çözülebilen problemlerde, problemin boyutunun artmasıyla birlikte çözüm için gereken zaman da artmaktadır. Bu zamanın nasıl arttığı da algoritmaların zaman karmaşıklığının hesaplanmasıyla elde edilmektedir. Çizelgeleme problemleri çok sayıda karar mekanizması içermektedir. Bu kararlar da birbirini etkilediği için de çözümü oldukça zordur ve standart bir çözüm yöntemi bulunmamaktadır. Literatürde çizelgeleme problemlerinin karmaşıklığıyla ilgili çeşitli araştırmalar yapılmıştır. Cooper ve Kingston (Cooper ve Kingston 1995) hazırlamış olduğu çalışmada zaman çizelgelemede ortaya çıkan çeşitli NP-tam alt problemlerin varlığını ve bu problemlerin nasıl çözüleceğini göstermiştir. Bunun dışında çok sayıda araştırma bu problemin karmaşıklığını incelemiş problemi NP, NP-tam ve NP-zor olarak gruplandırmışlardır.

1.2. Sınav Çizelgeleme Problemi

Sınav çizelgeleme üniversitelerdeki zaman alıcı görevlerden birisidir ve uzun zamandan beri araştırmacıların ilgisini çekmektedir. Şeklen gerçekleştirilecek sınavlar kümesinin sınıflar ve zaman dilimlerine atanması olarak tanımlanmaktadır. Bu atama işlemi sırasında çeşitli kısıtlar göz önünde bulundurulmaktadır. Şekil 1.2'de bir sınav çizelgesinin içerik ve ilişkileri verilmiştir. Çizelge 1.2 'de ise örnek bir sınav çizelgesi verilmiştir.



Şekil 1.2 Sınav çizelgesinin içeriği ve ilişkileri

Çizelge 1.2 Örnek sınav çizelgesi

	8-9	9-10	10-11	11-12
Pazartesi	Türkçe Sınıf 1 Hasan İnce	İngilizce Sınıf 4 Hakkı Var		
Salı				

Çizelge 1.2 'de görüleceği gibi hangi sınavın ne zaman yapılacağı ile ilgili bilgiler çizelgelerde verilmektedir. Bu bilgilerin yanı sıra sınav için gerekli kaynaklarda ilgili sınavla alakalı zaman dilimine Şekil 1.2'deki ilişki mantığına uygun şekilde atanmaktadır.

2. KAYNAK ARAŞTIRMASI

2.1. Graf Tabanlı Teknikler

Çizelgeleme problemi ilk defa 1960'larda eğitmenler, sınıflar ve zaman dilimlerinden oluşan üç küme olarak tanımlanmıştır. Çizelgeleme probleminin ilk uygulaması ise Welsh ve Powell tarafından graf boyama problemi üzerine olmuştur (Welsh ve Powell 1967). Detaylı bilgi Qu ve arkadaşlarının hazırlamış oldukları inceleme çalışmasında bulunabilir (Qu ve ark 2006). De Werra (1985) graf boyama problemindeki çözüm yaklaşımını sınav çizelgeleme problemi üzerine modelleyerek sınav çizelgeleme problemi için bir yaklaşım sunmuştur (Qu ve ark 2006). Bir başka çalışmada Ganguli ve Roy (2017) farklı kısıtlama türlerinin bulunduğu çoklu veri kümeleri üzerine graf boyama tekniklerinin uygulanması üzerine çalışmışlardır. Yapmış oldukları çalışmada kısıtlar, derslerin eşit dağılımının, çözümlerin tekillik testinin ve uygun değer çıktısının derecesi üzerinde durmuşlardır. Yaptıkları çalışma da başlangıç aşamasında çakışmalara (komşu düğümlerin aynı rengi alması durumu) izin verilerek graf renklendirilmektedir. Daha sonra ise çakışmalar olmayacak şekilde graf renklendirilerek düzgün bir ders çizelgeleme elde etmeye çalışmışlardır. Saharan ve Kumar (2016) sınav çizelgeleme probleminde kaynak kullanımını graf boyama ile gerçekleştirmişlerdir. Önerdikleri yöntem iki kısımdan oluşmaktadır. Birinci kısımda sınavdaki toplam öğrenci sayısı bütün sınıflardaki mevcut koltuk kapasitesinden büyük olması durumunda, sınavları graf boyama ile elde edilen bağımsız kümeden çıkarmışlardır. İkinci kısımda ise, kullanılan toplam zaman dilimini en aza indirgemek için çıkarılan bu sınavların diğer bağımsız kümelere aktarılmasını sağlamışlardır. Jindal ve Rani (2016) yapmış oldukları çalışmada graf boyama ve onun kullanım alanlarını anlatmışlardır. Malkawi ve arkadaşları (2008) sınav çizelgeleme probleminde graf boyama yaklaşımını kullanmışlardır. Adaletli, kesin ve ideal sınav çizelgesi elde etmeyi hedeflemişlerdir.

2.2. Kısıt Tatmini Teknikleri

Duong ve Lam (2004) Ho Chi Minh City University of Technology üniversitesinin sınav çizelgeleme problemi için tavlama benzetimi yöntemine dayalı bir çözüm sunmuşlardır. Yaptıkları çalışmada başlangıç çözümünü oluştururken kısıt

tatmininden faydalanmışlardır. Çözüme daha hızlı gitmek için ileri ve geri yönlü kontrol mekanizmasını kullanmışlardır (Qu ve ark 2006). Boizumault ve arkadaşları (1996) Angers'teki West Catholic Üniversitesi'nin sınav çizelgeleme problemini kısıt mantık programlama kullanarak çözmüşlerdir. David (1997) sınav çizelgeleme problemini kısıt mantık programlaması kullanarak bir dakikalık sürenin altında çözmeye çalışmıştır. Bu sebeple genel bir arama yöntemi yerine yerel arama teknikleri kullanan bir algoritma kullanmıştır. Goltz ve Matzke (1999) Berlin'deki Humboldt Üniversitesi Charite Tıp Fakültesi için interaktif ve dinamik sınav çizelgeleme hazırlamak için kısıt mantık programlamayı kullanmışlardır

2.3. Yerel Arama Tabanlı Teknikler

Burke ve arkadaşları (2004) hazırlamış oldukları çalışmada algoritmanın elde ettiği son çözümlere iyileştirme yapmaya çalışmışlardır. Bu amaçla arama süreleri girdi parametresi olarak verilen iki yerel arama tekniği çeşidi olan tavlama benzetimi ve büyük tufan (Great Deluge-GD) algoritmalarını kullanmışlardır. Yapmış oldukları deneysel çalışmalarda elde ettikleri sonuçları çeşitli sınav çizelgeleme sonuçları ile karşılaştırmışlardır. Bykov ve Petrovic (2016) çalışmalarında çizelgeleme probleminin çözümü için adım saymalı tepe tırmanma yerel arama algoritmasını geliştirmişlerdir. Bu yöntem, mevcut maliyeti bir dizi ardışık kademe için kabul sınırı olarak kabul eden çok basit bir yöntemdir. Bu adımlama süreçlerini üç farklı şekilde modelleyerek sınav çizelgeleme problemi üzerinde test etmişlerdir. White ve Xie (2000) çalışmalarında OTTABU tabu arama algoritmasını kullanmışlardır. Bulunan çözümlerin kalitesini artırmak için bu sistem yeniliğe dayalı kısa süreli bellek ve hareket (veya frekans) tabanlı uzun vadeli bellek kullanır. Burke ve arkadaşları (2010) hazırlamış oldukları çalışmada, değişken komşuluk arama tekniklerini kullanmışlardır. Basit değişken komşuluk arama yönteminin dışında taraflı (biased) değişken komşuluk algoritması ve onun bir GA ile melezlenmesini önermişlerdir. Çok sayıda komşuluk yapısını incelemişlerdir. Yapılan testlerin sonucunda önerildikleri algoritmanın çok sayıda kıyaslama problem örneklerinde yüksek kalitede çözümlere ulaştığını belirtmişlerdir. Caramia ve arkadaşları (2000) ince ayarlı yerel arama yöntemi geliştirmişlerdir. Önerdikleri algoritmada açgözlü bir yaklaşım ile çizelgelenecek sınavlar olası en düşük sayıdaki zaman dilimlerine atamışlardır. Bununla birlikte bir ceza azaltıcı mekanizmayla zaman dilimi sayısı arttırılmadan çizelge problemini iyileştirmişlerdir.

Herhangi bir iyileştirme yapılamadığında ise zaman dilimi sayısı kademeli olarak bir ceza fonksiyonu ile arttırılmışlardır (Qu ve ark 2006). Casey ve Thompson (2002), Açgözlü Rasgele Uyumlu Arama Prosedürleri (GRASP) yaklaşımını incelemişlerdir. GRASP'da açgözlü bir yöntemle üretilen ilk çözümler yerel en iyi çözüme ulaştıktan sonra yinelemeli yerel arama algoritması daha iyi çözümlerin olup olmadığı yeniden araştırmaktadır. Çalışmada her çevrimde ilk çözüm modifiye bir doyum derecesi (saturation degree) yöntemi ile oluşturulmuştur. Doyum derecesi yönteminde sıralanmış n adet sınavdan bir tanesi çizelgeye atanmıştır. Çalışmanın sonsuz bir döngüye girmesini engellemek için Geri izleme (Backtracking) algoritmasını tabu listesiyle birlikte kullanılmıştır. Geliştirme aşamasında, yüksek başlangıç sıcaklığı ve hızlı soğutma ile Tavlama Benzetimi'nin sınırlı bir formu kullanılmıştır. Kempe zinciri hamleleri, maliyete özellikle katkıda bulunan sınavlarda kullanılmıştır. Ardından sonuçlar çeşitli kıyaslama veri kümeleri üzerinde test edilmiştir (Qu ve ark 2006).

2.4. Popülasyon Tabanlı Algoritmalar

2.4.1. Evrimsel hesaplama tabanlı

Evrimsel hesaplama tabanlı algoritmalar karmaşık ve iyi uyumlu organik yapıların oluşma sürecini kullanarak doğadaki evrim sürecini taklit etmektedir (Back ve ark 1997).

2.4.1.1. Genetik algoritmalar

GA John Holland tarafından canlıların doğal seçilim sürecinden ilham alınarak geliştirilmiş olan bir evrimsel tabanlı algoritmadır (Man ve ark 1996)**.** Corne ve arkadaşları (1994) genel eğitim zaman çizelgelemesinde GA'nın kullanılabilirliği üzerine araştırma yapmışlar ve bununla ilgili gelecek beklentilerine değinmişlerdir. Corne ve arkadaşları (1994)'nın yapmış oldukları çalışmadan elde edilen en önemli çıkarımlardan biri GA'nın graf boyama problemi üzerine doğrudan uygulanması durumunda GA'nın özel olarak oluşturulan bazı graf boyama problemleri ile başa çıkamadığının gösterilmesidir (Qu ve ark 2006). Ross ve arkadaşları (2004) geçiş bölgelerinin çözülebilir çizelgeleme problemlerinde bulunduğunu, farklı bağlantı ve homojenlikteki özel oluşturulmuş graf boyama problemi üzerine uygulayarak göstermişlerdir. Ross ve arkadaşları (2004) yapmış oldukları çalışmanın karmaşık

çizelgeleme problemleri üzerine farklı algoritmaların nasıl bir performans göstereceğini anlamada yardımcı olacağını belirtmişlerdir (Qu ve ark 2006). Sani ve Yabo (2016) makalelerinde daha kolay ve etkin zaman çizelgeleri oluşturmak için GA ile çizelgeler oluşturan yöntemler kullanmışlardır. Abdelhalim ve El Khayat (2016) yapmış oldukları çalışmada çizelgeleme probleminin çözümü için GA tabanlı bir yaklaşım kullanmışlardır. Üzerinde çalıştıkları problemler çok yaygın olmayan zor ve esnek kısıtlar içermektedir. Çizelgeleme problemine yeni bir bakış açısı ile yaklaşmışlardır. Kaynak kullanımının maksimize etmeyi hedeflemişlerdir. GA'da başlangıç popülasyonu rastgele oluşturulmaktadır. Ancak Abdelhalim ve El Khayat (2016) tarafından yapılmış olan çalışmada başlangıç popülasyonu sezgisel bir algoritma kullanılarak oluşturulmuştur. Önerilmiş olan algoritma da öğretim elemanlarının taleplerini karşılayacak ve aynı zamanda çakışmaları (conflict) gidermek amacı ile basit ağırlıklı toplam formülü kullanmışlardır. Boşa harcamaları azaltmak için, öğrenme alanlarının kullanım oranları üzerine odaklanan bir çaprazlama türü ve GA'nın mutasyon operatörü kullanılmıştır. Bununla birlikte algoritmanın etkinliğini arttırmak için yerel bir arama algoritması da kullanmışlardır. Algoritma, alan kullanımı, olaylar arasındaki boşluklar ve günde maksimum ders sayısını dikkate alan bir birleşik uygunluk fonksiyonu kullanmaktadır. Algoritmayı test etmek için Mısır İskenderiye Üniversitesi Ticaret Fakültesi'nden alınan gerçek verilerle büyük bir veri kümesi kullanışlardır. Önerdikleri algoritmayı, literatürde yer alan çözülmesi zor benchmark problemi üzerinde test etmişlerdir. Ayrıca geliştirdikleri algoritmanın deneysel sonuçlarına göre üniversitelerde zaman çizelgeleri ve kaynakları yönetmek için etkin bir yaklaşım olduğunu söylemişlerdir. Algoritmanın benchmark problemlerinin üzerinde oldukça iyi bir performans gösterdiğini belirtmişlerdir. Erden ve arkadaşları (2016) GA kullanarak sınav gözetmen çizelgeleme problemine bir çözüm aramışlardır. Problemi iki aşamadan oluşturmuşlardır. İlk olarak uygun zaman dilimleri ve odalara dersler atanmış ardından gözetmenleri sınavlara atamışlardır. Çalışmayı sınav ve gözetmen atamalarında çakışma olmayan bir çözüm bulmak amacıyla gerçekleştirmişlerdir. Bunu yaparken gözetmenlerin tercihlerini mümkün olduğunca fazla dikkate almışlardır. Türkiye'deki bir devlet üniversitesinin verilerini kullanarak bir olay çalışması oluşturulmuştur. Önerilen GA tabanlı yaklaşımın bu problem üzerinde oldukça etkin bir performans sergilediğini belirtmişlerdir. Dahiya ve arkadaşları (2014) hazırlamış oldukları çalışmada GA'yı sınav çizelgeleme problemi üzerine uygulamışlardır.

Deneysel sonuçlarının başarılı olduğunu ve önerdikleri GA yaklaşımını karmaşık optimizasyon problemlerini etkin bir şekilde çözdüğünü söylemişlerdir**.** Jha (2014) hazırlamış olduğu çalışmada, İbra teknoloji kolejinden elde ettiği gerçek verilere GA uygulayarak probleme çözüm bulmaya çalışmıştır. Kolonias ve arkadaşları (2014) GPU üzerinde çalışan bir hibrit evrimsel algoritma geliştirmişlerdir. Önerilen hibrit evrimsel algoritma bir GA ve açgözlü en dik iniş algoritmalarını birlikte kullanmaktadır. GPU'nun hesaplama kabiliyetleri sayesinde oldukça fazla sayıda bireyden oluşan bir popülasyon kullanabilmişlerdir. Bu da daha geniş bir çözüm uzayında aramayı arttırmaktadır. Böylece optimum çözümün ulaşma şansı arttırılmış olmaktadır. Algoritmayı Toronto veri kümesi üzerine test etmişlerdir ve sonuçlarını diğer algoritmalar ile kıyaslamışlardır. Yaldır ve Baysal (2012) yapmış oldukları çalışmada üniversitelerde belirli dönemlerde manüel olarak hazırlanan sınav takvimi uygulamalarını otomasyon kapsamına aktarmışlardır. Yapmış oldukları çalışma da Evrimsel Hesaplama yöntemini kullanmışlardır. Önermiş oldukları algoritma iki aşamadan oluşmaktadır. Birinci aşamada web ortamında ilgili birimlerden veriler toplanmakta, ikinci bölümde ise geliştirdikleri masaüstü uygulamayı çalıştırılarak sonuçları alınmaktadırlar. Uygulamanın hayata geçirilmesi ile birlikte sınav takvimi hazırlamak için harcanan zamanın minimize edildiğini belirtmişlerdir. Pillay ve Banzhaf (2010) makalelerinde, sınav çizelgeleme probleminin çözümünde GA'ların kullanımını incelemişlerdir. GA iki aşamalı olarak oluşturmuştur. Birinci aşamada çizelgeler oluşturulurken sert kısıtların yerine getirilmesine odaklanılır. İkinci kısımda da oluşturulan bu çizelgeler iyileştirilerek esnek kısıtların maliyetleri düşürülür. Ayrıca sezgisel yöntem biçimindeki alana özel bilgi, evrim sürecine kılavuzluk etmek için kullanılır. Sistem, Carter bencmarks adındaki 13 gerçek dünya probleminden oluşan veri kümesi üzerinde test edilmiştir. Sistemin benchmark üzerindeki performansı diğer evrimsel tekniklerle kıyaslanabilir seviyededir. Hatta bazı durumlarda daha iyi olduğu görülmüştür. Ayrıca, geliştirilen sınav çizelgelerinin kalitesi alanda üretilen en iyi sonuçların arasındadır. Arogundade ve arkadaşları (2010) Nijerya Abeokutadaki Tarım Üniversitesinden elde ettikleri gerçek dünya sınav çizelgeleme verisi üzerinde GA'yı kullanarak çizelgeleme yapan bir yöntem sunmuşlardır. Sınav dönemi boyunca sınavları dağıtmaya yönelik yeni bir hedef fonksiyon kullanmışlardır. Bahsedilen yöntem gerçek verilerle test edilmiş ve sonuçları tartışılmıştır. Cupic ve arkadaşları (2009) üniversitelerinin Bologna sürecine girmesiyle karşılaştıkları çeşitli organizasyonel

problemlere ve çeşitli zaman ayarlama problemleri üzerine oluşturdukları sınav çizelgeleme modelini önermişlerdir. Mansour ve Timany (2007) hazırladıkları makalede değiştirilmiş ağırlıklı graf boyama problemi formülünü ve iki stokastik arama algoritmasını sınav çizelgelemede kullanmışlardır. Kullanılan stokastik arama algoritmaları tavlama benzetimi ve GA'lardır. Tavlama benzetimi prosedürlerini kullanarak FESP(Final Sınavı Çizelgeleme Paketi) adındaki bir "iyi" kümeleme tabanlı sezgisel prosedürüne iyileştirme önermişlerdir. Sezgisel prosedüre FESP-SA adını vermişlerdir. Ampirik olarak önerilen yöntemle FESP'i gerçek veri kullanarak karşılaştırmışlardır. Yapılan deney sonuçlarına göre GA ve tavlama benzetimi FESP'e göre daha iyi sınav çizelgeleri üretmektedir. Ayrıca GA ve tavlama benzetimi herhangi bir can sıkıcı çatışmaya veya adaletsizliğe sebep olmadan sınav günlerinin sayısını azaltmıştır.

2.4.1.2. Memetik algoritma

GA'lar biyolojik evrimden ilham alırken memetik algoritmalar (MA) kültürel evrimi taklit etmeye çalışmışlardır. Moscato, popülasyon tabanlı yaklaşımların başarısında ilişkili yüzeylerin (landspace) önemini fark etmiştir (Moscato 1989). Al-Betar ve arkadaşları (2014) hazırlamış oldukları makalede MA'ların ana öğelerinin kombinasyonlarının kapasitesiz sınav zaman çizelgelemesi problemlerinin çözümlerinin kalitesi üzerindeki etkinliğini araştırmışlardır. Bu öğeler rekombinasyon, rastgelelik ve komşuluk yapılarıdır. MA'nın bir türevi olan Harmoni araması algoritması, bu öğelerin farklı kombinasyonlarının çalıştırılması için kullanılmıştır. Bu yöntemin üç ayrı bileşeni vardır. Bunlar rekombinasyon kullanarak bellek değerlendirilmesi, rastgelelik kullanılarak rastgele değerlendirme ve komşuluk yapısı kullanarak saha ayarlamasıdır. Bu MA'ların 17 farklı senaryo için kombinasyonları değerlendirilmiştir. Daha sonra sonuçlar kıyaslama veri kümeleri üzerinde değerlendirilmiştir.

2.4.2 Sürü zekası tabanlı optimizasyon algoritmaları

Sürü zekası tabanlı optimizasyon algoritmaları birbiriyle etkileşim halinde olan ajanlar grubu veya kendi kendine organize olan sürüleri modelleyen algoritmalardır. Burada bu algoritmalardan PSO ve YAKA'dan bahsedilmiştir (Karaboga ve Basturk 2007).

2.4.2.1. Parçacık sürü optimizasyonu algoritması

PSO, doğada balık ve kuş gibi büyük gruplar halinde yaşayan hayvanların sürü davranışlarına dayalı olarak 1995 yılında Kennedy ve Eberhart tarafından geliştirilmiş bir sürü zekâsı algoritmasıdır (Yang 2010). Shiau (2011) makalesinde ders çizelgeleme problemi için PSO ilkelerine dayanan yeni bir meta-sezgisel algoritma önermiştir. Algoritma parçacık için mutlak pozisyon değeri gösterimi ve tasarımı, istedikleri gün ve zaman periyodunda ders vermek, vermek istedikleri derslerle alakalı esnek tercihler yapmalarına izin vermek gibi bazı özellikler içermektedir. Tüm uygulanabilir olmayan zaman çizelgeleri için onarım işlevine sahiptir. Ayrıca orijinal PSO algoritmasında parçacıklar çözümleri sürekli bir çözüm uzayında araştırmaktadır. Ders çizelgeleme probleminin çözüm uzayı ayrık olduğundan, daha iyi bir çözüm iyileştirme keşfi için önerilen bir PSO'ya yerel bir arama mekanizması da dahil edilmiştir. Algoritmalar, Tayvan'daki bir üniversiteden çizelgeleme verileri kullanılarak test edilmiştir. Deneysel sonuçlar hibrit algoritmanın etkili bir çözüm ürettiğini göstermiştir. Algoritma literatürdeki diğer GA'lardan daha iyi bir performans göstermiştir. Chu ve ark. (2006) katılmış oldukları konferansta sınav çizelgelemede PSO algoritmasını kullanmışlardır. Klasik PSO'nun sürekli optimizasyon problemlerinde çeşitli başarılı uygulamaları vardır. Bu makalenin en büyük katkısı ise ayrık bir problem olan zaman çizelgelemesinin PSO kullanılarak çözülmesidir. Deney sonuçları PSO'nun ayrık problemlerin de çözümünde kullanılabileceğini göstermiştir. Irene ve ark. (2009) sürekli optimizasyon problemlerinin çözümünde kullanılan PSO algoritmasını sınav çizelgelemede kullanmışlardır. Çalışmalarında PSO'nun ayrık bir problem olan sınav çizelgelemede ümit verici sonuçlar sergilediğini göstermişlerdir. Ghatei ve arkadaşları (2012) hazırlamış oldukları makalede PSO ve GD algoritmasını birleştirmiştir. Yöntemde PSO'nun global arama karakteriyle GD algoritmasının yerel arama faktörü sırasıyla kullanılmıştır. Öncelikle PSO ile ortam araması yapılmıştır. Daha sonra PSO'nun sonuçları GD algoritmasına verilerek alınan sonuçların daha kesin olarak aranması sağlanmıştır. Bu yaklaşım klasik PSO ve GA ile kıyaslanmıştır. Elde edilen sonuçlar dikkate değer ve istikrarlıdır. Ahandani ve arkadaşları (2012) hazırlamış oldukları makalede, ayrık PSO ile sınav çizelgeleme problemi üzerinde durmuşlardır. Ayrık PSO'daki parçacıkların konumu mutasyon, uzman rekombinasyon operatörü ve graf renklendirme sezgisellerinin bir birleşimi kullanılarak güncellenmiştir. İki aşamalı tepe tırmanışı adı verilen yeni bir yerel arama yöntemi teklif edilmiş ve ayrık PSO

algoritmasının melezleştirilmesinde kullanılmıştır. Sonuçlar literatürdeki diğer yöntemler ile kıyaslanmıştır.

2.4.2.2. Yapay arı kolonisi algoritması

YAKA Derviş Karaboğa tarafından doğada bal arılarının yem arama davranışlarından ilham alınarak geliştirilmiş bir sürü zekâsı algoritmasıdır (Gao ve Liu 2012). Alzaqebah ve Abdullah (2011) hazırlamış oldukları makalede YAKA'nın sınav çizelgeleme problemlerinin çözümünde nasıl kullanılacağını araştırmışlardır. Algoritmada üç kategorideki arılar kullanılmıştır. bunlar işçi arı, gözcü arı, kâşif arıdır. Sonuçlar literatürde sıklıkla kullanılan veri kümelerinde test edilmiştir. Elde edilen deney sonuçları teklif edilen yöntemin umut verici sonuçlar ürettiğini göstermiştir. Wen ve Bin Asmuni (2013) sınav çizelgeleme probleminin çözümünü YAKA kullanarak oluşturmuşlardır. Geleneksel YAKA'nın yüksek keşif yeteneği onun yakınsama hızında bir yavaşlamaya sebep olmaktadır. PSO'dan ilham alarak Global görüşlü YAKA (Global Best Concept – Artificial Bee Colony Algorithm) adı verilen yöntem teklif edilmiştir. Algoritma global en iyi konseptine göre formülize edilmiştir. İşçi arı safhasında global en iyi çözüm bilgisi çözümlere dahil edilmektedir. Bunun amacı arama sürecinin global en iyiye doğru yönlendirilmesidir. Ayrıca sömürme kabiliyetini arttırmak adına yerel arama metodu da gözcü arı safhasına dâhil edilmiştir. Yapılan deney sonuçlarına göre önerilen yöntem klasik YAKA'dan oldukça iyidir. Ayrıca sonuçlar literatürdeki diğer yöntemlerle kıyaslanabilecek seviyededir. Bolaji ve arkadaşları (2012) hazırlamış oldukları makalede YAKA'yı sınav çizelgelemesi probleminin çözümünde kullanmışlardır. Carter (1996) defacto veri kümesini kullanmışlardır. YAKA sınav çizelgeleme probleminde çalışabilmek için işçi arı ve gözcü arı safhasında kullanılmak üzere üç komşuluk yapısı içermektedir. Bunlar ilerletme, takas etme ve Kempe zinciri yöntemidir. Bu operatörlerin algoritmaya etkisi deneyler yapılarak gözlemlenmiştir. Elde edilen sonuçlar literatürdeki diğer yöntemlerle rekabet edebilecek düzeydedir. Alzaqebah ve Abdullah (2014) hazırlamış oldukları makalede sınav çizelgelemede uyarlamalı bir YAKA kullanmışlardır. Algoritmada popülasyondaki çeşitliliği arttırmak ve erken yakınsamadan kaçınmak için bozucu seçim stratejisi benimsenmiştir. Yerel yoğunlaşma kapasitesini arttırmak için de kendinden uyarlamalı komşu strateji benimsenmiştir. Son olarak uyarlanmış YAKA tepe tırmanma algoritması ile melezlenmiştir. Elde edilen sonuçlar oldukça iyidir. Fong

ve arkadaşları (2015) hazırlamış oldukları makalede üniversite ders ve sınav çizelgeleme problemlerini hibrit bir YAKA kullanarak çözmüştür. YAKA PSO'dan ilham alınarak global en iyi modeline göre geliştirilmiştir. Ayrıca yerel keşif kapasitesini arttırmak için GD algoritması ile melezlemişlerdir. Ayrıca bir yerel arama yaklaşımı GD algoritmasıyla birleştirilmiştir. Önerilen yöntemin vermiş olduğu sonuçlar oldukça kalitelidir.

2.5. Çoklu Kriterli Teknikler

Colijn and Layfield (1995) Calgary Üniversitesi'ndeki sınav çizelgeleme problemi için çok aşamalı bir yaklaşım uygulamışlardır. Birinci aşamada, bireysel sınavlar ve zaman dilimlerindeki sınavların tümü, iki sınavı arka arkaya giren öğrencilerin sayısını azaltmak için taşınmıştır. İkinci aşamada ise üst üste üç ve dört sınava giren öğrenciler benzer bir süreç kullanarak değerlendirilmiştir. Yazarlar ikinci aşamadaki beklenmeyen durumlarda zaman çizelgesinin değiştirilmesi durumunu da göz önünde bulundurmuşlardır. Bu aşama görsel bir arayüzle sınavların çizelgelerden taşındığı, silindiği veya kaldırıldığı oldukça etkileşimli bir süreçtir (Qu ve ark 2006).

Burke ve arkadaşları (2000), sınav çizelgeleme problemlerindeki dokuz kriter ile ilgili iki aşamalı çok kriterli bir yaklaşım geliştirmişlerdir (örneğin, oda kapasitesi, sınavların yakınlığı, sınavların zamanı ve sırası vb.). 1. aşamada, doygunluk derecesi (Saturation Degree) uygun çözümler kümesi oluşturmak için kullanılmıştır, burada her kriter ayrı ayrı ele alınmıştır. İkinci aşamada bu çözümler aynı anda sezgisel olarak geliştirilmiştir (Qu ve ark 2006).

2.6. Üst Sezgiseller(Hyper Heuristics)

Kendall ve Hussin (2005) çalışmalarında hareket stratejileri ve graf sezgisellerinin düşük seviye sezgisel olarak kullanıldığı bir Tabu arama sezgiselini incelemişlerdir. Algoritma MARA Teknoloji üniversitesi sınav çizelgeleme problemlerinde başarıyla uygulanmıştır (Qu ve ark 2006).

Ross ve arkadaşları (2004) çözümleri oluşturmak için basitleştirilmiş problem durum tanımları uzayında arama yapan genel sabit durumlu bir GA geliştirmişlerdir. GA'da arama, gerçek çözümler yerine sezgiseller üzerinden gerçekleştirilmiştir. Üç farklı uygunluk fonksiyonu test edilmiştir (Qu ve ark 2006).

Bilgin ve arkadaşları (2006) yedi sezgisel seçim yöntemi ve beş kabul kriterini içeren bir üst sezgiselle sınav çizelgeleme ve sınama fonksiyonları üzerinde ampirik bir çalışma yaparak analiz etmişlerdir (Qu ve ark 2006).

Ersoy ve arkadaşları (2007) , memetik bir algoritma içinde farklı sıralamalarda üç tepe tırmanma yönteminin uygulandığı bir üst sezgisel yaklaşım üzerinde çalışmışlardır. MA esnasında bireyler Toronto veri kümesindeki her bir kısıt türünün ihlalinin kaydının tutmuşlardır. Yöntem kendinden uyarlamalı MA ile karşılaştırılmıştır ve test edilen yöntemler arasında o güne kadar ki en iyi performansı göstermiştir (Qu ve ark 2006).

Qu ve arkadaşları (2015) yazdıkları makalede tahmin dağılım algoritmalarına dayalı melez üst sezgisel bir yaklaşım uygulayan bir yöntem sunmuşlardır. Ana amaçları arama yöntemlerinin genellenebilirlik seviyesini arttırmaktır. Üst sezgisellerinin amacı ise bir çok optimizasyon problemi için kabul edilebilir kalitede çözümler üretmektir. Bu amaçla genellenebilirliği test etmek için farklı türdeki sınav çizelgeleme problemleri üzerinde testler yapmışlardır. Üst sezgisel yalnızca alansal olmayan (non-domain) bilgiye dayalı olarak verilen düşük seviye sezgiseller arasından sezgisel seçimi araması yapan otomatik kurma yöntemi kullanmıştır. Üst düzey arama metodolojisi basit bir tahmin dağıtım algoritmasına dayanmaktadır. Farklı problem çözme durumlarında uygun sezgisel yöntemleri seçerek araştırmaya yön verebilme yeteneğine sahiptir.

Burke ve arkadaşları (2014) makalelerinde, zaman çizelgelemelerini iyileştirmek için düşük seviye sezgisel hareketleri melezleyen üst sezgisel bir yaklaşım sunmuşlardır. Zaman çizelgesinde yumuşak kısıt ihlaline neden olan sınavlar daha iyi bir zaman çizelgesi üretmek üzere sıralanmışlar ve yeniden çizelgelenmişlerdir. Sınavların yeniden çizelgelemelerinde sıralamanın ve sezgisel hamlelerin ikisinin de üretilen çözümün kalitesini etkileyerek çizelgelemeleri iyileştirdiği görülmüştür. Üst sezgisel melezlemelerin farklı kombinasyonları test edildikten sonra en iyi performansı Kempe zinciri hamle sezgiseli ve zaman dilimi takaslama sezgiselinin gösterdiği ispatladıklarını söylemişlerdir. Benzer şekilde, Doyma Derecesi kullanılarak sınavların düzenlenmesinin ve En Büyük Ağırlıklı Derece yöntemi kullanarak herhangi bir bağın koparılmasının en iyi sonuçları verdiğini göstermişlerdir.

3. MATERYAL VE YÖNTEM

3.1. Genetik Algoritmalar

1950'lerden başlayarak birçok bilgisayar uzmanı birbirinden bağımsız olarak evrimin bir optimizasyon aracı olarak mühendislik problemlerinde kullanımını araştırmışlardır (Mitchell 1998).

GA dışında çeşitli araştırmacılar tarafından farklı evrimsel yöntemler teklif edilmiştir. Örnek olarak Rechenberg 1960'larda evrim stratejileri adını verdiği bir yöntemi uçak kanadı gibi reel değerli cihazların optimizasyonunda kullanmıştır. Daha sonra Schwefel tarafından bu fikir daha da geliştirilmiştir. Ayrıca Fogel, Owens ve Walsh Evrimsel programlama adını verdikleri bir yöntem teklif etmişlerdir (Mitchell 1998).

GA, John Holland, öğrencileri ve iş arkadaşları tarafından 1960'larda Michigan Üniversitesinde icat edilmiştir. Evrimsel programlama ve evrim stratejileri yöntemlerinin aksine Holland'ın amacı belirli bir problemi çözmek amacıyla algoritma tasarlamak değildi. Bundan ziyade doğada rastladığımız doğal uyarlanım (adaptasyon) mekanizmasının bilgisayar sistemlerine aktarılmasını sağlamaktır. Holland 1975'te çıkardığı "Adaptation in Natural and Artificial Systems" adlı kitabında biyolojik evrimin soyutlaması olan GA'yı ve onun iskeletini sunmuştur (Mitchell 1998).

Holland'ın yapmış olduğu bu çalışma daha sonra öğrencisi olan Goldberg'in tezinde GA'yı kullanarak çözülmesi zor bir problem olan gaz boru hattının kontrolünü gerçekleştirmesiyle popülaritesini arttırmıştır (Goldberg 1989). Goldberg'in bu çalışmasından sonra GA farklı alanlarda birçok problemin çözülmesinde kullanılmıştır.

GA doğadaki evrim sürecinin bilgisayar ortamına uyarlamasıdır. İlk olarak Darwin tarafından teklif edilen evrim teorisinde Darwin doğal seçilim adını verdiği süreci kuramlaştırmıştır. Ona göre türdeşlerine göre daha güçlü ve işe yarar özelliklere sahip olan canlılar hayatta kalma yarışında daha avantajlı oluyorlardı. Bu nedenle soylarını devam ettirebiliyorlardı. Ayrıca yapmış olduğu gözlemlerde yeni türlerin zamanla hâlihazırdaki bir türden yavaşça farklılaştığı sonucuna da varmıştır (mutasyon). İşte bu biyolojik evrimsel süreç GA'nın ilham kaynağı olmuştur.

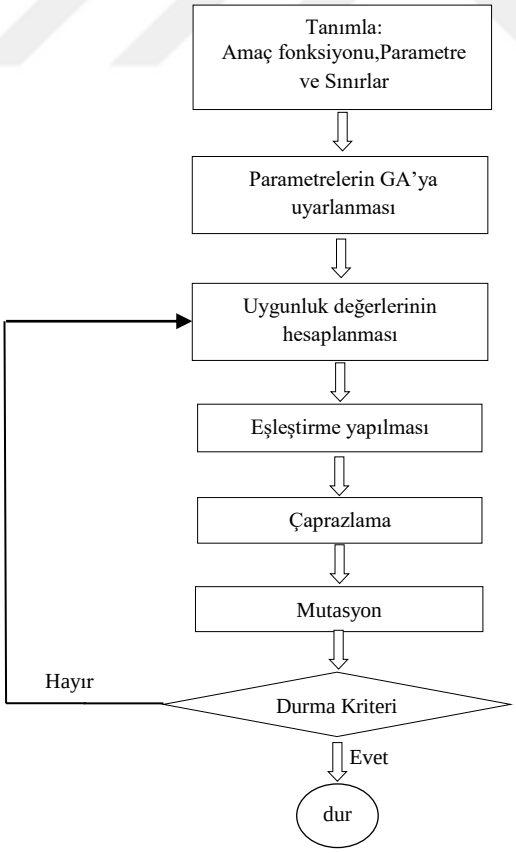
Doğada canlı organizmalar hücrelerden oluşur. Her bir hücre kromozom adı verilen canlının özelliklerinin planını içeren DNA dizilerine sahiptir. Kromozomlar da gen adı verilen alt birimlere bölünebilir. Bunlar her bir özelliği tanımlayan parçacıklar

olarak düşünülebilir (örnek göz rengi vs.). Bu genlerin aldığı her bir farklı değer allel olarak ifade edilir. Her bir genin kromozomdaki konumuna (pozisyon) lokus adı verilir.

Üreme esnasında bu genetik bilgiler yarısının anneden yarısının babadan alınması sonucu yeni oğul bireyler oluşur. Ayrıca mutasyon adı verilen bir süreçle genetik bilgilerde değişim gözlenir.

GA'da kromozom genelde bir aday çözümü ifade eder ve çoğunlukla bit dizileri şeklinde temsil edilir. Genler ise çözümü oluşturan tek bir biti veya bit bloklarını ifade etmektedir. Allel ise bir lokustaki genin alabileceği farklı değerlerdir. Örnek olarak bir bit dizisinde 0 veya 1 değerlerini alabilirler. Çaprazlama bu değerlerin takaslanması ile gerçekleştirilir.

Basit bir GA'nı akış diyagramı Şekil 3.1'deki gibidir. İlk popülasyon ile başlanır (rastgele oluşturulabilir veya diğer buluşsal yöntemlerle tohumlanmış olabilir), çiftleşmek için bu popülasyondan ebeveynler seçilir. Yeni yavrular üretmek için ebeveynlere çaprazlama ve mutasyon operatörleri uygulanır. Ve nihayet bu yavrular popülâsyondaki mevcut bireylerin yerini alır ve süreç tekrar eder. Bu şekilde GA canlıların evrim sürecini taklit etmeye çalışılır.



Şekil 3.1 Genetik algoritmanın genel akış diyagramı

Evrimsel hesaplama toplumunda GA'yı diğer evrimsel hesaplama yöntemlerinden farklı kılan, herkesin kabul ettiği kesin bir tanımının bulunmamasıdır. Ancak genelde GA'nın çoğunluğu kromozom popülasyonları, uygunluğa göre seçim, yeni yavru üretmek için çaprazlama ve yeni yavruların rastgele mutasyonu genellikle kullanılmaktadır. Holland'ın dördüncü öğesi olan ters çevirme bugünün uygulamalarında nadiren kullanılır ve varsa bile avantajları iyi belirlenememiştir (Mitchell 1998).

Bir GA popülasyonundaki kromozomlar tipik olarak bit dizeleri biçimindedir. Kromozomdaki her bir lokus 0 veya 1 olmak üzere iki muhtemel allel içerir. Her kromozom, aday çözümlerin araştırma uzayında bir nokta olarak düşünülebilir. GA, kromozom popülâsyonlarını sırasıyla bir popülâsyonu diğeriyle değiştirerek işler. GA genellikle, mevcut popülâsyondaki her kromozoma bir puan (uygunluk) atayan bir uygunluk fonksiyonu gerektirir. Bir kromozomun uygunluğu, o kromozomun eldeki problemi ne kadar iyi çözdüğüne bağlıdır (Mitchell 1998).

3.1.1. Kromozom gösterimi ve amaç fonksiyonu

Amaç fonksiyonları, giriş değişkenleri olarak alınan kromozomlara karşılık çıktı üreten fonksiyonlardır. Burada amaç, giriş değişkenleri için uygun değerleri bularak çıktıyı istenilen şekilde değiştirmektir (Haupt ve Haupt 2004).

GA, kromozomun veya optimize edilecek değişken dizilerinin tanımlanmasıyla başlar. Eğer kromozomun Ndeg adet değişkeni varsa (Ndeg boyutlu optimizasyon problemi) P_1, P_2, P_3,.....P_{Ndeg} olmak üzere (Haupt ve Haupt 2004);

Kromozom = [P_1, P_2, P_3,.....P_{Ndeg}] olarak tanımlanır.

Örneğin; topografik bir haritayı ele alalım. Amaç fonksiyonu; giriş değişkenleri, enlem (x) ve boylam (y) olarak kabul edilirse,

kromozom = [x,y] şeklinde tanımlanır. Burada N_{deg} = 2'dir.

Her bir kromozom için amaç fonksiyonu P1 ve P2 girdi değişkenlerinin değerlerine göre hesaplanır.

F(kromozom) = F(P_1,P_2,.....,P_{Ndeg})

3.1.2. Değişkenlerin tanımlanması ve kodlama

Bir GA'yı herhangi bir probleme uygularken verilmesi gereken en önemli kararlardan biri, çözümleri göstermek için kullanılacak temsili belirlemektir. Uygunsuz gösterimin GA'nın kötü performans göstermesine neden olabileceği gözlemlenmiştir.

GA'lar için en sık kullanılan temsillerden bazıları aşağıda verilmiştir. Değişkenler, temsil problemine oldukça bağlıdır ve bu temsillerin dışında başka temsillerin de tanımlanması gerekebilir.

İkili gösterim, GA'larda en basit ve en yaygın kullanılan gösterimlerden biridir. Bu tür gösterimde kromozom, bit dizelerinden oluşur. Çözüm uzayı boolean karar değişkenlerinden yani evet veya hayır'lardan oluşuyorsa, bu problemler için ikili gösterim kolay uygulanabilir olur. Genler tek bir bitten oluşabileceği gibi birden fazla bit de içerebilir. Şekil 3.2'de tek bitlik genlerden oluşan bir kromozomun temsili gösterilmiştir.

1	1	0	1	0	0	1	1	0

Şekil 3.2 Örnek bir kromozom temsili

Ayrık değerler için çözüm uzayı, her zaman evet ve hayırlarla sınırlanamaz. Örnek olarak Kuzey, Güney, Doğu, Batı şeklinde değerler alan bir problem için kromozom **0, 1, 2** ve **3** şeklinde tam sayı olarak temsil edilebilir. Şekil 3.3'te tam sayı değerler içeren bir kromozom örneği gösterilmiştir.

6	7	8	9	5	4	3	2	1

Şekil 3.3 Tam sayı ile örnek bir kromozom temsili

Yukarıda bahsettiğimiz gösterimler dışında reel sayı ve permutasyon gösterimi gibi türler de mevcuttur

3.1.3. Popülasyonun İlklendirilmesi

Genellikle aday çözümlerin ilk popülasyonu arama uzayı boyunca rastgele oluşturulur. Bununla birlikte, alana özgü bilgi veya diğer bilgiler kolayca popülasyona dahil edilebilir. Bu seçim genellikle N×M şeklinde bir matris olarak yapılır. N popülasyon sayısını gösterirken; M kromozomdaki bit sayısını ifade eder.

3.1.4. Doğal seçim

Seçim, çiftleşecek ve yeniden birleştirilecek olan anne-babaları seçme sürecidir. Seçim, biyolojik sistemlerde bulunan doğal seçime benzer bir şekilde popülasyona baskı uygulamaya çalışmaktadır. Düşük performanslı bireyler elenirken; daha iyi performans gösteren veya daha uygun bireyler, gelecek kuşaklarda içerdikleri bilgileri daha iyi bir şekilde yaygınlaştırma şansına sahiptirler (Coley 1999).

GA'larda kullanılan çok çeşitli sayıda seçim yöntemi vardır bunlardan bazılar aşağıdaki gibidir.

Üstten alta doğru eşleştirme yönteminde, listenin başından başlanarak eşleşme işlemi için en üstteki $N_{tutulacak}$ kromozom seçilir ve kromozomlar ikili olarak eşleştirilir. Böylece, çift sıralar tek sıralar ile eşleştirilir. Yani anneler için 1, 3, 5,… şeklinde seçim yapılırken; babalar 2, 4, 5 şeklinde seçilir. Bu yöntem çok fazla tercih edilmemektedir (Haupt ve Haupt 2004).

Rastgele seçim stratejisinde ise mevcut popülasyondan rastgele ebeveynler seçilir. Üsten alta eşleştirme yönteminde olduğu gibi daha uygun bireylere yönelik herhangi bir seçim baskısı bulunmadığından bu strateji genellikle tercih edilmemektedir

Ağırlıklı rastgele seçim, eşleşme havuzundaki kromozomların olasılıkları maliyetleriyle ters orantılıdır. En düşük maliyetli kromozomun, çiftleşme olasılığı en yüksek iken, en yüksek maliyetli kromozomun eşleşme olasılığı en düşüktür (Haupt ve Haupt 2004).

Turnuva seçiminde, popülasyondan K adet birey rastgele seçilir ve bunlardan en iyisi ebeveyn olarak işaretlenir. Bir sonraki ebeveyn seçimi için de aynı işlemler tekrarlanır. Turnuva Seçimi negatif uygunluk değerleri için de çalışabildiğinden literatürde son derece popülerdir.

Rulet tekerleği seçim yönteminde, önce toplam uygunluk değeri hesaplanır. Kromozomlar, toplam uygunluk değerine bölünerek her bir birey için 0 ve 1 arasında

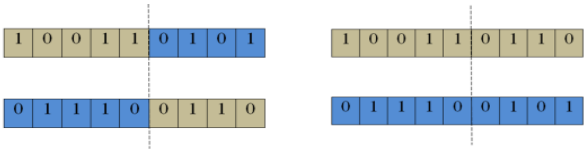
değişen seçim olasılıkları bulunur. Daha sonra, kümülatif olasılıkları hesaplanır. Popülasyon sayısı kadar rastgele 0 ve 1 arasında sayılar üretilir. Üretilen rastgele sayı, birinci kromozomun kümülatif seçim ihtimalinden küçük ise, birinci kromozom seçilir. Eğer değilse, ikinci kromozomun veya diğerlerinin kümülatif ihtimalleriyle karşılaştırılarak hangisinden küçükse o kromozom seçilir. Böylece Rulet seçim metodu gerçekleştirilmiş olur. Literatürde bu yöntemlerin dışında çok sayıda seçim yöntemi bulunmaktadır.

3.1. 5. Çaprazlama

Seçimden sonra, çiftleşme havuzundaki bireyler, yeni ve daha iyi yavrular oluşturulması amacıyla çaprazlanır (birleştirilir). Literatürde çok sayıda çaprazlama operatörü bulunmaktadır. Burada bunlardan bir kaçına değinilecektir (Sastry ve ark 2014).

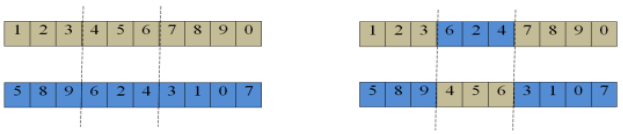
Birçok çaprazlama operatöründe iki birey rastgele seçilir ve çapraz olasılığı olarak adlandırılan p_c olasılığıyla çaprazlanır. Daha sonra r sayısı rastgele üretilir. Eğer $r \leq p_c$ ise, rastgele seçilen iki birey çaprazlamaya tabi tutulur. Çaprazlama olasılığı temsil eden p_c'nin değer seçimi, deneysel olarak ya da şema teoremine bağlı olarak gerçekleştirilebilir (Sastry ve ark 2014).

Temel olarak, çaprazlama, iki ebeveynin kromozomları arasındaki gen değişimi anlamına gelir. En basit durum olan tek noktalı çaprazlamada, iki string dizisini rastgele seçilmiş bir noktadan kesip kuyruk kısımlarını değiştirerek bu işlem gerçekleştirilmektedir. Şekil 3.4'te bu süreç görselleştirilmiştir (Bodenhofer 2003).



Şekil 3.4 Tek noktalı çaprazlama

Aşağıda literatürde kullanılan bazı çaprazlama operatörleri gösterilmiştir. Çok noktalı çaprazlama, tek noktalı çaprazlamanın genelleştirilmiş halidir ve burada, yeni yavruların oluşturulması için alternatif kısımlar değiştirilir. Şekil 3.5'te çok noktalı çaprazlama örneği gösterilmiştir.



Şekil 3.5 Çok noktalı çaprazlama

Tekdüze (Uniform) çaprazlamada, kromozom parçalara bölünmez, bunun yerine her gen ayrı ayrı işlenir. Burada, her bir kromozomun yeni bireye dahil edilip edilmeyeceğine karar vermek için yazı tura atılır. Bir ebeveynden çocuğa daha fazla genetik materyal alabilmek için yazı tura işlemi meyilli (bias) gerçekleştirebilir. Şekil 3.6'da örnek bir tekdüze çaprazlama örneği gösterilmiştir.



Şekil 3.6 Tekdüze çaprazlama

Literatürde yukarıdaki çaprazlama yöntemlerinin dışında kısmi planlı çaprazlama (Partially Mapped Crossover - PMX), sıra tabanlı çaprazlama (Order based crossover - OX2), karıştırma çaprazlaması (Shuffle Crossover) ve benzeri birçok operatör bulunmaktadır.

3.1. 6. Mutasyon

GA'nın son bileşeni mutasyon'dur. Mutasyon, bir kişinin genetik bilgisinin radyoaktif radyasyon veya diğer çevresel etkiler yoluyla rastgele deformasyonudur. Gerçek üreme işleminde, belirli bir genin mutasyon geçirme olasılığı, tüm genler ile neredeyse eşittir. Tek bir genin mutasyon geçirme olasılığı p_m'dir. Herhangi bir gen için rastgele bir sayı seçilir ve bu sayı p_m'den küçükse gen mutasyona tabi tutulur. Mutasyon operatörü, popülasyondaki çeşitliliği korumak için kullanılır ve genellikle düşük olasılık değeri ile uygulanır.

Mutasyon işlemi için çok sayıda farklı operatör kullanılmaktadır. Bit tersleme mutasyonu yönteminde, bir veya daha fazla rastgele seçilen bit ters çevrilir. Bu yöntem, ikili olarak kodlanmış GA'lar için kullanılır. Şekil 3.7'de bit tersleme örneği gösterilmiştir.



Şekil 3.7 Bit tersleme mutasyonu

Rastgele Sıfırlama yöntemi, tamsayı temsili için bit çeviricisinin eşdeğeridir. Burada, rastgele seçilen bir gene izin verilen değerlerler arasından rastgele bir değer atanır.

Takas mutasyonunda ise, kromozom üzerinde rastgele olarak iki nokta seçilir ve değerler değiştirilir. Permütasyon kodlama ile gösterim yapılan problemlerde bu yöntemin kullanımı yaygındır. Şekil 3.8'de Takas mutasyonu örneği gösterilmiştir.



Şekil 3.8 Takas mutasyonu

Burada anlatılan mutasyon operatörlerinin dışında literatürde çok sayıda farklı mutasyon operatörü bulunmaktadır. Bu operatörlerden hangisinin kullanılacağı problemden probleme farklılık göstermektedir.

3.1. 7. Sonlandırma koşulu

Bir GA'nın sonlandırma koşulu, GA çalışmasının ne zaman sona ereceğini belirlemede önemlidir. Başlangıçta GA'nın ilk birkaç çevrimde daha iyi çözümlere çok daha hızlı ilerlediği, ancak sonraki aşamalarda doymuş olma eğiliminde iyileştirmelerin çok küçük olduğu gözlemlenmiştir. Genelde sonlandırma koşulu, çalıştırmanın sonunda çözümümüzün optimuma yakın olması için istenir. Çeşitli sonlandırma koşulları vardır. Belirli bir çevrim (iterasyon) adedine erişince sonlandırma, belirli sayıdaki yineleme boyunca iyileştirme olmadığında sonlandırma ve önceden belirlenmiş bir amaç fonksiyonu değerine ulaşınca sonlandırma gibi sonlandırma koşulları mevcuttur.

3.2. Parçacık Sürü Optimizasyonu (PSO)

Sürü örgütlenmesi için görme en önemli duyudur (Grosan ve ark 2006). Kuşların farklı yapılardaki gözleri onların her iki taraftaki nesneleri aynı anda görmelerine izin verir. Kuşların diğer hayvan gruplarına göre daha büyük göz boyutlarına sahip olmaları hayvan popülasyonunda en yüksek seviyede görme duyusuna sahip hayvanlardan biri olmalarının da bir nedenidir (MacKinnon ve ark 2001). Kuşlar genellikle yiyeceklerden etkilenirler. Onların uzun mesafe göç ve yiyecek aramak için etkileyici eş zamanlı sürü yetenekleri bulunmaktadır. Kuşların etkili sürü iletişimleri onların uçarken çarpışmadan sıkça ani yön değiştirmelerine, dış tehditlere tepki verirken; hızlı dağılmalarına, toplanmalarına ve yırtıcılardan kaçabilmelerine imkân verir (Ahmed ve Glasgow 2012).

Kuş sürüsü davranışı, ortak bir grup hedefi olan çok sayıda etkileşim halindeki kuşun sürü halindeki ortak hareket davranışları olarak tanımlanabilir. Kuşlar arasındaki (parçacıklar) yerel etkileşimler genelde sürünün ortak hareket yönünde ortaya çıkar. Bu etkileşimler en yakın komşu prensibi adı verilen herhangi bir merkezi koordinasyondan bağımsız, hareketlerini yalnızca en yakın komşularına göre ayarladıkları belirli sürü kurallarını belirtir. Kuşların bu sürü davranışları bilgisayarda ilk olarak Craig Reynolds tarafından 1987'da simüle edilmiştir (Reynolds 1987). Yazar, öncü çalışmasında kuşların sürü davranışını simüle etmek için üç basit sürü kuralı önermiştir. Bunlar: **sürü merkezleme** (sürü üyeleri yakındaki sürü arkadaşlarının merkezlerine yakın tutan bir yönde uçarak diğer sürü üyelerinin yakınında kalmaya çalışmaktadır), **çarpışmadan kaçınma** (sürü üyeleri, yakındaki sürü arkadaşlarının bağıl konumlarına bakarak çarpışmalardan kaçınır) ve **hız eşleme** (sürü üyeleri hızını yakındaki sürü arkadaşlarıyla eş tutmaya çalışırlar) kurallarıdır (Reynolds 1987, Ahmed ve Glasgow 2012).

Bahsi geçen sürü davranışlarının kuralları kolay gibi algılansa da ayrı kuşlardan oluşmasına rağmen tüm hareket akıcı görünümüyle görsel olarak oldukça karmaşıktır (Reynolds 1987). Burada dikkat edilmelidir ki çarpışma önleme kuralı, gerekli minimum aralık mesafesini pekiştirmeyi sağlar. Hız eşitleme kuralı ise sürü uçarken böyle bir aralık mesafesinin korunmasına yardımcı olur. Dolayısıyla, her iki kural birbirlerini tamamlayıcı niteliktedir. Aslında, her iki kural da, simüle edilmiş sürü üyelerinin, kaç tane olurlarsa olsunlar, birbirlerine çarpmadan uçabilmelerini sağlar (Ahmed ve Glasgow 2012)

PSO, doğada balık ve kuş gibi büyük gruplar halinde yaşayan hayvanların sürü davranışlarına dayalı olarak 1995 yılında Kennedy ve Eberhart tarafından geliştirilmiştir

(Kennedy 1995). Karınca kolonisi ve sanal karınca algoritmaları gibi birçok algoritma sürü zekası davranışını kullanır. PSO, GA'lar ve karınca kolonisi algoritmaları ile benzerlik gösterebilir ancak mutasyon/çaprazlama operatörlerini ve feromon güncelleme mantığını kullanmadığı için uygulanması daha kolaydır. Bunlar yerine, gerçek sayı rastsallığı ve sürü parçacıkları arasında küresel bir iletişim kullanılır. Ayrıca PSO, gerçek sayı dizilerini de kullanabildiğinden ve GA'larda olduğu gibi ikili sayı dizi parametrelerinin kodlanması veya çözülmesi olmadığından uygulanması daha kolaydır (Yang 2010).

PSO başlangıçta doğrusal olmayan sürekli optimizasyon problemlerini çözmek için önerilmiştir ancak son zamanlarda birçok pratik gerçek hayat problemine uygulanmıştır.

3.2.1. PSO algoritması

Bu kısımda PSO algoritmasının ilk halinden bugünkü haline gelişi kısaca açıklanmaktadır.

Kennedy ve Eberhart, ilk PSO algoritmasını geliştirirken iki ana strateji uygulamışlardır. Bunlar hız eşitleme ve çılgınlık (craziness) adı verilen iki kuraldır. Bu strateji, kuşların halka şeklinde rastgele dağıldığı bir sistem oluşturmuştur. Her yinelemede, programdaki döngüde, en yakın komşusu ile bir parçacık belirlenir. Ardından parçacığın X ve Y hızları, etkileşime girdiği parçacığa atanır. Tahmin edilebileceği gibi, böyle bir stratejinin tek başına kullanımı sürülerin zamanla ortak ve değişmeyen bir yönde hareket etmesine sebep olduğu görülmüştür. Bunu önlemek için, çılgınlık olarak adlandırılan rastgele belirlenen (stokastik) bir değişken eklenmiştir. Her yinelemede, rastgele seçilen X ve Y hızlarına bir miktar değişim eklenmiştir. Bu sisteme yeterince çeşitlilik katmıştır ve simülasyona "gerçek yaşam benzeri" bir görünüm kazandırmıştır.

Kennedy ve Eberhart, Heppner (Heppner ve Grenander 1990)'in arkadaşıyla yapmış olduğu konaklama (kuşların önceden bildiği bir yer) kavramını yiyecek (kuşların aramaları gereken yer) ile değiştirerek bir sonraki adımı atmıştır. Böylece bu sosyal simülasyon algoritması, bir optimizasyon paradigmasına dönüştürülmüştür. Ana fikir, aracıların (kuşların) arama alanlarında (yiyecek kaynağı) birbirlerinin bilgisinden yararlanarak bilinmeyen bir yeri bulmasını sağlamaktı. Her aracı kendisinin ve tüm sürünün en iyi pozisyonunu hatırlayabilmektedir (Kaveh 2017).

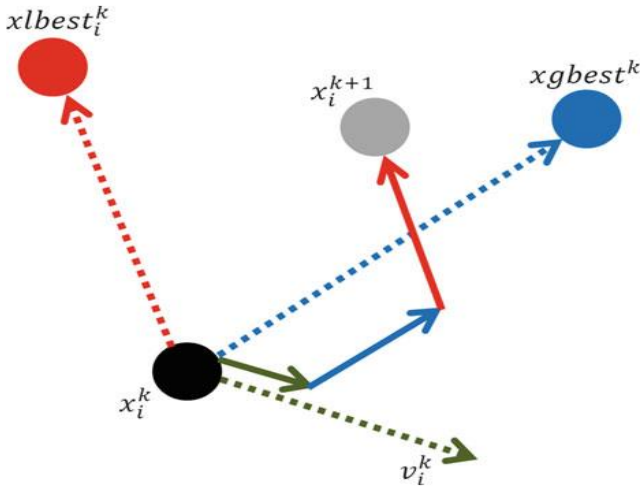
Optimize edilecek matematiksel fonksiyonun ekstremumu (maksimum veya minimum noktası) besin kaynağı olarak düşünülebilir. Bir dizi küçük değişikliklerden ve yardımcı değişkenlerin ortadan kaldırılmasından sonra, bir parçacığın sonraki konumunun hesaplanması için güncelleme kuralları Denklem 3.1 ve Denklem 3.2'deki gibi belirlenmiştir (Kaveh 2017).

$$V_{i,j}^{k+1} = V_{i,j}^{k} + c_1 r_1 \left(Xbest_{i,j}^{k} - X_{i,j}^{k}\right) + c_2 r_2 (Xgbest_{j}^{k} - X_{i,j}^{k}) \quad (3.1)$$

$$X_{i,j}^{k+1} = X_{i,j}^{k} + V_{i,j}^{k+1} \quad (3.2)$$

$X_{i,j}^{k}$ ve $V_{i,j}^{k}$ sırasıyla j. öğenin i. parçacığının konum ve hız vektörleridir. k mevcut konum ve hız değerlerini k+1 ise yeni konum ve hız değerlerini temsil etmek için kullanılmıştır. r1 ve r2 aralığı (1,0) olan düzgün dağılmış iki rastgele sayıdır; $xbest_i$ ve xgbest sırasıyla şimdiye kadar i inci parçacık ve bütün sürü tarafından deneyimlenen en iyi pozisyonlardır. Ayrıca c1 ve c2 parçacıkların kendine (biliş) ve sürüye (sosyal davranış) olan güvenini temsil eden iki parametredir. Bu iki parametre, Kennedy ve Eberhart'ın sunduğu ilk sürümde 2'ye eşitlenmiştir, böylece parçacıklar yarı zamanlı olarak hedefe doğru uçmaktadırlar. Bu iki parametre, arama ve arama eğilimleri arasındaki dengeyi kontrol ettikleri için algoritmanın en önemli parametreleri arasındadır. Nispeten yüksek bir c_1 parçacıkları yerel en iyi deneyimlerine doğru yönlendirmeye teşvik ederken, c_2'nin yüksek olduğu değerler küresel en iyi konuma daha hızlı bir şekilde yakınsama sağlayacaktır. Şekil 3.9'da bir parçacığın hareketi gösterilmiştir (Kaveh 2017).

Rastgele dağıtılan parçacıklarla (potansiyel çözümler) başlayan algoritma, çözümleri bir kalite ölçüsüne (uygunluk fonksiyonuna) göre geliştirmeye çalışmaktadır. Uyarlama, arama uzayı etrafında dolaşan parçacıkların arasındaki iletişimleri modelleyen yukarıdaki basit matematiksel ifadeler vasıtasıyla gerçekleştirilir. PSO'nun en basit biçiminde bu matematiksel ifadeler, her parçacığın kendi geçmişindeki en iyi konumuna ve şimdiye kadar sürüdeki en iyi konuma doğru bazı rastgele sapmalarla hareket etmesi şeklinde önerilmiştir. Bununla birlikte, farklı güncelleme kuralları kullanan bir sürü çeşidi de vardır (Kaveh 2017).



Şekil 3.9 Örnek bir parçacık (Kaveh 2017)

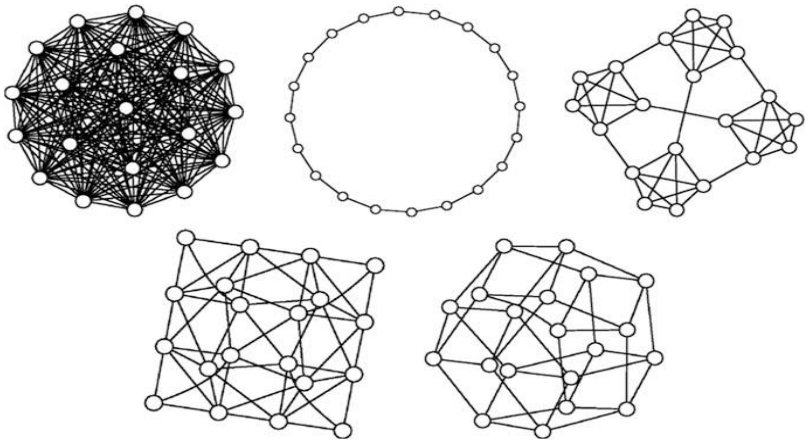
PSO'nun temel adımları, Şekil 3.10'da gösterilen kaba kodla özetlenebilir:

---

1. amaç fonksiyonu f{x), x = (xi, ...,xp)T
2. n adet parçacığın ilk Xi konum ve Vi hız ata
3. Global en iyi g* 'yi bul min f{x1),..., f(xn)} (t = 0 'daki)'dan
4. while (şart sağlandığı sürece)
5. t = t + 1 (kaba zaman veya çevrim sayacı)
6. for tüm n parçacıklarının tüm p boyutları için döngü oluştur
   i. yeni hız değerini $V_{i,j}^{k+1}$ (3.1).denklemi kullanarak
   ii. yeni konumu hesapla $X_{i,j}^{k+1} = X_{i,j}^{k} + V_{i,j}^{k+1}$
   iii. yeni konum değeri $X_{i,j}^{k+1}$ için amaç fonksiyonunu hesapla
   iv. şimdiki kişisel en iyiyi $X_{*,j}^{k+1}$'deki her bir parçacık için bul
7. bitiş for
8. şu anki global en iyiyi g* bul
9. bitiş while
10. x* ve g* için son değerleri yazdır

Şekil 3.10 Kaba kod PSO

Toplam parçacık sayısı, toplam yineleme sayısı, eylemsizlik ağırlığı ve/veya kısıtlama faktörü ve bilişsel ve sosyal davranış katsayıları (c1 ve c2) basit PSO'da bulunan ana parametrelerdir. Şekil 3.11 bazı komşuluk topolojileri gösterilmiştir



Şekil 3.11 Bazı komşuluk topolojileri (Kaveh 2017)

Denklem (3.1)'deki Xgbest klasik PSO'da sürüdeki en iyi konum olarak kabul edilir, ancak bu her zaman geçerli değildir. PSO'da parçacıklar arası iletişim için farklı topolojiler tanımlanmış ve kullanılmıştır (Mendes ve ark 2004, Matsushita ve Nishio 2009). Yani, bir parçacığın konumunu güncellerken xgbest, bütün sürü yerine bağlı olduğu parçacıkla sınırlı bir komşuluktaki en iyi parçacık konumunu ifade edebilir. PSO'daki yığın topolojilerinin algoritmanın performansını önemli ölçüde etkilediği gösterilmiştir. Şekil 3.11'de Mendes ve arkadaşları (Mendes ve ark 2004) tarafından geliştirilen bazı temel PSO komşuluk topolojileri gösterilmiştir. Bunların dışında çok sayıda farklı topoloji de mevcuttur (Kaveh 2017).

Farklı topolojiler, bilginin parçacıklar arasında dolaşımını etkilemektedir. Bu da algoritmanın keşif (exploration) ve sömürme (exploitation) yeteneklerini ve yakınsama oranını etkilemektedir. Standart PSO tam bağlı bir topoloji kullanmaktadır. Bu topolojide bütün parçacıklar komşudur. Bu tarz bir topoloji hızlı bir yakınsama (muhtemelen olgunlaşmamış) sergilemektedir. Bütün parçalar, global en iyiye bağlı olduğu için bu durumdan doğrudan etkilenirler. Bu sebepten ötürü problem uzayının başka alanlarını keşfedemezler. Bu yüzden muhtemelen yerel optimuma takılırlar (Kaveh 2017).

Tam bağlı topolojinin yerine kullanılan yağmur topolojisi ise minimum düğüm derecesine sahip sıradan bir graf şeklinde temsil edilir. Bu da bilgi dolaşımının en yavaş yolu olarak düşünülebilir. Global en iyi parçacıktaki bilginin diğer parçacıklara ulaşması daha uzun sürdüğünden yakınsama oranı da düşük olur. Bu iki topoloji dışında Şekil 3.11'de gösterilen topolojiler ve literatürde mevcut olan başka topolojiler de bulunmaktadır.

3.3. Yapay Arı Kolonisi Algoritması

Bireylerin kolektif olarak davrandıkları birçok doğal olay, araştırmacıların popülasyon tabanlı optimizasyon algoritmalarını geliştirmelerine ilham kaynağı olmuştur. Bu popülasyon tabanlı optimizasyon algoritmaları, uygunluk değerlendirmesi üzerinde çalışırlar bu nedenle olası çözüm popülasyonunun, arama uzayının daha iyi uygunluk alanlarına doğru ilerlemesi beklenir. Popülasyon temelli optimizasyon algoritmaları, doğadan esinlenerek optimizasyon problemlerine en yakın optimal çözümleri bulurlar. Bu yöntemlerden biri de YAKA'dır (Karaboga 2005). Derviş Karaboğa tarafından sunulan YAKA, sayısal optimizasyon problemleri için en uygun çözümü bulmakta kullanılan bir yaklaşımdır. Bu algoritma, kaliteli bir yiyecek kaynağı arayan bal arısının davranışlarından esinlenmiştir. YAKA oldukça basit bir yapıda, hızlı ve nüfusa dayalı stokastik (rastgele belirlenen) bir arama tekniğidir (Bansal ve ark 2013).

3.3.1. Doğada bal arılarının davranışları

YAKA, bal arılarının yiyecek arama davranışından esinlenmiştir. Doğada bulunan sürü türlerinden biri olan bal arısı sürüsü yiyecek araştırırken, kolektif zeki bir davranış sergilemektedir. Sürüdeki arılar, bilgi paylaşımı, çevreyi ezberleyebilme, bilgiyi depolayabilme bilgiyi paylaşabilme ve buna dayalı kararlar alabilme gibi birçok nitelik taşırlar. Çevredeki değişikliklere göre sürü kendini günceller, görevleri dinamik bir şekilde belirler ve sosyal öğrenme ve öğretme yoluyla daha uzaklara ilerler. Arılardaki bu zeki davranışlar, araştırmacıları bal arılarının yiyecek arama davranışlarını simüle etmeye teşvik etmiştir (Bansal ve ark 2013).

Bal arısı sürülerinin toplu zekâlarının ortaya çıkmasını sağlayan yem seçiminin minimal modeli, üç temel bileşenden oluşur. Bunlar Yiyecek kaynakları, çalışan toplayıcılar (işçi arılar) ve işsiz (gözlemci arılar) toplayıcılardır. Model başlıca iki yönde davranış türü taşımaktadır. Bunlar zengin bir nektar kaynağından toplamaya başlama ve yoksul bir kaynağı terk etmedir (Karaboga ve ark 2014).

Yiyecek kaynakları: Arılar yiyecek ararken, kendilerine belirli bir çiçek (yiyecek kaynağı) seçerler. Bu besin kaynağından bal arısı, besin kaynağının içerdiği nektar miktarı, bu nektarın çiçekten ne kadar kolaylıkla alınabileceği, yuvadan ne yönde ve ne kadar uzakta olduğu gibi bilgileri toplar. Arılar kolay ve basit olması için bu

olguları tek bir miktar olarak depolar (belirlenen bu yiyecek kaynağı için toplam verimlilik olarak adlandırılır) (Bansal ve ark 2013).

İşçi arılar: İşçi arılar hâlihazırda bal toplama işlemini gerçekleştirdikleri veya "çalıştıkları" belirli bir yiyecek kaynağıyla ilişkilendirilirler. Kendileriyle birlikte bu yiyecek kaynağı hakkında kovana bilgi taşırlar. Bu bilgiler: yuvaya olan mesafe ve yön, kaynağın verimliliği gibi bilgilerdir. Arılar bu bilgileri paylaşması belirli bir olasılığa bağlıdır (Karaboga ve ark 2014).

İşsiz arılar: İşçi arılar, bilgilerini başka bir grup arıyla (işsiz arılar) belli bir olasılıkla paylaşırlar. İşsiz arılar, işçi arılardan elde ettikleri bilgileri özetlemek ve toplama için bir yiyecek kaynağı seçmekle yükümlüdürler. Bu işsiz arılar daha sonra gözcü arılar ve kâşif arılar olarak iki alt kategoriye ayrılırlar. Gözcü arılar, kovandaki işçi arıların getirdiği bilgileri toplayan arılardır. Bu verileri analiz ettikten sonra kendileri için bir yiyecek kaynağı belirlerken, kâşif arılar kovanın çevresinde yeni besin kaynakları bulmaktan sorumludurlar. Mevcut yiyecek kaynaklarının bir kısmı tükendiğinde bu arılar, kovan çevresinde çevreyi araştırmaya başlarlar ve rastgele yeni yiyecek kaynakları bulurlar. Bal arısı sürülerinin genelde ortalama % 50'si işçi arılardan ,% 50'si işsiz arılardan oluşur. Ayrıca tüm arıların % 5 ila % 10'u kâşif arılardır (Bansal ve ark 2013).

Arılar arasındaki bilgi paylaşımı, kolektif bilginin oluşumunda en önemli olaydır. Kovan incelenirken, tüm kovanlarda yaygın olarak bulunan bazı kısımları fark etmek mümkündür. Bilgi paylaşımıyla ilgili olarak kovanın en önemli kısmı dans alanıdır. Arılar arasında yiyecek kaynaklarının kalitesi ile ilgili iletişim dans alanında gerçekleşir. Bu dansa sallanma (waggle) dansı denir. Mevcut zengin kaynaklarla ilgili bilgi dans pistinde bulunan gözcü arıya sunulduğundan, muhtemelen çok sayıda dansı izlemekte ve en verimli kaynağa yönelmektedir. Verimli kaynaklar hakkında daha fazla bilgi dolaşıma girdiğinden gözcü arıların daha verimli kaynakları seçme olasılığı daha büyüktür. Dolayısıyla, toplanma olasılığı yiyecek kaynağının verimliliği ile orantılıdır (Tereshko ve Loengarov 2005, Karaboga ve ark 2014).

3.3.2. YAKA'nın algoritmik yapısı

Diğer sürü tabanlı optimizasyon algoritmaları gibi, YAKA da potansiyel çözümlerden oluşan bir popülasyondur. YAKA'ya ilişkin, potansiyel çözümler bal arılarının besin kaynaklarıdır. Uygunluk, besin kaynağının kalitesi (nektar miktarı)

bakımından belirlenir. Kolonide üç çeşit arı vardır: işçi arılar, gözcü arıları ve kâşif arılar. İşçi arıların veya gözcü arıların sayısı gıda kaynaklarına eşittir. İşçi arılar besin kaynakları ile ilgilenirken, gözcü arılar kovan içinde kalıp işçi arılarından toplanan bilgileri yiyecek kaynağına karar vermek için kullanırlar. Yiyecek kaynağı tükenen işçi arılar gözcü arı olur ve yeni yiyecek kaynağını rastgele aramaya başlar (Bansal ve ark 2013).

Diğer sürü tabanlı algoritmalara benzer şekilde, YAKA iteratif (yinelemeli) bir yöntemdir. Bir YAKA popülasyonunun evrimini sağlayan iki temel işlem vardır. Bunlar : arama uzayının farklı alanlarını keşfetmeye olanak tanıyan varyasyon işlemi ve önceki deneyimlerden yararlanılmasını sağlayan seçim işlemidir. YAKA yöntemi dört aşamadan oluşur: başlangıç aşaması, işçi arı aşaması, gözcü aşaması ve kaşif aşaması. Bunların her biri aşağıda açıklanmaktadır (Bansal ve ark 2013):

Aşağıda başlangıçtan itibaren sırayla YAKA algoritmasını oluşturan aşamalar anlatılmıştır**.**

Başlangıç aşamasında, YAKA, her çözümü x_i (i = 1, 2, ..., SN), D boyutlu bir vektör olan, eşit dağılmış (uniform) SN adet çözümlü bir popülasyon üretir. Burada D optimizasyon problemindeki değişken sayısıdır ve x_i popülasyondaki i. besin kaynağını temsil eder. Tüm yiyecek kaynakları Denklem 3.3'deki gibi üretilir:

$$X_i^j = X_{min}^j + rand(0,1)\left(X_{max}^j - X_{min}^j\right), \forall j = 1, 2, \ldots, D \quad (3.3)$$

X_{max}^j ve X_{min}^j X_i 'nin j.yöndeki sınırlarıdır.

Daha sonra işçi arı aşaması başlar. Bu aşamada, işçi arılar bireysel deneyimlerini ve yeni çözümün uygunluk değeri (nektar miktarı) bilgilerini kullanarak mevcut çözümü değiştirirler. Yeni yiyecek kaynağının uygunluk değeri eski yiyecek kaynağından daha yüksekse, arı konum olarak yeni konumunu günceller ve eskisini atar. Bu aşamadaki i. adayın j. boyutundaki konum güncelleme işlemi Denklem 3.4'te gösterilmiştir:

$$V_i^j = X_i^j + \phi_i^j (X_i^j - X_k^j) \quad (3.4)$$

$\phi_i^j(X_i^j - X_k^j)$ adım büyüklüğü olarak adlandırılır. $k \in 1,2,\ldots,SN$, $j \in 1,2,\ldots,D$ rastgele seçilen iki indekstir. adım boyutunun belirgin bir katkısı olması için k değeri i'den farklı olmalıdır. ϕ_i^j ise [-1, 1] arasındaki rastgele bir sayıdır.

Gözcü arı aşaması işçi arı aşamasının tamamlanmasının ardından başlatılır. Bu aşamada, tüm işçi arılar, güncellenmiş çözümlerin (besin kaynakları) uygunluk bilgilerini (nektar) ve konum bilgilerini kovandaki gözcü arıları ile paylaşmaktadırlar. Gözcü arıları mevcut bilgileri analiz eder ve uygunluğuna bağlı olarak p_i ihtimalle bir çözüm seçerler. p_i olasılığı Denklem 3.5 kullanılarak hesaplanabilir:

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{3.5}$$

Burada fit_i *i.* çözümün uygunluk değeridir. İşçi arısı aşamasında olduğu gibi, gözcü arı, hafızasındaki pozisyonda bir değişiklik yapmak için adayın çözümün uygunluğunu kontrol eder. Eğer uygunluk değeri önceki uygunluk değerinden yüksekse, arı yeni pozisyonu ezberleyip eskisini unutur.

Son olarak terk edilecek besin kaynaklarının belirlenip yenilenmesi amacıyla kâşif arı aşaması başlar. Bir yiyecek kaynağının konumu önceden belirlenmiş bir sayıda döngü süresince güncellenmezse, yiyecek kaynağının terk edileceği varsayılır ve kaşif arı aşaması başlatılır. Bu aşamada, terk edilecek yiyecek kaynağıyla ilişkili arı kaşif arısı haline gelir ve yiyecek kaynağı arama uzayında rastgele seçilen yiyecek kaynağı ile değiştirilir. YAKA'da terk etme sınırı (limit) olarak adlandırılan önceden belirlenmiş döngü sayısı önemli bir kontrol parametresidir. Terk edilecek kaynağın x_i olduğunu varsayalım, o zaman kaşif arı mevcut gıdayı yeni x_i ile değiştirir. Denklem 3.6'da bu durum gösterilmiştir.

$$X_i^j = X_{min}^j + rand(0,1)\left(X_{max}^j - X_{min}^j\right), \forall j = 1,2,\ldots,D \tag{3.6}$$

X_{max}^j ve X_{min}^j X_i 'nin j.yöndeki sınırlarıdır.

3.3.4. YAKA'nın ana adımları

YAKA'nın kaba kodu Şekil 3.12'deki gibidir.

Denklem (3.3) ile çözümler popülasyonunu oluştur ve ilk değerleri ver x_i (i=1,2,….SN);
cevrim=1;
while (cevrim<MCS) **do**

1. denklem(3.4)'yi kullanarak işçi arılar için yeni çözüm v_i oluştur ve amaç değerlendir (fonksiyonlarını hesapla);
2. aç gözlü seçim yöntemini işçi arılar üzerinde kullan;
3. denklem(3.5)'ü kullanarak x_i için p_i olasılık değerlerini hesapla;
4. p_i olasılığına bağlı olarak seçilmiş(rulet tekerleği yöntemiyle) x_i çözümlü gözcü arılar için denklem(3.4)'yi kullanarak yeni çözüm v_i oluştur ve amaç değerlendir (fonksiyonlarını hesapla);
5. aç gözlü seçim yöntemini gözcü arılar üzerinde kullan;
6. kaşif arılar için terk edilecek çözüm varsa denklem (3.6) kullanarak rastgele oluşturulan çözüm x_i ile değiştir;
7. Şimdiye kadar ki en iyi çözümü ezberle;
8. cevrim=cevrim+1;

end while

Şekil 3.12 Kaba kod YAKA

3.3.5. YAKA kontrol parametreleri

YAKA için 3 adet kontrol parametresi bulunmaktadır. Bunlar yiyecek kaynağı sayısı, limit, ϕ_i^j ise [-1, 1] arasındaki rastgele bir sayıdır. Karaboğa (Karaboga 2005) tarafından önerilen bazı parametre değerleri şöyledir;

- ϕ_i^j rand [-1, 1] arasındaki rastgele bir sayıdır.
- Limit=($\frac{sürü\ boyutu}{2}$) x D
- Sürü boyutu 50-100 arı olmalıdır.

3.4. Sınav Çizelgelemesi Problemi

En basit manada sadece t (zaman dilimi) e (sınav) olarak ifade edilebilecek bir çizelge düşünüldüğünde arama uzayımızın boyutu t^e olacaktır. Bu çözüm uzayı içinde elde edilecek farklı çizelgelerden en iyi olanı belirlerken kullanılacak bazı şartlar olacaktır. Uyulmasını istenilen bu şartlara kısıt adını verilir.

Sınav çizelgeleme problemlerinde genellikle iki tip kısıt söz konusudur. Birincisi sert kısıtlar adını verilen her çözüm için karşılanması zorunlu olan kısıtlardır. İkincisi ise esnek kısıtlar olarak adlandırılan çözümlerde karşılanması istenilen ancak ihlalde edilebilen kısıtlardır. Literatürde sıklıkla kullanılan ve bu çalışmada da kullanılan bazı sert kısıtlar aşağıdaki gibidir.

1) Aynı öğrenci iki farklı sınava aynı anda giremez.
2) Sınıf kapasitesi dersi alanlardan düşük olamaz.
3) Bir gözetmen aynı anda iki sınava giremez
4) Aynı anda yapılması istenen sınavlara dikkat et
5) Aynı anda yapılmaması istenen sınavlara dikkat et
6) Birbirini izleyen zamanlarda atanması istenen sınavlara dikkat et
7) Belirli bir zaman dilimine atanması istenmeyen sınavlara dikkat et
8) Belirli bir zaman dilimine atanması istenen sınavlara dikkat et

Yukarındaki sert kısıtların dışında Selçuk Üniversitesi Bilgisayar Mühendisliği bölümünde uygulanan sınavlarda aynı anda sadece tek sınava izin verilmektedir. Sebebi ise bölümdeki derslere kayıtlı öğrenci sayılarının oldukça fazla, sınıfların kapasitelerinin ve sayılarının öğrenci sayısına oranla oldukça düşük kalmasıdır. Bu yüzden bir sınava birden fazla sınıf atanması gerekmektedir. Fiziki kısıtlar göz önüne alındığında kromozomda sınıf ve gözetmen için ayrılacak genlerin uzunluğu kromozomun da boyutunu arttıracaktır. Bu da elde edilecek çözümün kalitesini oldukça düşürecektir. Bu durumdan kaçınmak ve çözüm kalitesini arttırmak için bu çalışmadaki genler sadece sınava ait zaman bilgilerini içermektedir.

Esnek kısıtlar ise şu şekildedir.

1) Bir öğrenci için aynı günde 2'den fazla sınav atamamayı çok şiddetli tercih et.
2) Bir öğrenci için aynı yarı gün içinde birbirini izleyen sınav atamamayı şiddetli tercih et.
3) Bir öğrenci için aynı gün farklı yarı günlerde birbirini izleyen sınav atamamayı tercih et.
4) Bir günde zorluk derecesi yüksek ve çok yüksek olan birden fazla sınav atamamayı tercih et.
5) Bir öğrenci için belirli zaman aralığında belirli sayıda sınavdan fazla ders atamamayı tercih et
6) Yüksek kapasiteli sınavlara çizelgenin sonlarında atama yap

Çizelgeler için sınıf ve gözetmen değerleri açgözlü yaklaşım kullanılarak verilmektedir. Bu işlem uygulanan yöntemlerden bağımsız olarak

gerçekleştirilmektedir. Nedeni ise gerekli sınıf ve gözetmen sayılarının oldukça büyük olmasına bağlı olarak çözüm uzayının büyümesi, çözüm kalitesinin düşmesidir. Bölüm 3.4.1, 3.4.2 ve 3.4.3 'de uygulanan model ve gösterime dair ayrıntılı bilgi için McCollum ve arkadaşlarının (2012) çalışmasına bakınız.

3.4.1. Parametreler ve matematiksel gösterim

$\boldsymbol{E}$: Sınavlar kümesi.

$\boldsymbol{s_i^E}$: Sınavın boyutu $i \in E$

$\boldsymbol{S}$: Öğrenciler kümesi.

$\boldsymbol{t_{is}}$: Eğer *s* öğrencisi *i* sınavına katılıyorsa 1 aksi halde 0

$\boldsymbol{R}$: Sınıflar kümesi.

$\boldsymbol{P}$: zaman dilimleri kümesi.

$\boldsymbol{EP^{sbt}}$ **:** p zaman diliminde yapılması gereken önceden belirlenmiş r sınavları kümesi (e,p) ∈ EP^{sbt} ⊂ E x P

$\boldsymbol{EP^{ysk}}$ **:** p zaman diliminde yapılmaması gereken önceden belirlenmiş r sınavları kümesi (e,p) ∈ EP^{ysk} ⊂ E x P

HARİCİ : aynı anda yapılması istenmeyen sınavlar kümesi (i,j) ∈ HARİCİ ⊂ E x E

DAHİLİ : aynı anda yapılması istenen sınavlar kümesi (i,j) ∈ DAHİLİ ⊂ E x E

İZLEYEN: birbirini izleyen sınavlar kümesi (i,j) ∈ İZLEYEN ⊂ E x E

$\boldsymbol{W^{2R}}$**:** Birbirini izleyen iki sınav için ağırlık değeri

$\boldsymbol{W^{2D}}$**:** Aynı gün içerisindeki iki sınav için ağırlık değeri

$\boldsymbol{W^{DH}}$**:** Ayrı yarım günlerde birbirini izleyen iki sınav için ağırlık değeri

$\boldsymbol{W^{PD}}$**:** Ayrı yarım günlerde birbirini izleyen iki sınav için ağırlık değeri

$\boldsymbol{X_{ip}^{P}}$**:1** eğer *i* sınavı *p* periyodunda ise, aksi halde 0

$\boldsymbol{X_{ir}^{R}}$**:1** eğer *i* sınavı *r* sınıfında ise, aksi halde 0

$\boldsymbol{X_{ipr}^{PR}}$**:1** eğer *i* sınavı p periyodunda ve *r* sınıfında ise, aksi halde 0

$\boldsymbol{Y_{pq}}$ **: 1** eğer p ve q periyotları aynı gündeyse , aksi halde 0

$\boldsymbol{f_p^P}$ **:1** eğer p periyodu ön yük cezası kapsamında ise, aksi halde 0

İhlal durumlarında için geçerli ceza değerleri

$\boldsymbol{C_S^{2R}}$**:** s öğrencisi için art arda iki sınav cezası

$\boldsymbol{C_S^{2D}}$**: :** s öğrencisi için aynı günde iki sınav cezası

$\boldsymbol{C_S^{DH}}$ **:** s öğrencisi için aynı günde farklı yarım günlerde birbirini izleyen iki sınav cezası

$\boldsymbol{C_S^{PD}}$ **:** s öğrencisi için belirli zaman aralığında belirli sayıdan fazla sınav cezası

Hedef fonksiyonumuz Denklem 3.7'de verilmiştir.

$$F(x) = \sum_s^S (W^{2R} C_S^{2R} + W^{2D} C_S^{2D} + W^{DH} C_S^{DH} + W^{PD} C_S^{PD}) + C^R + C^{ÖY} + C^P \quad (3.7)$$

Dikkat edilebileceği gibi sınıf kapasite ihlali için ayrıca bir ağırlık belirlenmemiştir çünkü zaten tanımı gereği ilgili ağırlıkları içermektedir. C^P ise periyot ile bağlantılı sert kısıtların (birbirini izleyen sınavlar, belirli zamana atanması istenen sınavlar vs.) tamamını temsil etmektedir. $C^{ÖY}$ yüksek katılımlı sınavların sonlarda olması ile bağlantılı kısıttır.

3.4.2. Sert kısıtlar

Her sınava en az bir zaman dilimi atanmalıdır, (Denklem 3.8).

$\forall i \in E$ *için*

$$\sum_{p=1}^{P} \mathrm{X}_{\mathrm{ip}}^{\mathrm{P}} = 1 \quad (3.8)$$

Herhangi bir zaman diliminde, herhangi bir öğrenci en fazla bir sınava girmelidir, (Denklem 3.9).

$\forall p \in P \quad \forall s \in S$ *için*

$$\sum_{i=1}^{E} t_{is} \, \mathrm{X}_{\mathrm{ip}}^{\mathrm{P}} \leq 1 \quad (3.9)$$

Birbirini izleyen sınavlara dikkat edilmelidir, (Denklem 3.10).

$\forall (i, j) \in$ *İZLEYEN* $\forall p, q \in P,$ *VE* $p \leq q$ *için*

$$\mathrm{X}_{\mathrm{ip}}^{\mathrm{P}} + \mathrm{X}_{\mathrm{jq}}^{\mathrm{P}} \leq 1 \quad (3.10)$$

Aynı anda olması gereken sınavlara dikkat edilmelidir, (Denklem 3.11).

∀(i, j) ∈ DAHİLİ ∀p ∈ P için

$$X_{ip}^{P} = X_{jp}^{P} \quad (3.11)$$

Aynı anda olmaması gereken sınavlara dikkat edilmelidir, (Denklem 3.12).

∀(i, j) ∈ HARİCİ ∀p ∈ P için

$$X_{ip}^{P} + X_{jp}^{P} \leq 1 \quad (3.12)$$

Belirli bir zaman dilimine atanması istenmeyen sınavlara dikkat edilmelidir, (Denklem 3.13).

(i,p) ∈ EP^{ysk} ve i ∈ E ve p ∈ P için

$$X_{ip}^{P} = 0 \quad (3.13)$$

Belirli bir zaman dilimine atanması istenen sınavlara dikkat edilmelidir, (Denklem 3.14).

(i,p) ∈ EP^{sbt} ve i ∈ E ve p ∈ P için

$$X_{ip}^{P} = 1 \quad (3.14)$$

3.4.3. Esnek kısıtlar

Art arda yer alan i, j sınavlarına katılan s öğrencisine rastlanırsa, $\boldsymbol{C_S^{2R}}$ olarak belirtilen art arda iki sınav cezası bir arttırılır, (Denklem 3.15).

$$C_S^{2R} = \sum_{i,j\ j\neq i}^{E} \sum_{p,q,q=p+1\ ,ypq=1}^{P} t_{is}\, t_{js}\ X_{ip}^{P}\, X_{jq}^{P} \quad (3.15)$$

Aynı günde yer alan i, j sınavlarına katılan s öğrencisine rastlanırsa, $\boldsymbol{C_S^{2D}}$ olarak belirtilen aynı günde iki sınav cezası bir arttırılır, (Denklem 3.16).

$$C_S^{2D} = \sum_{\boldsymbol{i,j}\ j\neq i}^{\boldsymbol{E}} \sum_{\substack{p,q \\ ,q>p+1 \\ ypq=1}}^{\boldsymbol{P}} t_{is}\, t_{js}\ X_{ip}^{P}\, X_{jq}^{P} \quad (3.16)$$

Belirli bir g zaman aralığında yer alan i,j sınavlarına katılan s öğrencisine rastlanırsa, $\boldsymbol{C_S^{PD}}$ olarak belirtilen periyot dağılım cezası bir arttırılır, (Denklem 3.17).

$$C_S^{PD} = \sum_{\boldsymbol{i,j}\ j\neq i}^{\boldsymbol{E}} \sum_{\substack{p,q \\ ,p<q\ \leq p+q}}^{\boldsymbol{P}} t_{is}\, t_{js}\ X_{ip}^{P}\, X_{jq}^{P} \quad (3.17)$$

Ön yük cezası herhangi bir i sınavı için f_p^P =1ile $f_i^E = 1$olduğunda uygulanır, (Denklem 3.18).

$$C_S^{2D} = \sum_i^E \sum_p^P f_i^E \; f_p^P \; X_{ip}^P \, X_{jq}^P \tag{3.18}$$

3.5. Genetik Algoritmanın Sınav Çizelgeleme Problemine Uygulanması

GA'lar sezgisel yöntem olduklarından dolayı çözüm için bütün seçenekleri üretmek yerine eldeki belirli sayıdaki alternatiften en iyi olanını seçerler. Bu yüzden optimum çözüm elde edilmesini garanti etmez. Ancak kısıtlı zaman ve kaynaklı optimuma yakın çözümler üretir. Ayrıca çözümü aramak için kullandığımız amaç fonksiyonumuzu çok geniş bir spektrum da arayabilir. Çözüm ile alakalı önceden herhangi bir bilgisi olmasa da şartlara uyum sağlayarak öğrenme yapabilir.

- GA'lar fazla parametreye duyarlıdır.
- GA için kullanılacak çaprazlama, mutasyon gibi yöntemlerden hangilerinin kullanılacağını belirlemek zordur.
- Ayrıca bazen problemleri ve verileri GA'nın istediği formata çevirmek oldukça güçtür.

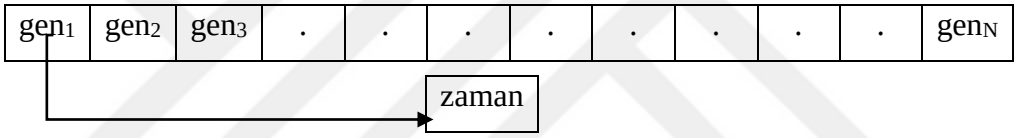
Yukarıda belirtildiği gibi bir problemin çözümü için gerekli formatı belirlemek oldukça güç olabilir. Bu çalışmada Selçuk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümündeki sınav çizelgeleme problemlerinin kısıtlarını göz önünde bulundurularak GA yöntemini uygulanmıştır.

3.5.1. GA parametrelerinin tanımlanması ve kodlama

GA ile bir problem çözerken yapılması gerek ilk önemli adım çözümün kromozom olarak nasıl gösterilmesi gerektiğinin belirlenmesidir. Sınav çizelgeleme probleminde kromozomların gösterim şekli sıralı sayı dizileri şeklinde olacaktır. Örnek olarak çizelgelenmesini istenilen 3 sınav ve 5 zaman dilimi olduğunu varsayılırsa. Kromozom her sınavın yapılacağı zaman dilimi şeklinde ifade edilebilir örnek olarak; 2,4,1 şeklinde ifade edilen sınav çizelgesinde 1. Ders 2. Zaman diliminde, 2. Ders 4. Zaman diliminde, 3. Ders ise 1. Zaman diliminde yapılacaktır. Bu şekilde yapılacak bir kromozom gösteriminde kullanılacak standart çaprazlama ve mutasyon operatörlerinden sonra kromozom üzerinde herhangi bir çatışma kontrolüne ve tamire ihtiyaç olmayacaktır. Bu da zamandan tasarruf ederek problem kısıtlarına odaklanmamıza imkân sağlayacaktır. Bunun dışında zaman çizelgelerinin aslına benzer şekilde genlerin zaman dilimi ve onların aldığı değerlerin sınav olduğu gösterimlerde mevcuttur. Bu

gösterimin temel problemi ise şöyledir. Çaprazlama veya mutasyon sonucunda elde edilecek çözümlerde çizelgelenmesi zorunlu olan sınavlar atlanabilir. Bu da daha sonra bir algoritmayla kromozomdaki çizelgelenmesi zorunlu bu derslerin yeniden tanımlanmasını gerektirir. Bu çalışmada gösterim olarak birinci anlatılan sınavların zaman dilimi değerlerini aldığı yöntem kullanılmıştır. Uygulanan kromozom yapısı aşağıdaki gibidir. Bütün derslere sınav değeri verilebilmesi amacıyla kromozom ders adedince genden oluşmaktadır. Her bir ders için genler, dersin yapılacağı sınav zamanı bilgisini içermektedir.

Fiziki kısıtlar göz önüne alındığında kromozomda sınıf ve gözetmen için ayrılacak genlerin uzunluğu kromozomun da boyutunu arttıracaktır. Bu da elde edilecek çözümün kalitesini oldukça düşürecektir. Bu durumdan kaçınmak ve çözüm kalitesini arttırmak için genleri sadece sınava ait zaman bilgilerini içermektedir. Şekil 3.13'te sınav çizelgeleme için GA kromozom yapısı gösterilmiştir.



Şekil 3.13 Sınav Çizelgeleme için GA Kromozom Yapısı

Şekil 3.14'te örnek bir kromozom yapısı gösterilmiştir. Genler sınav zamanı bilgisini tutmaktadır ve sayısı ders sayısına eşittir.

4	13	9	5	2	12	7	2	3	6	2	3	8	5	2

Şekil 3.14 Örnek Kromozom

3.5.2. Başlangıç popülasyonu , doğal seçim ve uygunluk fonksiyonu

Başlangıç popülâsyonu ataması ile çözüm uzayını budanmasına yardımcı olur. Daha sonra bu çözümler üzerine çaprazlama, mutasyon ve değerlendirme (amaç fonksiyonu) gibi GA işlemlerini yaparak nesilden nesile çözümler geliştirilebilir.

Başlangıç popülâsyonu rastgele seçilmektedir. Doğal seçimde ise rulet tekerleği yöntemi kullanılmıştır. Bu sayede kalitesi yüksek bireylerin seçilme olasılığı artacak ve daha hızlı bir yakınsama elde edilecektir.

Uygunluk fonksiyonu için ise Denklem 3.7'de gösterilen hedef fonksiyonu(F(x)) kullanılmıştır. Kullanılan F(x) fonksiyonu için uygunluk değeri Denklem 3.19'daki formülle bulunur.

$$f(x) = \frac{1}{1+F(x)} \tag{3.19}$$

3.5.3. Çaprazlama

Çaprazlama operatörü olarak tek noktalı çaprazlama seçilmiştir. Çaprazlama olasılık değeri olarak 0.9 seçilmiştir. Seçilen kromozom çiftleri arasında çaprazlama yapılıp yapılmayacağını belirlemek amacıyla 0,1 arasında rastgele bir sayı seçilmektedir. Seçilen sayı 0.9 değerinden küçükse kromozomlar çaprazlamaya tabi tutulup 2 adet yeni kromozom şekilde edilmektedir.

3.5.4. Mutasyon

Mutasyon olasılığı 0.01 belirlenmiştir. 0 ve 1 arasında rastgele seçilen sayının büyüklüğü bu değerden küçük ise popülâsyondaki ilgili kromozomdaki ilgili gen mutasyona tabi tutulur. Eski değeri yenisi ile değiştirilir.

3.5.5. Gelecek nesil ve yakınsama

Mutasyon ve çaprazlamaya tabi tutulan kromozomlar yenileri ile değiştirilir. Daha sonra yeniden uygunluk değerleri hesaplanır ve algoritma çalışmasına devam eder. Algoritmada yakınsama yerine çevrim sayısı belirlenmiştir. Algoritma toplam 2500 kez tekrar etmektedir.

3.6. PSO Algoritmasının Sınav Çizelgeleme Problemine Uygulanması

PSO algoritması sürekli yani sayısal değerli problemlerin çözümü için geliştirilmiştir. Sınav çizelgeleme problemleri ise ayrık değerli problemler olduğu için PSO algoritması modifiye edelerek çözümde kullanılmıştır. Bunun için GA'da kullanılan çaprazlama operatörüne benzeyen bir yöntem kullanılmıştır. Bu yöntem 3

adımdan oluşmaktadır. Aşağıda bu yöntemden ve kullanılan PSO-S algoritmasından bahsedilmiştir.

Hedef (amaç) fonksiyonu olarak Denklem 3.7'de matematiksel gösterimini verilen fonksiyon kullanılmaktadır.

3.6.1. Sınav programı için PSO-S algoritması elemanları

Gösterim olarak GA'da kullanılan gösterim şekli kullanılmıştır. Buna göre her bir parçacık, ders sayısı adedince bitten (gen) oluşmaktadır. Her birinin içerdiği değer ise çizelgede bir zaman dilimini belli etmektedir. Başlangıç popülasyonu ise rastgele oluşturulmuştur.

Amaç fonksiyonu Denklem (3.7)'de verilmiştir. Bu fonksiyonu kullanarak parçacıklar sıralanacak, yerel en iyi ve global en iyi parçacıkları belirlenecektir. Yerel en iyi, bir parçacığın önceki çevrimler boyunca kendisinin tecrübe ettiği en iyi konumdur (k. parçacık için i. çevrimde yerel en iyi P_k^i). Global en iyi ise sürüdeki herhangi bir parçacık tarafından önceki çevrimler boyunca tecrübe edilmiş en iyi konumdur (bütün parçacıklar için i. çevrimde global en iyi G^i). Global ve yerel en iyileri kullanarak bir parçacığın hareketini şu şekilde gerçekleştirilmiştir.

Birinci adımda mevcut parçacığı (X_k^i) oluşturan belirli sayıda bitler rastgele değiştirilmiştir (Denklem 3.20).

$$S_k^{i+1} = rastgele - değiştir(X_k^i) \tag{3.20}$$

İkinci adımda mevcut parçacığa belirli sayıda bit yerel en iyiden (P_k^{i+1}) kopyalanmıştır (Denklem 3.21).

$$W_k^{i+1} = rastgele - takas(S_k^{i+1}, P_k^{i+1}) \tag{3.21}$$

Üçüncü adımda mevcut parçacığa belirli sayıda bit global en iyiden (G^i) kopyalanmıştır (Denklem 3.22).

$$X_k^{i+1} = rastgele - takas(W_k^{i+1}, G^i) \tag{3.22}$$

Kullanılan bu yöntem sayesinde yerel optimuma takılmaktan kaçınılarak daha iyi çözümler bulmak hedeflenmiştir.

Hazırlanan PSO-S Algoritması için kaba kod Şekil 3.15'te gösterildiği gibidir

---

Her bir parçacık için:
I.Parçacığa ilk değeri rastgele ver
Do:
II.Her bir parçacık için:
1) Uygunluk değerini hesapla
2) Uygunluk değeri geçmişteki en iyi uygunluk değerinden(pbest) iyi ise şu anki değeri yeni pbest değeri olarak kaydet
3) Uygunluk değeri geçmişteki en iyi uygunluk değerinden(gbest) iyi ise şu anki değeri yeni gbest değeri olarak kaydet
III.Bitiş

IV.Her bir parçacık için:
1) Yeni çözüm değerini kullanarak hesapla
1. Parçacığın belirli sayıda bitini rastgele değiştir denklem(3.20)
2. Yerel en iyiden belirli sayıda biti rastgele kopyala denklem(3.21)
3. Global en iyiden belirli sayıda biti rastgele kopyala denklem(3.22)
V.Bitiş
While maksimum çevrim sayısına veya en düşük hata kriterine ulaşılmadığı müddetçe

---

Şekil 3.15 PSO-S kaba kodu

3.7. YAKA'nın Sınav Çizelgeleme Problemine Uygulanması

PSO gibi YAKA da sürekli değerli fonksiyonların optimizasyonunda kullanılmaktadır. YAKA ile sınav çizelgeleme problemini çözerken PSO'ya benzer bir yapı kullanılarak GA'ya benzetim yapılmıştır. Bu sefer global en iyi ve yerel en iyi gibi kavramlar olmadığı için yerine literatürde kullanılan 3 adet komşuluk operatörünü kullanılmıştır. Kullanılan hedef fonksiyonu ve kısıtlar GA ve PSO ile aynıdır.

Alt bölümlerde YAKA-S'in temel adımları gösterilmiştir;

3.7.1. Sınav programı için YAKA-S elemanları

Gösterim olarak GA'larda kullanılan gösterim şekli kullanılmıştır. Buna göre her bir parçacık ders sayısı adedince bitten (gen) oluşmaktadır. Her birinin içerdiği değer ise çizelgede bir zaman dilimini belli eder. Başlangıç popülasyonu ise rastgele

oluşturulmuştur. Amaç fonksiyonu Denklem (3.7)'dir. Kullanılan komşuluk operatörleri aşağıdaki gibidir;

1) Mutasyon operatörü: çözümlerde çeşitliliği sağlamak amacıyla kullanılan bu operatör komşuluk kullanmamaktadır. Mevcut çözümün belirli sayıda bitini rastgele değiştirmektedir.

2) Rastgele alt dizi takas operatörü diziden rastgele seçilen iki alt dizinin konumlarını değiştirmektedir.

3) Rastgele alt dizi ters takas operatörü, diziden rastgele seçilen iki alt dizinin konumlarını değiştirmektedir. Ancak seçilen iki alt dizi takastan sonra yerleştirilirken %50 olasılıkla ters çevrilmektedir.

Kullanılan rastgele alt dizi takas operatörü ve rastgele alt dizi ters takas operatörü Şekil 3.16'da gösterilmiştir. Solda normal takas operatörü sağda ise ters takas operatörüne yer verilmiştir.



Şekil 3.16 Rastgele alt dizi ters takas operatörü

Çözümlere ve parametrelere ilk değerleri verildikten sonra, uygunlukları değerlendirilir. Daha sonra ise işçi arı safhası başlar. Bu safhada her bir arı için yeni çözümler belirli sayıda tekrarla komşuluk operatörlerinden birini kullanarak yeniden hesaplanır. Her bir operatör için %33 seçilme olasılığı vardır. Seçim işlemi [0,1] arasında rastgele bir sayı seçilerek gerçekleştirilir. Seçilen sayı [0,0.33] aralığında ise mutasyon operatörü çalıştırılır, [0.34,0.66] aralığında ise rastgele alt dizi takas operatörü çalıştırılır, [0. 67,1] aralığında ise rastgele alt dizi ters takas operatörü çalıştırılır. Elde edilen yeni çözümler şimdiye kadarki en iyi çözümden iyiyse en iyi çözüm değeri yeni çözümle değiştirilir. İşçi arı aşaması bütün işçi arılar için bir defa gerçekleştirildikten sonra (döngü ile), gözcü arı aşamasına geçilir. Bu aşamada da her bir çözümün uygunluk değerine bağlı olarak turnuva seçimi kullanılarak pi olasılıkla bir çözüm seçilir. Seçilen bu çözüm için komşuluk operatörleri kullanılarak yeni çözüm değeri hesaplanır ve uygunluk değeri bulunur. Daha sonra şimdiye kadarki en iyi çözüm değeri ile şu anki yeni çözüm arasında açgözlü seçim uygulanır. Yani şu anki yeni çözüm daha iyiyse en iyi çözüm değeri güncellenir. Bu işlem gözcü arı sayısı adedince tekrarlanır.

Gözcü arıların tamamı bitince kaşif arı aşamasına geçilir. Bu aşamada ise her bir çözüm (işçi arı) için limit değeri kontrol edilir. Eğer limit değeri aşılmış ise yeni yiyecek kaynağı (çözüm) Denklem (3.6) kullanılarak hesaplanır. Bu işlem maksimum çevrim sayısına ulaşıncaya kadar devam eder. Algoritmaya YAKA-S adı verilmiştir.

HazırlananYAKA-S için kaba kod Şekil 3.17'de gösterildiği gibidir

1. İlk popülasyonu(z_i i=1,2,….,SN) oluştur ve değerleri ata
2. Popülasyondaki çözümlerin uygunluk(f_i) değerlerini hesapla
3. cevrim sayacını 1 'e eşitle
4. **Tekrar et**
5. **Döngü** Her bir işçi arı için
   a. Rastgele bir sayı belirle bu sayıya göre yeni çözümü v_i aşağıdaki üç operatörden biriyle oluştur.
      i. Sayı [0,0.33] aralığında ise mutasyon operatörü
      ii. Sayı [0.34,0.66] aralığında ise rastgele alt dizi takas operatörü
      iii. Sayı [0. 67,1] aralığında ise rastgele alt dizi ters takas operatörü
   b. Uygunluk değeri f_i'yi hesapla
   c. Aç gözlü seçim işlemini uygula
6. **Döngü bitiş**
7. Çözüm z_i için olasılık değeri p_i yi rulet tekerleği yöntemini kullanarak hesapla
8. Döngü her bir gözcü arı için
   a. Çözüm z_i 'yi olasılık değeri p_i 'ye göre rastgele seç
   b. Rastgele bir sayı belirle bu sayıya göre yeni çözümü v_i aşağıdaki üç operatörden biriyle oluştur.
      i. Sayı [0,0.33] aralığında ise mutasyon operatörü
      ii. Sayı [0.34,0.66] aralığında ise rastgele alt dizi takas operatörü
      iii. Sayı [0. 67,1] aralığında ise rastgele alt dizi ters takas operatörü
   c. Uygunluk değeri f_i'yi hesapla
   d. Aç gözlü seçim işlemini uygula
9. Terk edilecek bir yiyecek kaynağı(çözüm) varsa denklem (3.6)'yı kullanarak rastgele yenisini oluştur ve değiştir.
10. Şu ana kadar ki en iyi çözümü kaydet
11. cevrim = cevrim +1
12. cevrim=MCS olana kadar

Şekil 3.17 YAKA-S Kaba kod

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

4.1. Deney Setinin Özellikleri

Bu çalışmada sınav çizelgeleme probleminin GA, PSO-S ve YAKA-S algoritmaları ile çözülmesi amaçlanmıştır. Bu amaçla Selçuk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği verileri ve deney setleri kullanılmıştır. Yöntemler karşılaştırılıp incelenmiştir.

Bu veri kümesi ile gerçekleştirilen eski iki sınavın özellikleri aşağıdaki gibidir. Birinci kolon 2016 -2017 güz dönemine ait vize sınavına, ikinci kolon ise final sınavına aittir. Her iki sınavda bir gün içerisinde 6 adet sınav yapılmıştır. Vize dönemi 7 gün final dönemi ise 10 gün sürmüştür. Sınavlarda 16 farklı gözetmen sırasıyla 9 ve 10 adet sınıfta görev almıştır. Bu kısıtlar göz önünde bulundurularak hazırlanan sınav programlarında ilgili parametrelere ait değerler kullanılmıştır. Çizelge 4.1'de veri kümesi özellikleri verilmiştir.

Çizelge 4.1 Veri kümesi özellikleri

Veri	Vize	Final
Kayıtlı öğrenci sayısı	843	843
Toplam ders sayısı	33	33
Gözetmen sayısı	16	16
Günlük zaman dilimi sayısı	6	6
Toplam gün sayısı	7	10
Sınıf sayısı	9	10

E adet sınavdan ve T adet zaman diliminden oluşan basit bir çizelgeleme problemi için arama uzayının boyutu, her bir sınav dilimi E adet değer alabileceği için T adet zaman dilimi için T^E olacaktır. Problemin boyutunun artması ve problemdeki kısıtların değişkenliği nedeniyle optimum çözümün elde edilmesi zorlaşacaktır. Bu sebeplerden ötürü GA, YAKA-S, PSO-S gibi sezgisel yöntemleri çizelgeleme problemlerinin çözümünde kullanılması mantıklı olacaktır.

4.2. Kullanılan Parametreler

Çizelge 4.1'de belirtilen şartlara ve kısıtlara göre daha önce elle oluşturulan sınav çizelgelerimizin ceza puanları sırasıyla 9400 ve 4835 olarak hesaplanmıştır.

Gerçekleştirilecek deneyler için 125.000 fes(fonksiyon çağırma sayısı) değeri seçilmiştir. Bu sebeple kullanılan 3 sezgisel yöntemde de popülasyon sayısı 10 ile 100 arasında, maksimum çevrim sayısı ise 1250 ile 12500 arasında değişen büyüklüklerde seçilmiştir. GA'da çaprazlama ve mutasyon olasılıkları sırasıyla 0.90 ve 0.01 olarak seçilmiştir. Bunlara ait parametreler Çizelge 4.2'de verilmiştir.

Kullanılan YAKA-S algoritmasında işçi ve gözcü arı sayısı eşit ve 25 olarak belirlenmiştir. Ayrıca kullanılan komşuluk operatörlerindeki takas işlemi 1 kez gerçekleştirilmiştir. Maksimum geliştirilememe limiti 155 olarak belirlenmiştir.

Kullanılan PSO-S algoritmasında rastgele değiştirilen, global ve yerel en iyiden kopyalanan bilgilerin sayısı, ders sayısı olan 33 değerinin üçte biri seçilerek bu üç parametrenin de çözüme eşit oranda etki etmeleri hedeflenmiştir.

Kullanılan parametreler mümkün olduğunca birbirine eşit seçilerek karşılaştırılmanın daha adil olması amaçlanmıştır. Bir sonraki bölümde bu parametreler kullanılarak iki sınav için (vize sınavı, final sınavı) onar adet deney gerçekleştirilmiştir.

Çizelge 4.2 Sınav çizelgelemede kullanılan parametreler

Parametre adı	GA	PSO-S	*YAKA-S*
Popülasyondaki Birey Sayısı	10-100	10-100	10-100
Çaprazlama Oranı	0.9	-	-
Mutasyon Oranı	0.1	-	-
Fonksiyon Çağırma Sayısı	125000	125000	125000
İşçi Arı Sayısı	-	-	%50
Gözcü Arı	-	-	%50
Takas İşlemi	-	-	1
Limit	-	-	155
Global En İyiden Kopyalanan Ders Boyutu	-	11	-
Yerel En İyiden Kopyalanan Ders Boyutu	-	11	-
Rastgele Değiştirilen Ders Boyutu	-	11	-

Selçuk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği 2016 -2017 güz dönemine ait ders listesi Çizelge 4.3'te verilmiştir.

Çizelge 4.3 Ders listesi

Ders Adı	**İd**
MATEMATİK I	1
TÜRK DİLİ VE EDEBİYATI I	2
ATATÜRK İLK. VE İNK. TAR. I	3
FİZİK I	4
İNGİLİZCE I	5
ALGORİTMA VE PROGRAMLAMA I	6
BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ	7
İŞ SAĞLIĞI VE GÜVENLİĞİ I	8
AYRIK MATEMATİK	9
LİNEER CEBİR	10
LOJİK TASARIM	11
VERİ YAPILARI	12
NESNEYE YÖNELİK PROGRAMLAMA	13
TEMEL ELEKTRONİK	14
İŞLETİM SİSTEMLERİ	15
BİLGİSAYAR ORGANİZASYONU	16
VERİ TABANI I	17
OTOMATA TEORISI	18
BİLİŞİM HUKUKU	19
BİLGİSAYARLA GRAFİK İŞLEMEYE GİRİŞ (SEÇMELİ)	20
E-İŞ/ E-TİCARETE GİRİŞ (SEÇMELİ)	21
WEB PROGRAMLAMA (SEÇMELİ)	22
MİKROİŞLEMCİ TABANLI TASARIM (SEÇMELİ)	23
DOSYA YAPILARI (SEÇMELİ)	24
BİYOMEDİKAL VERİ İŞLEME (SEÇMELİ)	25
SAYISAL GÖRÜNTÜ İŞLEME (SEÇMELİ)	26
SİSTEM PROGRAMLAMA (SEÇMELİ)	27
UZMAN SİSTEMLER (SEÇMELİ)	28
YAPAY SİNİR AĞLARINA GİRİŞ (SEÇMELİ)	29
EVRİMSEL HESAPLAMAYA GİRİŞ (SEÇMELİ)	30
NESNEYE YÖNELİK TASARIM VE ANALİZ (SEÇMELİ)	31
VERİ TABANI II (SEÇMELİ)	32
BULUT BİLİŞİME GİRİŞ (SEÇMELİ)	33

4.2.1. Vize sınavı deneyleri

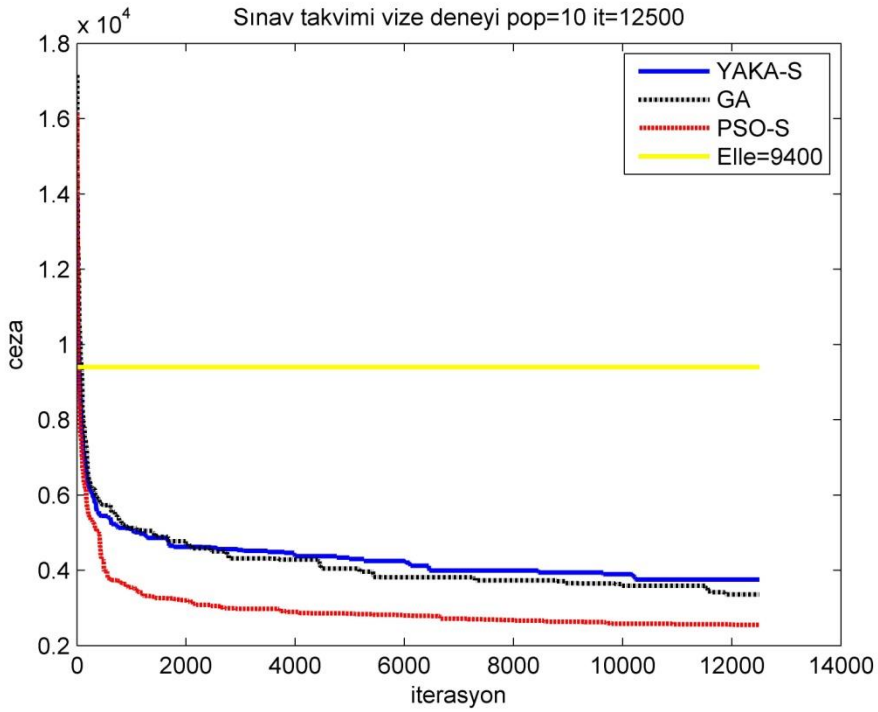
Vize sınavı için Çizelge 4.1'deki veri kümesi ve Çizelge 4.2'deki parametre bilgileri kullanılarak farklı çevrim ve popülasyon büyüklerinde 10'ar adet vize deneyi gerçekleştirilmiştir. Gerçekleştirilecek deneyler için 125.000 fes değeri seçilmiştir ve deneylerin popülasyon sayısı ve çevrim sayısı buna bağlı olarak seçilmiştir. Popülasyon

sayısı ve çevrim sayısı değişkenlerinin çarpımı 125.000 olan fes değerine eşittir. Örnek olarak popülasyon sayısı 10 olarak seçildiğinde 125.000 fes değerini elde etmek için çevrim sayısı 12500 olarak seçilmiştir.

Bu fes değeri kullanılarak popülasyon büyüklükleri sırasıyla 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 olarak belirlenmiştir. İterasyon sayıları ise bu değerlere bağlı olarak sırasıyla 12500, 6250, 4165, 3125, 2500, 2080, 1785, 1560, 1385, 1250 olarak belirlenmiştir.

Deneylere popülasyon sayısı 10 ve çevrim sayısı 12500 seçilerek başlanmıştır. Şekil 4.1'de popülasyon sayısı 10 ve çevrim sayısı 12500 olan vize deneylerine ait 10 adet deneyin ortalama yakınsama değerleri çizelge olarak verilmiştir.

Şekil 4.1'de görüldüğü gibi en iyi sonuçlar PSO-S yöntemi ile elde edilmiştir. Ardından GA en iyi sonuçları vermiştir. YAKA-S ile elde edilen sonuçlar diğer yöntemlere göre daha az başarılıdır.



Şekil 4.1 Vize deneyi 1

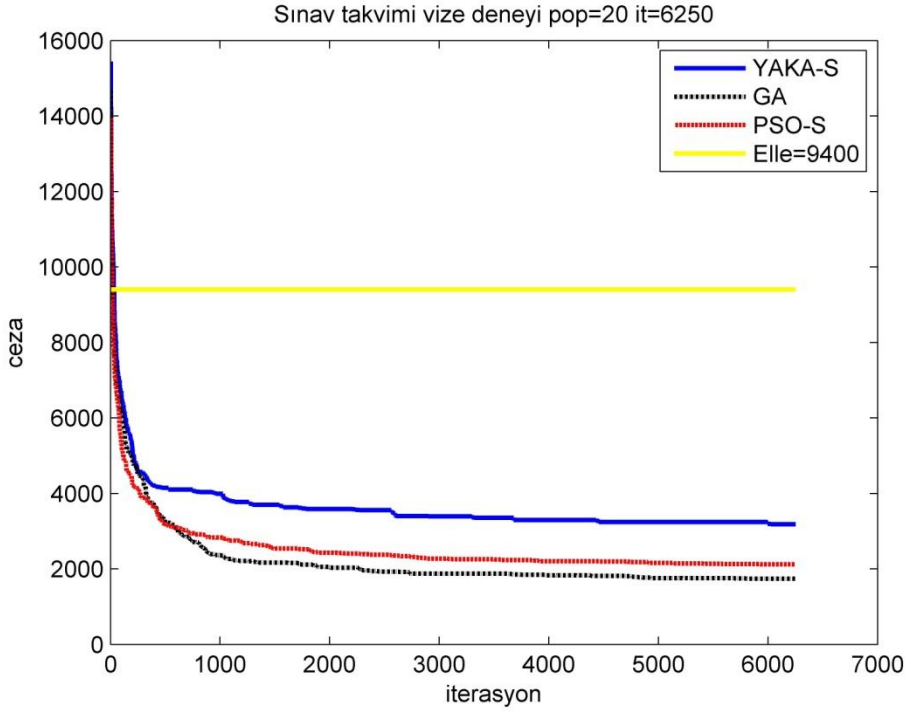
Üç yöntem de popülasyon sayısının küçük olması nedeniyle çok iyi sonuçlar üretememiştir. Bunun başlıca sebebi çevrim ilerledikçe popülasyondaki bireylerin çeşitliliğinin azalması ve birbirine benzemesidir. Ancak PSO-S algoritmasında rastgele değiştirilen derslerin boyutu büyük olduğu için PSO-S algoritması bu çeşitliliği daha

uzun süre sürdürebilmekte ve daha iyi sonuçlar üretebilmektedir. Çizelge 4.4'te de görülebileceği gibi en iyi sonucu ve ortalama değerlerini PSO-S üretmiştir. Ancak verdiği sonuçların kararsız olduğu standart sapma değerinin büyüklüğünden anlaşılmaktadır. YAKA-S yöntemi ise çözüme ait önemli bilgileri saklamadığı için en etkisiz yöntem olmuştur.

Çizelge 4.4 Vize deney 1 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	3359	2615	4035	533.2645	3352.5
YAKA-S	3757	2600	4370	546.9216	3820.0
PSO-S	2547	1750	3790	711.8684	2405.0
El ile	9400	9400	9400	0.0000	9400.0

İkinci deneyde popülasyon sayısı 20 ve çevrim sayısı 6250 seçilmiştir. Şekil 4.2'de ilgili deneyin ortalama yakınsama değerleri Şekil 4.2'de verilmiştir. Şekil 4.2'de görüldüğü gibi en iyi sonuçlar GA yöntemi ile elde edilmiştir. Ardından PSO-S en iyi sonuçları vermiştir. YAKA-S ile elde edilen sonuçlar diğer yöntemlere göre daha az başarılıdır.



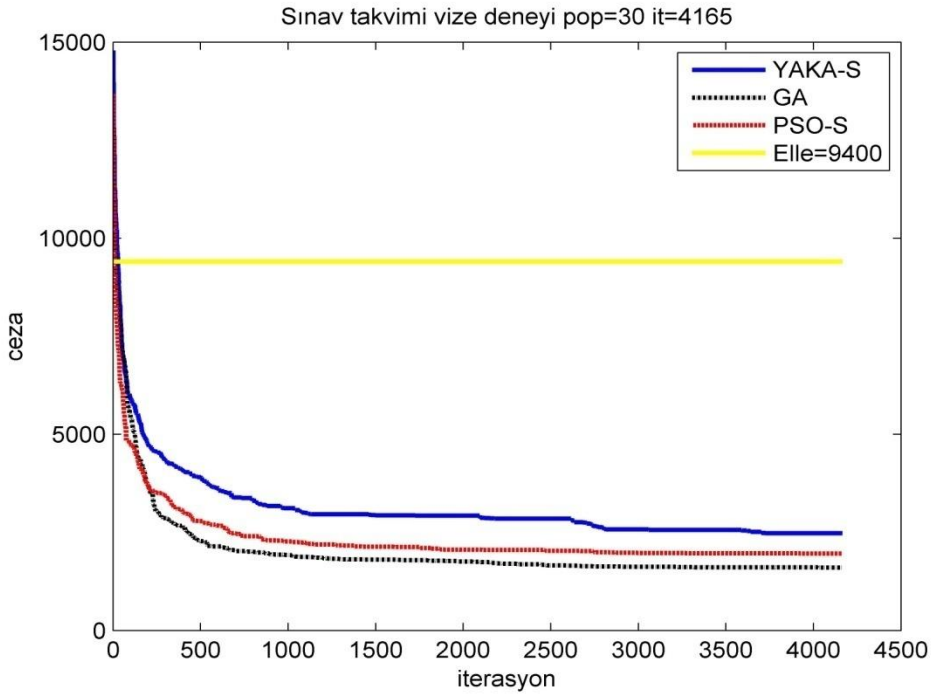
Şekil 4.2 Vize deneyi 2

Popülasyon sayısının artmasıyla beraber GA yönteminin etkinliği Çizelge 4.5' te görülebileceği gibi diğer yöntemlere oranla daha fazla artmıştır. PSO-S'deki rastgele değişime uğrayan derslerin boyutu bir dezavantaj olmaya başlamıştır. YAKA-S yöntemi ise daha önce söylediğimiz gibi çözüme ait önemli bilgileri unutmaktadır. Bu da onun en etkisiz yöntem olmasındaki en önemli faktördür.

Çizelge 4.5 Vize deney 2 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	1739.5	1425	1995	176.6108	1737.5
YAKA-S	3178.0	2690	3550	366.5090	3377.5
PSO-S	2120.0	1575	3480	516.1718	2072.5
El ile	9400.0	9400	9400	0.0000	9400.0

Üçüncü deneyde popülasyon sayısı 30 ve çevrim sayısı 4165 seçilmiştir. Şekil 4.3'de ilgili deneyin ortalama yakınsama değerleri Şekil 4.3'te verilmiştir. Şekil 4.3'de görüldüğü gibi en iyi sonuçlar GA yöntemi ile elde edilmiştir. Ardından PSO-S en iyi sonuçları vermiştir. YAKA-S ile elde edilen sonuçlar diğer yöntemlere göre daha az başarılıdır.



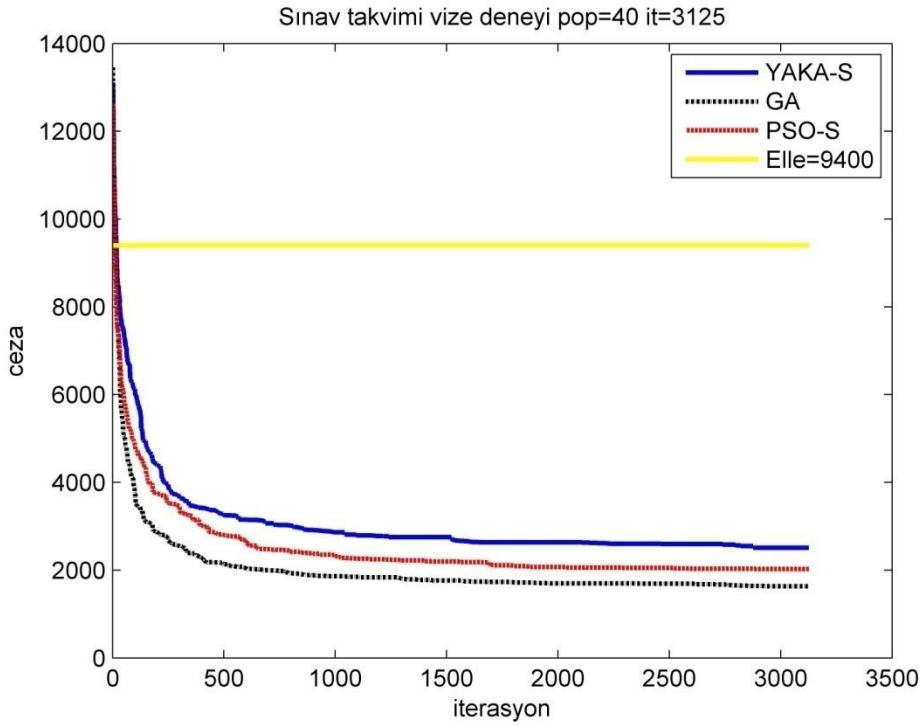
Şekil 4.3 Vize deneyi 3

Popülasyon sayısının artmasıyla beraber 3 yöntemin de etkinliği Çizelge 4.6' da görülebileceği gibi artmaktadır. YAKA-S yöntemindeki artış miktarı PSO-S yöntemine göre daha büyüktür. YAKA-S yöntemi çözüm uzayını araştırma gücünün fazla olması nedeniyle daha büyük gelişim kaydetmektedir. PSO-S yönteminin gelişim hızı ise yavaşlamıştır.

Çizelge 4.6 Vize deney 3 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	1600.0	1480	1770	97.5818	1580.0
YAKA-S	2479.5	1880	3250	374.5549	2512.5
PSO-S	1962.5	1575	2905	383.9433	1825
El ile	9400.0	9400	9400	0.0000	9400.0

Dördüncü deneyde popülasyon sayısı 40 ve çevrim sayısı 3125 seçilmiştir. Şekil 4.4'te ilgili deneyin ortalama yakınsama değerleri verilmiştir. Şekil 4.4'te görüldüğü gibi en iyi sonuçlar GA yöntemi ile elde edilmiştir. Ardından PSO-S en iyi sonuçları vermiştir. YAKA-S ile elde edilen sonuçlar diğer yöntemlere göre daha az başarılıdır.



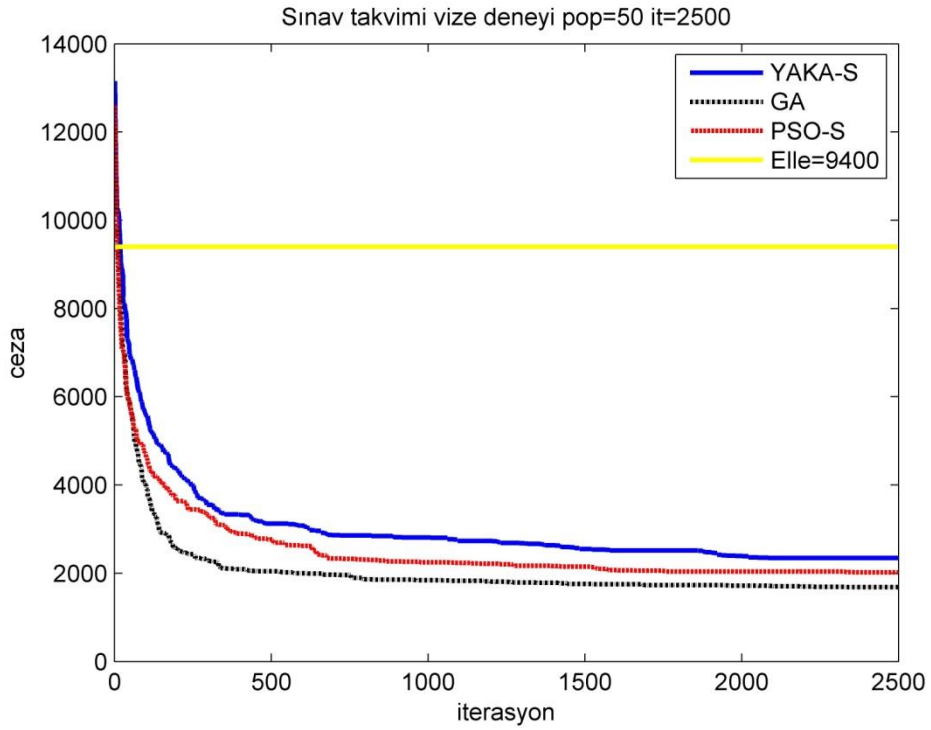
Şekil 4.4 Vize deneyi 4

Çizelge 4.7'de görülebileceği gibi popülasyon sayısının büyütmenin yöntemlere etkisi azalmıştır. Her üç yöntemde önceki deneydekine benzer ortalamada sonuçlar göstermektedir. Yine GA yöntemi en iyi sonuçları vermektedir. Bu deneyde fazla bir gelişim gözükmese de YAKA-S performans olarak PSO-S'e yaklaşmaktadır.

Çizelge 4.7 Vize deney 4 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	1633.5	1475	1900	126.7554	1610
YAKA-S	2506.0	2155	2980	243.5250	2490
PSO-S	2025.5	1675	2650	328.4936	1920
El ile	9400.0	9400	9400	0.0000	9400

Beşinci deneyde popülasyon sayısı 50 ve çevrim sayısı 2500 seçilmiştir. Şekil 4.5'te ilgili deneyin ortalama yakınsama değerleri verilmiştir. Şekil 4.5'te görüldüğü gibi en iyi sonuçlar GA yöntemi ile elde edilmiştir. Ardından PSO-S en iyi sonuçları vermiştir. YAKA-S ile elde edilen sonuçlar diğer yöntemlere göre daha az başarılıdır.



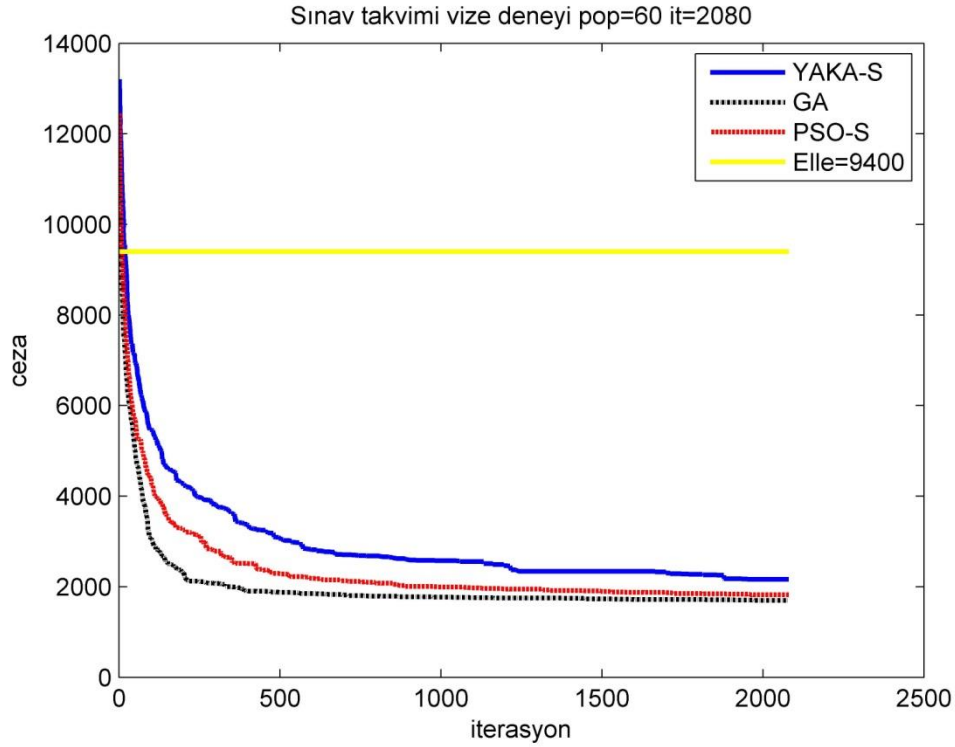
Şekil 4.5 Vize deneyi 5

Çizelge 4.8'de görülebileceği gibi GA ve PSO-S sabit bir seviyede performans göstermektedirler. Her üç yöntemde önceki deneydekine benzer ortalamada sonuçlar göstermektedir. GA en iyi sonuçları üretmiştir. YAKA-S performans olarak gelişerek PSO-S'e yaklaşmıştır.

Çizelge 4.8 Vize deney 5 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	1682.5	1320	1900	176.9062	1725.0
YAKA-S	2346.0	1705	2930	303.3406	2342.5
PSO-S	2016.5	1700	2535	283.7452	1947.5
El ile	9400.0	9400	9400	0.0000	9400.0

Altıncı deneyde popülasyon sayısı 60 ve çevrim sayısı 2080 seçilmiştir. Şekil 4.6'da ilgili deneyin ortalama yakınsama değerleri verilmiştir. Şekil 4.6'da görüldüğü gibi en iyi sonuçlar GA yöntemi ile elde edilmiştir. Ardından PSO-S en iyi sonuçları vermiştir. YAKA-S ile elde edilen sonuçlar diğer yöntemlere göre daha az başarılıdır.



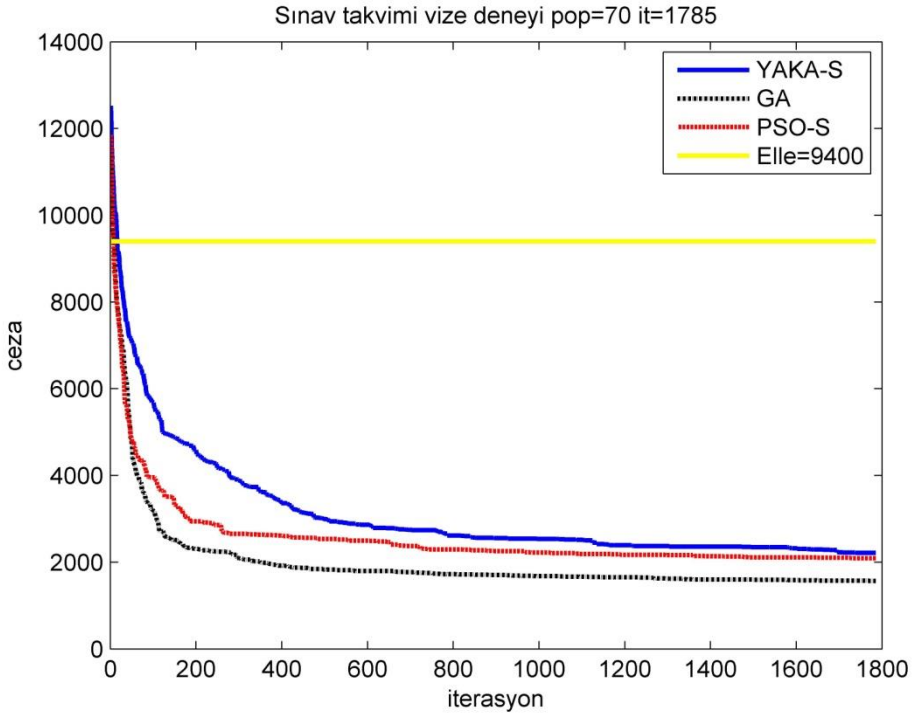
Şekil 4.6 Vize deneyi 6

Çizelge 4.9'da görülebileceği gibi. Her üç yöntemde önceki deneydekine benzer ortalamada sonuçlar göstermektedir. GA en iyi sonuçları üretmiştir. YAKA-S PSO-S'e yakınlaşmaktadır. Sonuçlardaki ufak değişimler çözüm uzayının rastgele aranmasından kaynaklanmaktadır.

Çizelge 4.9 Vize deney 6 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	1698.0	1535	1850	115.0169	1702.5
YAKA-S	2163.5	1595	2655	371.9546	2300.0
PSO-S	1820.0	1525	2345	244.9262	1812.5
El ile	9400.0	9400	9400	0.0000	9400.0

Yedinci deneyde popülasyon sayısı 70 ve çevrim sayısı 1785 seçilmiştir. Şekil 4.7'de ilgili deneyin ortalama yakınsama değerleri verilmiştir. Şekil 4.7'de görüldüğü gibi en iyi sonuçlar GA yöntemi ile elde edilmiştir. Ardından PSO-S en iyi sonuçları vermiştir. YAKA-S ile elde edilen sonuçlar diğer yöntemlere göre daha az başarılıdır.



Şekil 4.7 Vize deneyi 7

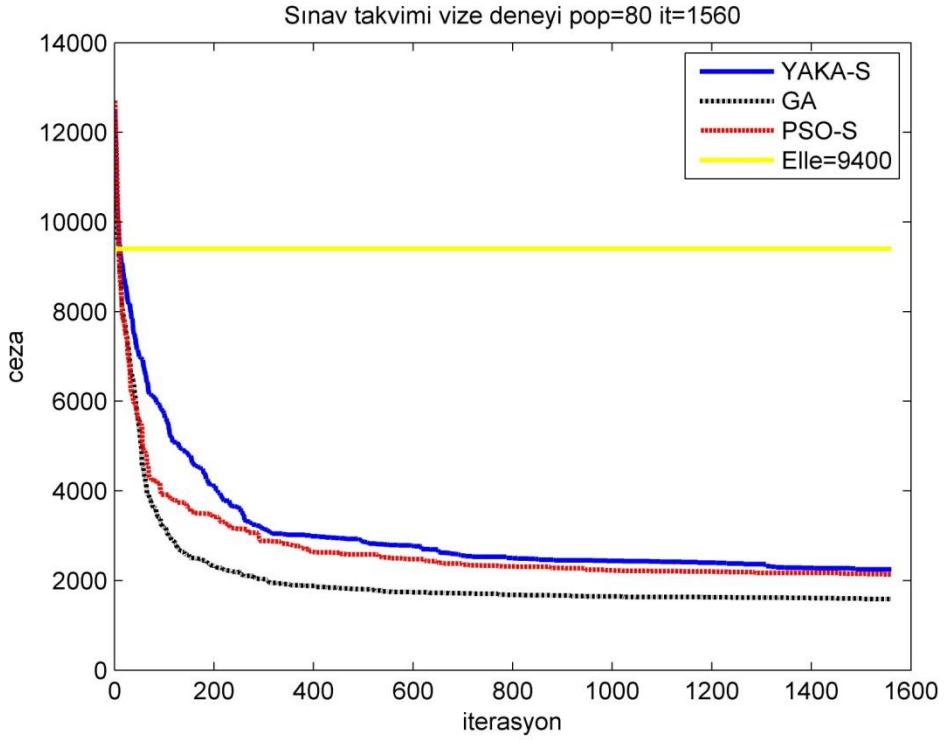
Çizelge 4.10'da görülebileceği gibi üç yöntemde önceki deneylerdekine benzer ortalamada sonuçlar vermektedir. GA algoritma yine en iyi sonuçları üretmiştir Her iki

yönteminde elde ettikleri ortalama ve minimum ceza puanlarından da anlaşılacağı gibi YAKA-S ile PSO-S hemen hemen aynı sonuçları üretmiştir.

Çizelge 4.10 Vize deney 7 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	1568.0	1375	1845	171.3054	1527.5
YAKA-S	2213.5	1575	2555	315.3749	2327.5
PSO-S	2092.0	1575	2510	316.1504	2112.5
El ile	9400.0	9400	9400	0.0000	9400.0

Sekizinci deneyde popülasyon sayısı 80 ve çevrim sayısı 1560 seçilmiştir. Şekil 4.8'de ilgili deneyin ortalama yakınsama değerleri verilmiştir. Şekil 4.8'de görüldüğü gibi en iyi sonuçlar GA yöntemi ile elde edilmiştir. Ardından PSO-S en iyi sonuçları vermiştir. YAKA-S ile elde edilen sonuçlar diğer yöntemlere göre daha az başarılıdır.



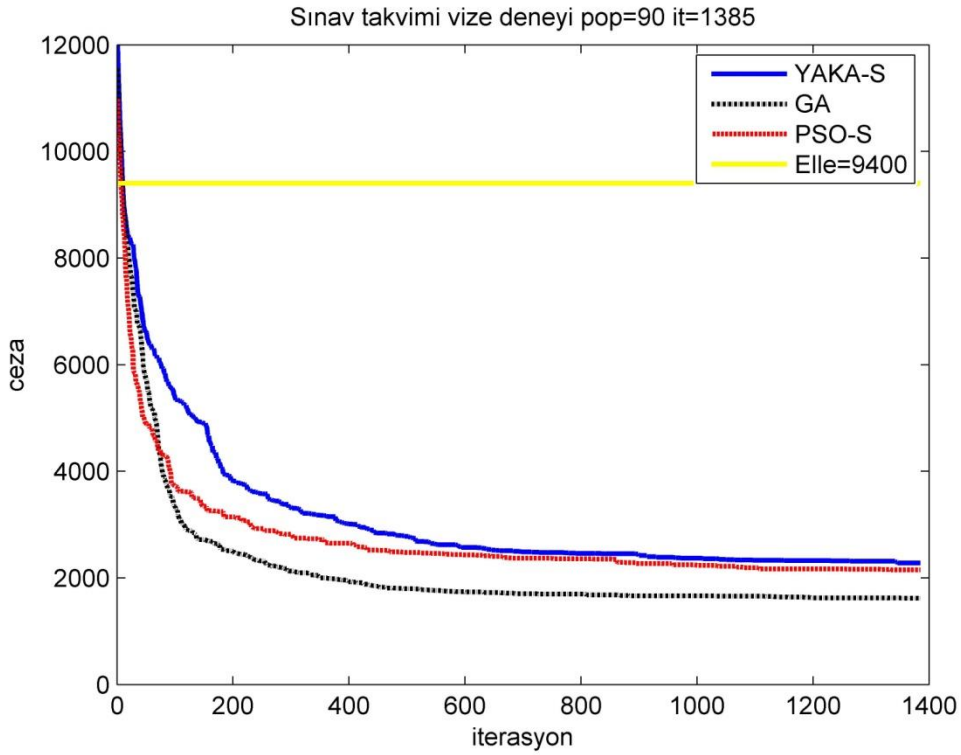
Şekil 4.8 Vize deneyi 8

Çizelge 4.11'de görülebileceği gibi. Her üç yöntemde önceki deneylerdekine benzer ortalamada sonuçlar göstermektedir. GA algoritma en iyi sonuçları üretmiştir. YAKA-S ile PSO-S hemen hemen aynı sonuçları üretmişlerdir.

Çizelge 4.11 Vize deney 8 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	1584.0	1420	1710	93.7727	1587.5
YAKA-S	2245.0	1600	2530	295.6161	2337.5
PSO-S	2135.5	1540	3205	510.7372	2015.0
El ile	9400.0	9400	9400	0.0000	9400.0

Dokuzuncu deneyde popülasyon sayısı 90 ve çevrim sayısı 1385 seçilmiştir. Şekil 4.9'da ilgili deneyin ortalama yakınsama değerleri verilmiştir. Şekil 4.9'da görüldüğü gibi en iyi sonuçlar GA yöntemi ile elde edilmiştir. Ardından PSO-S ve YAKA-S ile benzer sonuçlar elde edilmiştir.



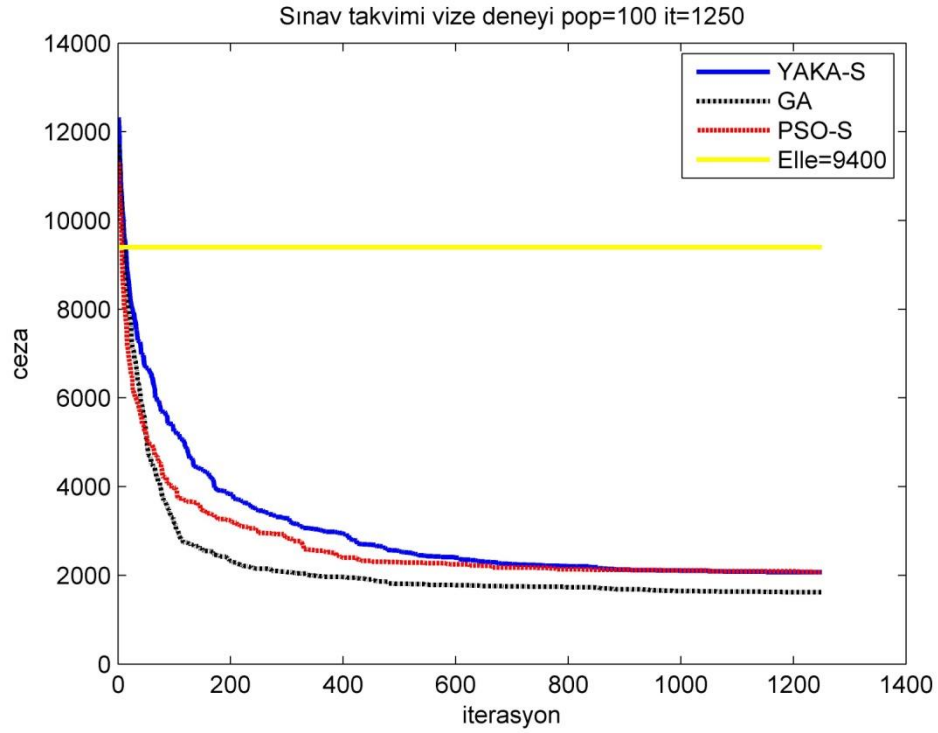
Şekil 4.9 Vize deneyi 9

Çizelge 4.12'de görülebileceği gibi. Her üç yöntemde önceki deneylerdekine benzer ortalamada sonuçlar vermektedir. GA yöntemi en iyi sonuçları üretmiştir. YAKA-S ile PSO-S hemen hemen aynı sonuçları üretmiştir.

Çizelge 4.12 Vize deney 9 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	1618.0	1450	1915	142.3259	1610.0
YAKA-S	2281.0	1705	2505	223.4676	2317.5
PSO-S	2148.5	1600	2765	420.0333	2065.0
El ile	9400.0	9400	9400	0.0000	9400.0

Onuncu deneyde popülasyon sayısı 100 ve çevrim sayısı 1250 seçilmiştir. Şekil 4.10'da ilgili deneyin ortalama yakınsama değerleri verilmiştir. Şekil 4.10'da görüldüğü gibi en iyi sonuçlar GA. Ardından YAKA-S ve en son PSO-S ile elde edilmiştir.



Şekil 4.10 Vize deneyi 10

Çizelge 4.13'de görülebileceği gibi. Her üç yöntemde önceki deneylerdekine benzer ortalamada sonuçlar üretmiştir. GA algoritma en iyi sonuçları üretmiştir. YAKA-S ile PSO-S hemen hemen aynı sonuçları üretmiştir. Ancak bu sefer YAKA-S yöntemi PSO-S'den daha iyi sonuçlar üretmiştir.

Çizelge 4.13 Vize deney 10 sonuçlarının kıyaslanması

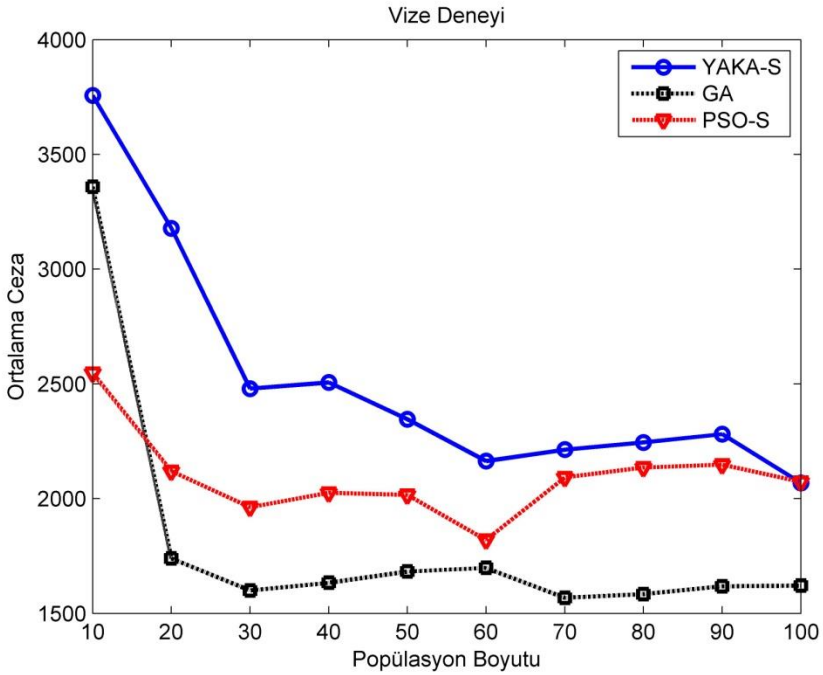
	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	1621.0	1445	1795	100.5208	1625
YAKA-S	2068.5	1275	2375	382.5357	2265
PSO-S	2073.0	1550	3050	427.56545	1980
El ile	9400.0	9400	9400	0.0000	9400

Çizelge 4.14'te üç yönteme ait ortalama ceza puanları yapılan vize deneyleri için toplu olarak gösterilmiştir.

Çizelge 4.14 Vize deney sonuçlarının ortalamalarının toplu kıyaslanması

GA	**YAKA-S**	**PSO-S**
3359	3757	2547
1739.5	3178.0	2120.0
1600.0	2479.5	1962.5
1633.5	2506.0	2025.5
1682.5	2346.0	2016.5
1698.0	2163.5	1820.0
1568.0	2213.5	2092.0
1584.0	2245.0	2135.5
1618.0	2281.0	2148.5
1621.0	2068.5	2073.0

Şekil 4.11'de popülasyon boyutunun yöntemlerin başarısına etkisi toplu olarak gösterilmiştir.



Şekil 4.11 Popülasyon boyutunun vize sınavında yöntemlerin başarısına etkisi

Şekil 4.11 ‘de de görülebileceği gibi GA ve PSO-S yöntemleri için ideal popülasyon boyutu 60 ila 70 ‘dir. Popülasyon boyutunun bu miktarlardan daha fazla olması ile birlikte yöntemlerin başarısı düşmektedir. Ayrıca popülasyon boyutu bu değerlere ulaşana kadar yöntemlerin başarısı genelde artma eğilimi göstermektedir. Ancak yöntemlerin çözüm uzayını rastgele araması elde edilen sonuçlarında kararsız olmasına sebep olmaktadır. Bu sebeple ortalama başarıda zaman zaman değişiklikler olmuştur. YAKA-S yöntemi için de benzer bir durum söz konusudur ancak son deneyde elde edilen başarı artışı nedeniyle ideal popülasyon boyutunun anlaşılması için ilave deneyler yapılabilir.

Çizelge 4.15’te GA yöntemi ile elde edilen en düşük ceza puanına sahip vize çizelgelerinden biri örnek olarak gösterilmiştir. Bu çizelgede Çizelge 4.3’te gösterilen derslerin kısaltmaları kullanılmıştır.

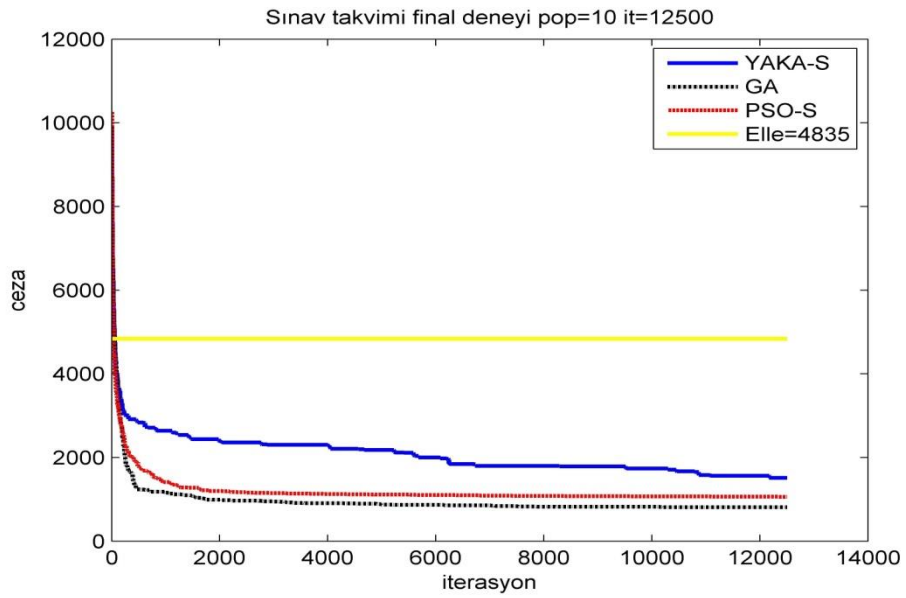
Çizelge 4.15 GA ile elde edilen vize çizelgesi

	Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi	Pazar
08:30 - 10:30					VERİ TB.II	SY. GÖR. İŞ.	
10:30 - 12:00	İŞLETİM S.	OTOMATA T.	NESNEYE Y. P.	İNGİLİZCE I	İNK. TAR. I	FİZİK I	T. ELEKTRONİK
12:00 13:30	YAPAY S. A. G.	E-İŞ/ E-TİC. G.	Edebiyat I'	VERİ YAP.	LOJİK T.	BİL. HUKUK	AYRIK MAT
13:30 15:30	MATEMATİK I	WEB PR.	MİKROİŞLEMCİ T. T.	SİS. PR.		N. Y. TAS. VE AN.	
15:30 17:00	DOSYA Y.		BULUT BİL. G.	B. GRAFİK İŞ. G.	ALG VE PROG. I	BİYOMED. V. İŞ.	
17:00 - 18:30	İŞ S. VE GÜV. I	VERİ T. I	LİNEER CEBİR	UZMAN SİS.	EVRİMSEL H. G.	BİL. MÜH. GİRİŞ	BİL. ORG.

4.2.1. Final sınavı deneyleri

Final sınavı için Çizelge 4.1'deki veri kümesi ve Çizelge 4.2'deki parametre bilgileri kullanılarak farklı çevrim ve popülasyon büyüklerinde 10'ar adet final deneyi gerçekleştirilmiştir. Kullanılan popülasyon büyüklüğü ve çevrim sayısı parametreleri vize sınavı deneylerinde olduğu gibi 125.000 fes sayısına göre değişiklik göstermektedir.

Şekil 4.12'de yakınsama çizelgesi gösterilen birinci final sınavı deneyinde popülasyon sayısı olarak 10 çevrim sayısı olarak 12500 seçilmiştir. Şekil 4.12'de görüldüğü gibi en iyi sonuçlar GA yöntemi ile elde edilmiştir. Ardından PSO-S en iyi sonuçları vermiştir. Daha sonra YAKA-S ile elde edilmiştir.



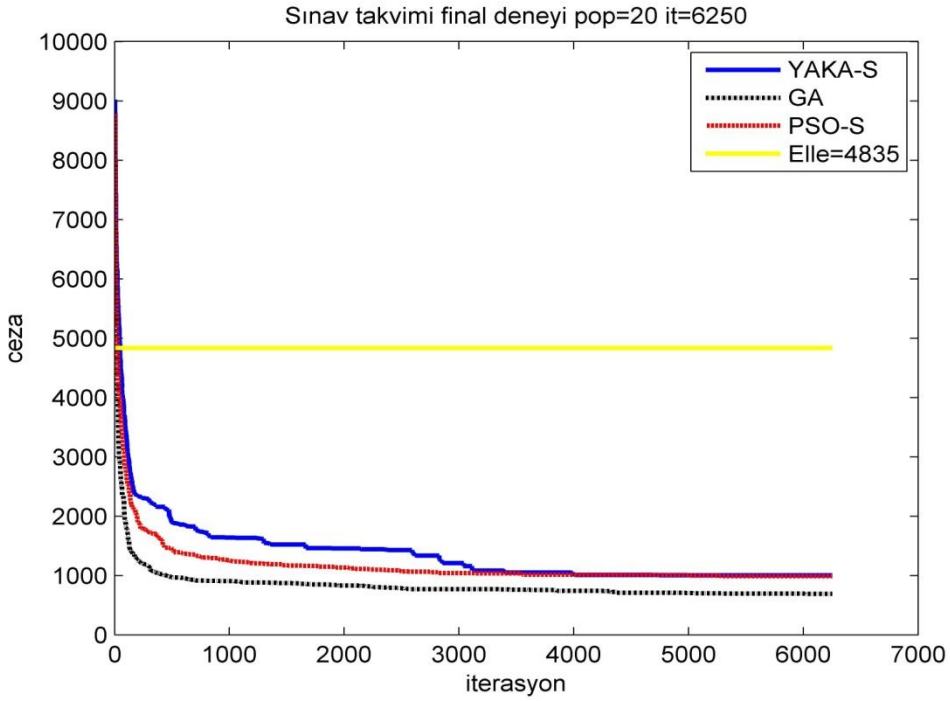
Şekil 4.12 Final deneyi 1

Vize deneylerinde olduğu gibi en iyi sonuçları GA yöntemi vermektedir. Vize deneyinin birincisinde olduğu gibi PSO-S algoritması popülasyon sayısının küçük olduğu bu deneyde çok iyi bir performans göstermektedir. Bunun başlıca sebebi daha önce belirttiğimiz gibi rastgele değiştirilen derslerin boyutu büyük olmasının yönteme katmış olduğu çeşitlilik yeteneğidir. GA yöntemi ise popülasyon sayısı küçük olmasına rağmen final sınavı nispeten daha büyük bir arama uzayında yer aldığı için daha iyi performans sergilemiştir. YAKA-S yöntemi ise kararsız performansını bu deneyde de sürdürmüştür. Çizelge 4.16'da Final deney 1 sonuçlarının kıyaslanması verilmiştir.

Çizelge 4.16 Final deney 1 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	810.0	700	925	71.870	825.0
YAKA-S	1511.5	905	2115	445.034	1450.0
PSO-S	1056.0	890	1225	129.460	1027.5
El ile	4835.0	4835	4835	0.000	4835.0

İkinci deneyde popülasyon sayısı 20 çevrim sayısı 6250 olarak seçilmiştir. Deneye ait yakınsama çizelgesi Şekil 4.13'te gösterilmiştir. Şekil 4.13'te görüldüğü gibi en iyi sonuçlar GA ardından PSO-S ve en son YAKA-S ile elde edilmiştir.



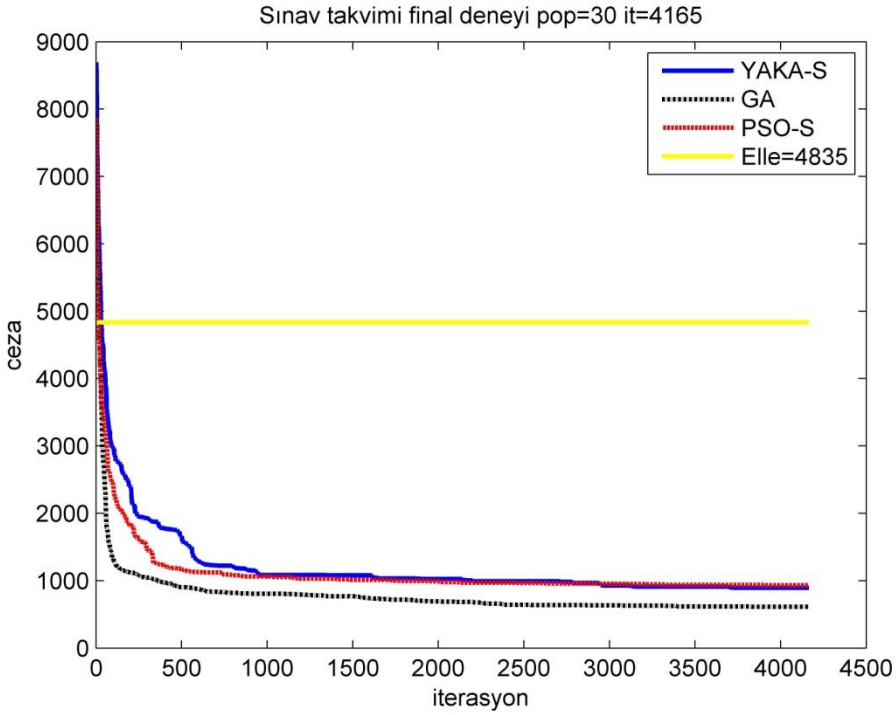
Şekil 4.13 Final deneyi 2

Final deneyi 1'de olduğu gibi en iyi sonuçları GA yöntemi vermektedir. Önceki deneydeki gibi PSO-S algoritması popülasyon sayısının küçük olduğu bu deneyde iyi performansına devam etmektedir. YAKA-S yöntemi ise karasız performansını bu deneyde de sürdürerek başarı olarak PSO-S algoritmasına yaklaşmaktadır. Minimum ceza puanı olarak YAKA-S yöntemi PSO-S'den daha düşük olsa da kararsız çalışma profili nedeniyle ortalama başarı olarak daha düşük kalmıştır. Çizelge 4.17'de Final deney 2 sonuçlarının kıyaslanması

Çizelge 4.17 Final deney 2 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	688.0	600	775	55.7374	700.0
YAKA-S	1001.5	705	1310	183.788	1032.5
PSO-S	988.0	880	1215	114.3143	945.0
El ile	4835.0	4835	4835	0.0000	4835.0

Üçüncü deneyde popülasyon sayısı 30 çevrim sayısı 4165 olarak seçilmiştir. Deneye ait yakınsama çizelgesi şekil 4.14'te gösterilmiştir. Şekil 4.14'te görüldüğü gibi en iyi sonuçlar GA ardından YAKA-S ve en son PSO-S ile elde edilmiştir.



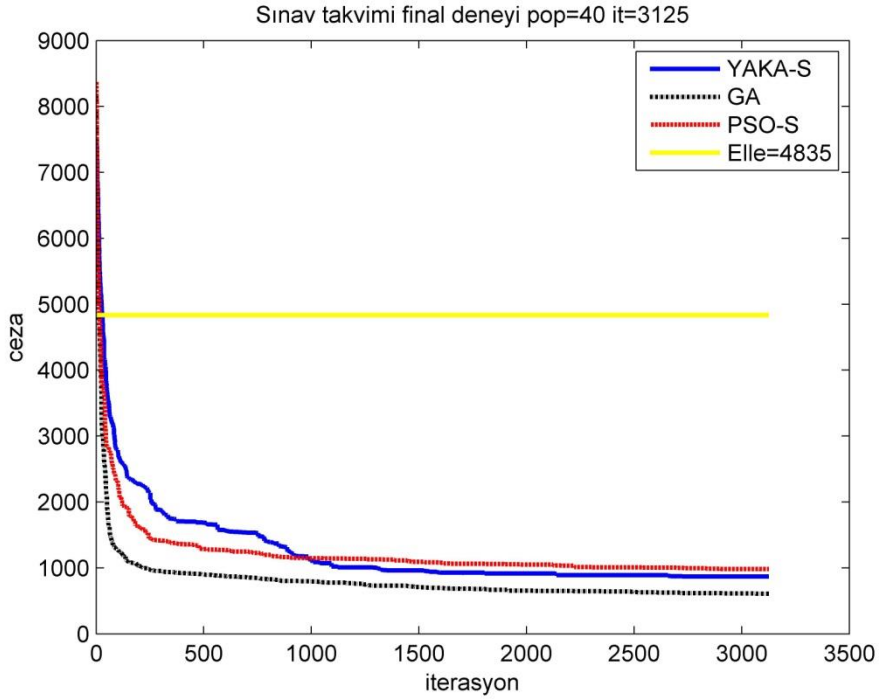
Şekil 4.14 Final deneyi 3

Yine en iyi sonuçları GA yöntemi vermiştir. Genelde minimum ürettiği değere yakın sonuçlar üretmektedir. YAKA-S yöntemi çözüm uzayını araştırma yeteneğindeki üstünlüğü sayesinde popülasyon sayısının artması ile birlikte PSO-S yönteminden daha iyi sonuçlar üretmeye başlamıştır. Ancak minimum ceza puanı ve standart sapma değerlerinden de görülebileceği gibi PSO-S'e göre daha karasız sonuçlar üretmektedir. Çizelge 4.18'de Final deney 3 sonuçlarının kıyaslanması verilmiştir.

Çizelge 4.18 Final deney 3 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	610.0	600	675	24.1522	600
YAKA-S	892.0	1205	1340	193.4941	855
PSO-S	931.5	805	1110	112.8679	885
El ile	4835	4835	4835	0.0000	4835

Dördüncü deneyde popülasyon sayısı 40 çevrim sayısı 3125 olarak seçilmiştir. Deneye ait yakınsama çizelgesi Şekil 4.15'de gösterilmiştir. Şekil 4.15'de görüldüğü gibi en iyi sonuçlar GA ardından YAKA-S ve en son PSO-S ile elde edilmiştir.



Şekil 4.15 Final deneyi 4

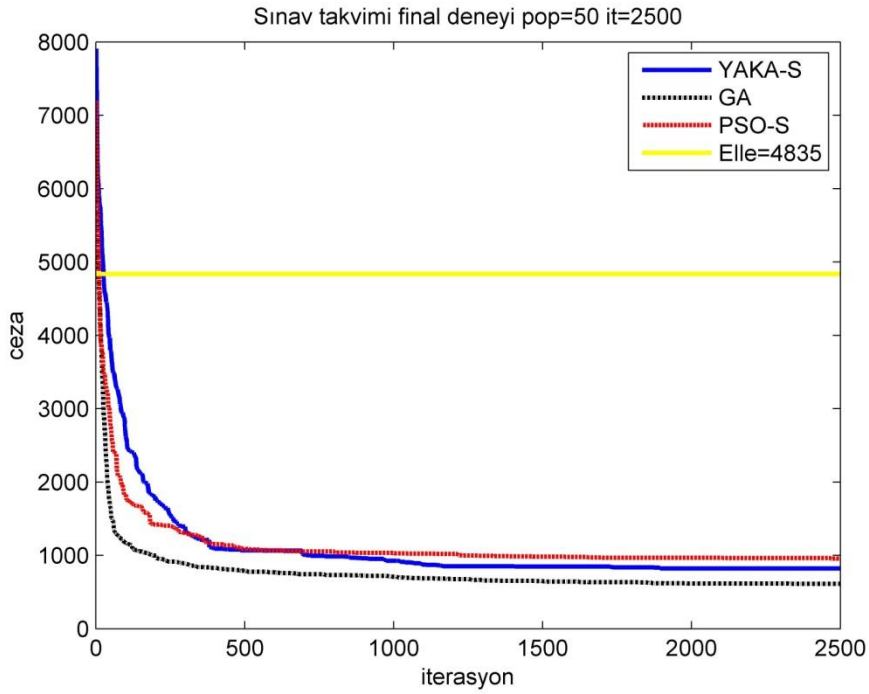
GA yöntemi her zamanki gibi en iyi sonuçlar üreten yöntemdir. Genelde minimum ürettiği değere yakın sonuçlar üretmektedir. YAKA-S yöntemi sömürme yeteneğindeki üstünlüğü sayesinde PSO-S yönteminden daha iyi sonuçlar üretmeye devam etmektedir. Minimum ceza puanı ve standart sapma bilgilerinde daha kararlı değerler elde etmiştir. Çizelge 4.19'da Final deney 4 sonuçlarının kıyaslanması verilmiştir.

Çizelge 4.19 Final deney 4 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	607.5	600	650	16.8737	600.0
YAKA-S	869.0	735	1005	75.8946	857.5
PSO-S	981.5	805	1310	167.3660	922.5
El ile	4835.0	4835	4835	0.0000	4835

Beşinci final deneyinde popülasyon sayısı 50 çevrim sayısı 2500 olarak seçilmiştir. Deneye ait yakınsama çizelgesi Şekil 4.16'da gösterilmiştir. Şekil 4.16'da

görüldüğü gibi en iyi sonuçlar GA ardından YAKA-S ve en son PSO-S ile elde edilmiştir.



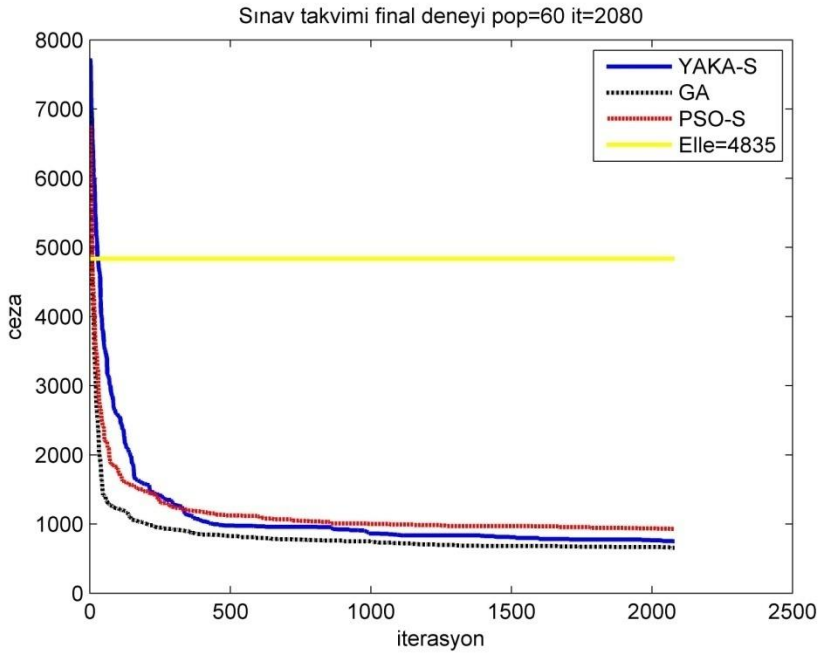
Şekil 4.16 Final deneyi 5

GA yöntemi her zamanki gibi en iyi sonuçlar üreten yöntemdir. PSO-S ve YAKA-S performanslarını arttırmaktadırlar. GA algoritma ise sabit bir performans göstererek kendini geliştirememektedir. Bunun sebebi bulunan çözümün bu deney için optimum çözüm olması olabilir. Çizelge 4.20'de Final deney 5 sonuçlarının kıyaslanması verilmiştir.

Çizelge 4.20 Final deney 5 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	610	600	700	31.6227	600.0
YAKA-S	820	710	930	66.5832	832.5
PSO-S	954	830	1240	121.4450	932.5
El ile	4835	4835	4835	0.0000	4835.0

Altıncı final deneyinde popülasyon sayısı 60 çevrim sayısı 2080 olarak seçilmiştir. Deneye ait yakınsama çizelgesi Şekil 4.17'de gösterilmiştir. Şekil 4.17'de görüldüğü gibi en iyi sonuçlar GA ardından YAKA-S ve en son PSO-S ile elde edilmiştir.



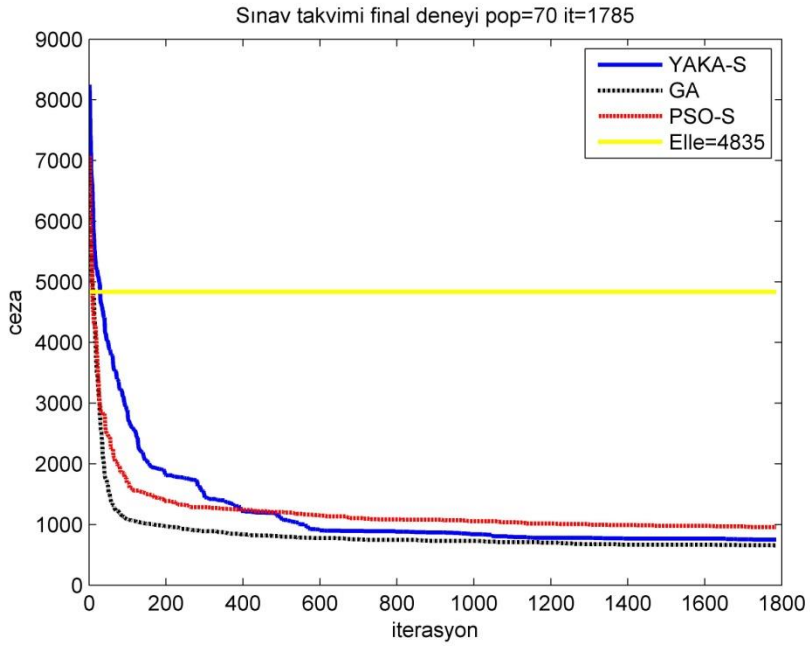
Şekil 4.17 Final deneyi 6

GA yöntemi her zamanki gibi en iyi sonuçlar üretmiştir fakat performansı sabit devam ettiği için YAKA-S performans olarak yaklaşmaya başlamıştır. Önceden de belirtildiği gibi bunun sebebi bulunan çözümün bu deney için optimum çözüm olması olabilir. YAKA-S sömürme yeteneği sayesinde üstünlüğünü korumaktadır. Çizelge 4.21'de Final deney 6 sonuçlarının kıyaslanması verilmiştir.

Çizelge 4.21 Final deney 6 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	656.5	600	815	79.2482	612.5
YAKA-S	749.5	600	890	74.9462	755.0
PSO-S	928.5	835	1035	68.2743	930.0
El ile	4835.0	4835	4835	0.0000	4835.0

Yedinci final deneyinde popülasyon sayısı 70 çevrim sayısı 1785 olarak seçilmiştir. Deneye ait yakınsama çizelgesi Şekil 4.18'de gösterilmiştir. Şekil 4.18'de görüldüğü gibi en iyi sonuçlar GA ardından YAKA-S ve en son PSO-S ile elde edilmiştir.



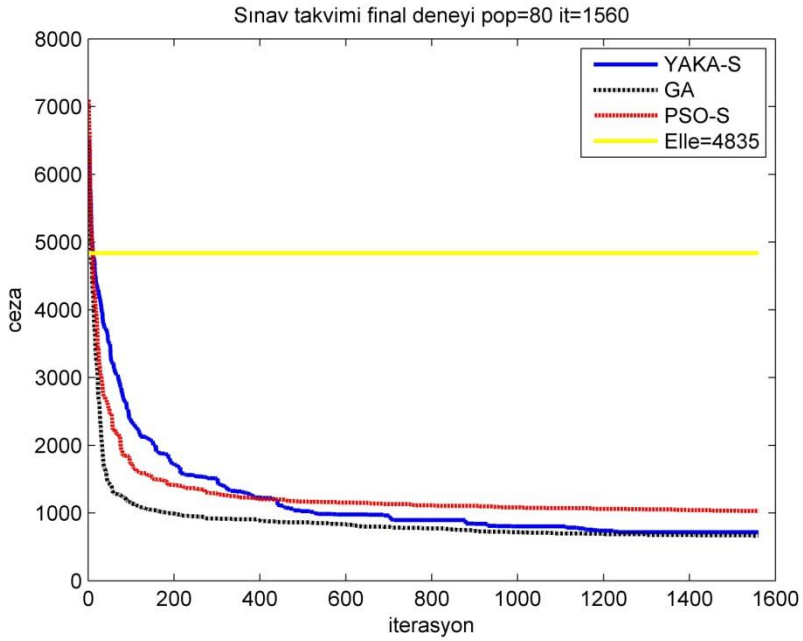
Şekil 4.18 Final deneyi 7

GA yöntemi en iyi sonuçlar üreten yöntemdir. Ancak popülasyon sayısının artmasıyla performansında küçük bir azalma görülmektedir. YAKA-S sabit bir performans göstermiştir. PSO-S ise sabit bir performans ile çalışmaya devam etmektedir. Çizelge 4.22'de Final deney 7 sonuçlarının kıyaslanması verilmiştir.

Çizelge 4.22 Final deney 7 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	655.5	600	825	79.4232	600
YAKA-S	750.5	655	835	58.2833	745
PSO-S	955.0	755	1160	127.2570	970
El ile	4835.0	4835	4835	0.0000	4835

Sekizinci final deneyinde popülasyon sayısı 80 çevrim sayısı 1560 olarak seçilmiştir. Deneye ait yakınsama çizelgesi Şekil 4.19'da gösterilmiştir. Şekil 4.19'da görüldüğü gibi en iyi sonuçlar GA ardından YAKA-S ve en son PSO-S ile elde edilmiştir.



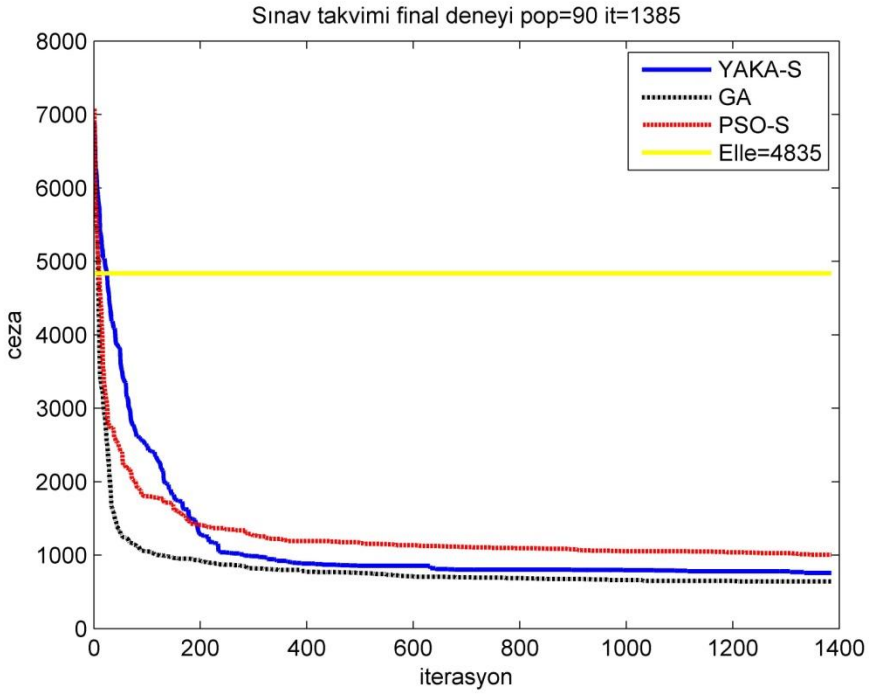
Şekil 4.19 Final deneyi 8

GA yöntemi en iyi olmaya devam etmektedir. YAKA-S'in performans artışı devam etmektedir. PSO-S ise sabit bir performans ile çalışmaya devam etmektedir. Çizelge 4.23'de Final deney 8 sonuçlarının kıyaslanması verilmiştir.

Çizelge 4.23 Final deney 8 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	665.0	600	750	57.9750	700.0
YAKA-S	715.5	630	810	65.8048	692.5
PSO-S	1031.5	850	1240	130.4703	1032.5
El ile	4835.0	4835	4835	0.0000	4835

Dokuzuncu final deneyinde popülasyon sayısı 90 çevrim sayısı 1385 olarak seçilmiştir. Deneye ait yakınsama çizelgesi Şekil 4.20'de gösterilmiştir. Şekil 4.20'de görüldüğü gibi en iyi sonuçlar GA ardından YAKA-S ve en son PSO-S ile elde edilmiştir.



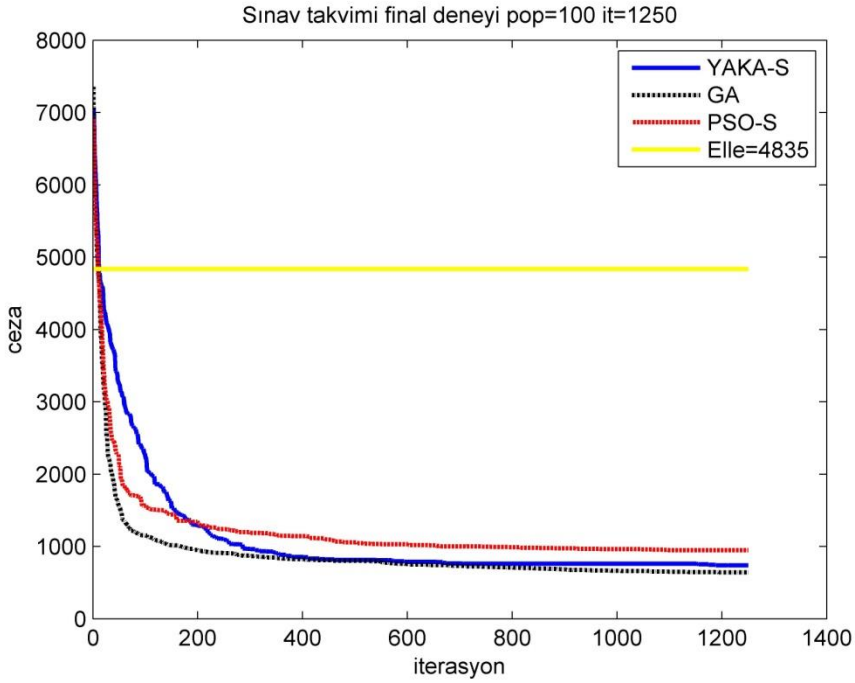
Şekil 4.20 Final deneyi 9

GA yöntemi en iyi olmaya devam etmektedir. YAKA-S'in performansında bir azalma gözlenmiştir. PSO-S diğer yöntemlere göre daha az başarı göstermiştir. Çizelge 4.24'te Final deney 9 sonuçlarının kıyaslanması verilmiştir.

Çizelge 4.24 Final deney 9 sonuçlarının kıyaslanması

	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	640	600	700	51.6397	600.0
YAKA-S	758	655	965	87.3117	742.5
PSO-S	1004	850	1235	139.7776	960.0
El ile	4835	4835	4835	0.0000	4835.0

Onuncu ve son final deneyinde popülasyon sayısı 100 çevrim sayısı 1250 olarak seçilmiştir. Deneye ait yakınsama çizelgesi Şekil 4.21'de gösterilmiştir. Şekil 4.21'de görüldüğü gibi en iyi sonuçlar GA ardından YAKA-S ve en son PSO-S ile elde edilmiştir.



Şekil 4.21 Final deneyi 10

GA yöntemi en iyi olmaya devam etmektedir. 3 yöntemde önceki deneylere benzer performanslar göstermektedirler. Çizelge 4.25'te Final deney 10 sonuçlarının kıyaslanması verilmiştir.

Çizelge 4.25 Final deney 10 sonuçlarının kıyaslanması

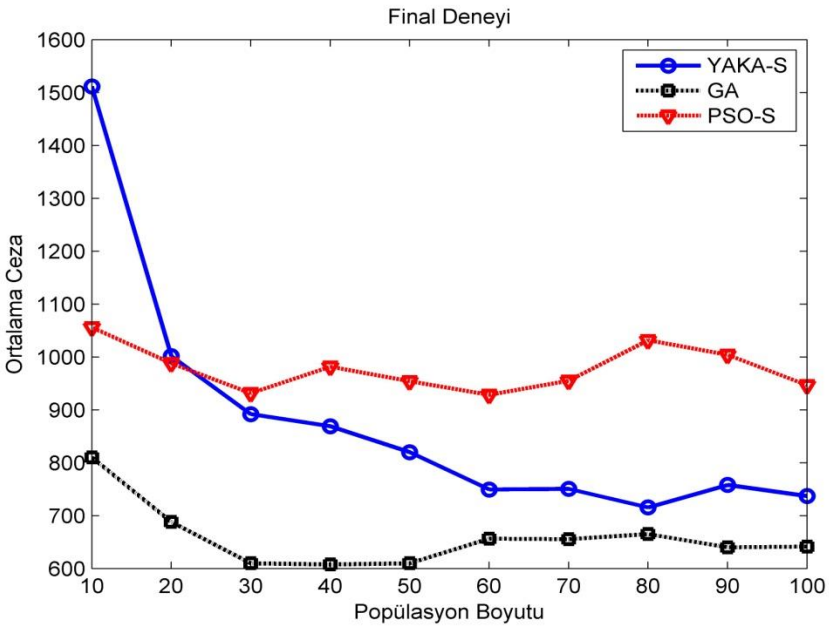
	Ortalama	**Minimum**	**Maksimum**	**Standart Sapma**	**Medyan**
GA	641.5	600	775	61.4658	600.0
YAKA-S	737.0	655	835	58.5567	732.5
PSO-S	946.5	700	1110	120.4632	950.0
El ile	4835.0	4835	4835	0.0000	4835.0

Çizelge 4.26'da üç yönteme ait ortalama ceza puanları bütün final deneyleri için toplu olarak gösterilmiştir. Çizelge 4.26 'da görüldüğü gibi yöntemler arasında en başarılı sonuçları genelde GA yöntemi üretmiştir.

Çizelge 4.26 Final deney sonuçlarının ortalamalarının toplu kıyaslanması

GA	YAKA-S	PSO-S
810.0	1511.5	1056.0
688.0	1001.5	988.0
610.0	892.0	931.5
607.5	869.0	981.5
610	820	954
656.5	749.5	928.5
655.5	750.5	955.0
665.0	715.5	1031.5
640	758	1004
641.5	737.0	946.5

Şekil 4.22 ‘de de görülebileceği gibi PSO-S yöntemi için ideal popülasyon boyutu 60 ila 70 ‘dir. Vize deneyi ve final deneyi birlikte göz önünde tutulursa YAKA-S yöntemi için ideal popülasyon boyutunun 80 olduğu görülmektedir. GA için ise ideal popülasyon boyutu 40 ila 50 civarıdır. Yöntem bu miktara kadar sürekli bir artış göstermiştir. Bu değerden sonra hafif daha başarısız olmaya başlamıştır. Ayrıca vize ve final deney sonuçlarının ikisinde birden görülebileceği gibi popülasyon boyutunun 10 gibi çok küçük bir değer olması yöntemlerin başarısını olumsuz etkilemektedir.



Şekil 4.22 Popülasyon boyutunun final sınavında yöntemlerin başarısına etkisi

Çizelge 4.27'de ve Çizelge 4.28'de GA yöntemi ile elde edilen en az ceza puanına sahip örnek bir final çizelgesi gösterilmiştir. Çizelge 4.27'de ve. Çizelge 4.28'de sırasıyla final sınavının birinci ve ikinci haftası gösterilmiştir. Bu iki çizelge tek bir sınav dönemi içindir.

Çizelge 4.27 GA ile elde edilen final çizelgesi birinci hafta

	Pazartesi	Salı	Çarşamba	Perşembe	Cuma
08:30 - 10:30	İŞLETİM S.			LİNEER CEBİR	
10:30 - 12:00		BİL. MÜH. GİRİŞ	FİZİK I		İNK. TAR. I
12:00 -13:30	N. Y. TAS. VE AN.	BİL. ORG.	Edebiyat I'		VERİ T. I
13:30 - 15:30	MATEMATİK I			AYRIK MAT	
15:30 - 17:00			EVRİMSEL H. G.		VERİ YAP.
17:00 - 18:30	İŞ S. VE GÜV. I	LOJİK T.	ALG VE PROG. I	İNGİLİZCE I	

Çizelge 4.28'de GA yöntemi ile elde edilen en düşük ceza puanına sahip örnek bir final çizelgesinin ikinci haftası gösterilmiştir.

Çizelge 4.28 GA ile elde edilen final çizelgesi ikinci hafta

	Pazartesi	Salı	Çarşamba	Perşembe	Cuma
08:30 - 10:30		BİYOMED. V. İŞ.	B. GRAFİK İŞ. G.	VERİ TB.II	
10:30 - 12:00	BULUT BİL. G.		DOSYA Y.	BİL. HUKUK	OTOMATA T.
12:00 -13:30	UZMAN SİS.				E-İŞ/ E-TİC. G.
13:30 - 15:30		SY. GÖR. İŞ.	T. ELEKTRONİK	MİKROİŞLEMCİ T. T.	
15:30 - 17:00					WEB PR.
17:00 - 18:30		NESNEYE Y. P.	YAPAY S. A. G.	SİS. PR.	

4.2.1. Deney sonuçlarının yorumlanması

Yapılan deneylerin sonuçlarına ait ilgili ortalama, standart sapma, medyan, maksimum ve minimum ceza bilgileri Çizelge 4.3'den itibaren Çizelge 4.22'ye kadar verilmiştir.

Yapılan deneylerde görülebileceği gibi hazırlanan yöntemler elle hazırlanan sınav çizelgelerinden daha etkili ve başarılıdırlar. Görevlilerin hazırladığı çizelgelerde bütün öğrenciler dikkate alınmamaktadır. Çünkü öğrencileri dikkate alarak sınav çizelgelemesi hazırlamak zaman talep eder ve zordur. Bu sebeple hazırlanan yöntemlerle daha kısa sürede daha etkili sınav çizelgeleri hazırlanmıştır.

Yapılan deneylerde Çizelgelerden de görülebileceği gibi en iyi sonuçlar genelde GA yöntemi ile elde edilmiştir. Bunun en büyük sebebi ise GA'nın doğal seçim ve çaprazlama operatörleri sayesinde çözüm uzayındaki global ve yerel arama yeteneğinin son derece iyi olmasıdır. PSO-S global en iyi kayıt bilgisini kullanarak global arama yeteneğine sahiptir ancak buna rağmen PSO-S algoritmasının çözüm uzayını araştırma yeteneği zayıf kalmıştır. YAKA-S algoritmasının en büyük dezavantajı ise çözümle alakalı önemli bilgileri unutmasıdır. Bu da arama uzayını bilinçsizce aramasına ve çözüme ait önemli bilgileri unutmasına sebep olmaktadır.

Yapılan ilk vize deneyinde en iyi sonuçları PSO-S yöntemi üretmiştir. YAKA-S ve GA yöntemleri popülasyon sayısının küçük olması nedeniyle kısa bir süre sonunda popülasyon içerisindeki çeşitliliğini kaybederek daha az başarılı sonuçlar üretmişlerdir. PSO-S algoritmasında rastgele değiştirilen derslerin boyutu fazla olduğu için çeşitliliğini daha uzun süre korumuş ve daha iyi sonuçlar üretmiştir. Ancak popülasyon sayısının artmasıyla birlikte bu çeşitlilik PSO-S algoritmasının yakınsama kabiliyetini azaltmıştır. GA ise artışla beraber daha iyi sonuçlar üretmiştir.

Vize deneylerinde popülasyonun boyutunun artması ile son çevrimde üretilen çizelgelerin kalitesi arasındaki fark azalmıştır. Bunun sebebi YAKA-S ve PSO-S algoritmalarında meydana gelen iyileşmenin yanı sıra GA yönteminin hızla yakınsaması ve çözüm için optimum değeri bulması da olabilir. Deneylerin yakınsama grafiğinden de görülebileceği gibi en hızlı yakınsayan yöntem GA'dır. Ancak ilerleyen çevrimlerde hem popülasyonun çeşitliliğinin azalması hem de alternatif iyi çözümlerin azalmasıyla yakınsama yavaşlayıp durmaktadır.

Kullanılan YAKA-S'in sömürme yeteneğinin PSO-S algoritmasına göre üstünlüğünü yapılan final deneyleri sonuçları ile de görülebilir. Final deneylerinde gün sayısı daha fazla olduğu için kullanılan komşuluk yöntemleri ile elde edilen yeni çözümlerin kısıt ihlali ihtimali daha düşüktür. Bu sebeple YAKA-S algoritması final sınavı deneyinden ortalama olarak daha iyi sonuçlar elde etmiştir.

Vize sınavı ise daha dar bir aralıkta yapıldığı için global arama yeteneğinin önemi artmaktadır. Bu durum elde edilen sınavların ortalama sonuçlarının gösterildiği Şekil 4.12'de ve diğer vize çizelgelerinin ortalama kolonunda görülebilir. Vize deneylerinde PSO-S algoritması global aramadaki üstünlüğü (çözüm ile ilgili önemli bilgileri unutmaması) sayesinde ortalama olarak daha başarılı olmuştur. Ancak Çizelge 4.12'nin minimum ceza puanı kolonunda görüldüğü gibi PSO-S arama algoritmasının yerel arama kabiliyetindeki zayıflık nedeni ile YAKA-S daha düşük minimum ceza puanlarını elde edebilmiştir.

YAKA-S arama algoritmasının çözüm uzayını bilinçsizce araması Çizelge 4.12'de ve diğer bazı vize sınavı çizelgelerinde çok iyi görülmektedir. Çizelge 4.2'de görüleceği gibi minimum ceza puanında GA'ya yakın değerler elde etmesine rağmen ortalama ve maksimum ceza puanlarında PSO-S algoritmasının gerisinde kalmıştır. Bu da PSO-S algoritmasının global en iyiye ait bilgileri sonraki çözümlerde kullanmasından kaynaklanmaktadır. Bu sayede ortalama olarak daha iyi çözümler elde edebilmiştir. YAKA-S algoritmasının sömürme yeteneğindeki üstünlüğü vize ve final sınavındaki minimum ceza puanı değerlerindeki üstünlükle anlaşılabilmektedir.

Yapılan deneylerde elde edilen sonuçlara göre kullanılan yöntemlerde global arama yeteneği görece en iyi olan GA'dır. Sömürme yeteneği görece en iyi olan ise YAKA-S'tir. YAKA-S'in GA'ya göre daha az başarılı sonuç vermesindeki en büyük etken bilinçsizce arama yaparak çözüme ait kıymetli bilgileri unutmasıdır. Bu sebeple ve global ve yerel arama özelliklerinin toplamıyla GA daha iyi sonuçlar üretmiştir. Bu ikisinin toplamının önemi vize ve final sınavlarının minimum değerlerinin birbirine yakın olması ancak ortalama değerlerde GA'nın üstün olması ile görülmektedir. PSO-S algoritması ise global aramadaki üstünlüğünü vize sınavında YAKA-S'den ortalama olarak daha iyi sonuçlar üretmesi ile göstermiştir. Ancak minimum ceza puanlarını elde etmesindeki eksiklik sömürme yeteneğindeki eksiklik olarak yorumlanabilir. Bu ikisinin birleşimi olarak GA'ya karşı zayıf kalmış ve GA daha iyi sonuçlar göstermiştir.

5. SONUÇ VE TARTIŞMA

Gerçekleştirilen bu çalışmada 3 farklı algoritma kullanarak sınav çizelgelemesi yapılmıştır. Bu amaçla Selçuk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği ‘ne ait öğrenci, öğretim üyesi ve ders bilgileri kullanılmıştır.

Bu çalışmada öğrencilere ihtiyaç duyacakları çalışma ve dinlenme sürelerini maksimum miktarda vererek başarılarını arttırmak hedeflenmiştir. Çalışmada Selçuk Üniversitesi Bilgisayar Mühendisliği Bölümü’ne ait sınav çizelgelemesi problemi çözüldüğü için ona ait kısıt ve altyapı göz önünde bulundurulmuştur. Bu amaçla ilgili matematiksel model tasarlanıp problem 3 farklı sezgisel algoritma kullanılarak çözülmüştür.

Geliştirilen algoritmalardan biri olan GA’da tamir fonksiyonuna ihtiyaç duyulmaması için ders sayısı kadar zaman bilgisinin tutulduğu bir kromozom gösterimi kullanılmıştır. Bu gösterimi kullanan ilgili çaprazlama ve mutasyon operatörleri kodlanmıştır.

Geliştirilen diğer bir yöntem olan PSO-S algoritmasında ise ayrık bir problem olan sınav çizelgelemesinin çözülebilmesi amacıyla çeşitli değişiklikler yapılmıştır ve GA’da kullanılan çaprazlama operatörüne benzeyen bir yöntem geliştirilmiştir. Bu yöntemde klasik PSO-S’den ilham alınarak global ve yerel en iyi parçacıkların bilgileri yeni konum hesaplanırken mevcut denklemlerin yerine kullanılmıştır. Bu sayede PSO-S algoritmasının ayrık bir problem olan sınav çizelgelemesi problemini çözmesi sağlanmıştır.

Üçüncü yöntem olan YAKA-S, PSO-S gibi sayısal optimizasyon problemlerinin çözümünde kullanılmaktadır. Bu yüzden üzerinde çeşitli değişiklikler yapılmıştır. Literatürde kullanılan üç komşuluk operatörü işçi ve gözcü arı aşamasında yeni yiyecek kaynağı hesaplanırken kullanılan denklemlerin yerini almıştır.

Tasarlanan yöntemler kıyaslandığında GA’nın görece daha iyi sonuçlar verdiği gözlemlenmiştir. YAKA-S algoritması ve PSO-S algoritması ayrık bir problem olan sınav çizelgelemesi probleminde çözüm sağlayabilecek algoritmalar olarak görülebilir. Bu algoritmalar diğer sezgisel algoritmalar ile melezlenerek elde edilen çözüm kalitesinin arttırılması sağlanabilir. Ayrıca bu çalışma bir arayüz ve veritabanı ile desteklenebilir. Bu modelin üstüne gözetmen ve sınıf kapasitesi gibi kısıtlar da dikkate alınarak yeni bir model inşa edilebilir. Ancak bu işlem yapılırken kısıtların artması

nedeniyle karşılaşılacak karmaşıklık ve performans faktörleri de göz önünde bulundurulmalıdır.

İleriki çalışmalar için GA ve PSO-S algoritması farklı yerel arama teknikleri ile melezlenerek yerel arama kabiliyeti arttırılabilir. Bu sayede algoritmaların elde ettikleri iyi çözümler türetilerek komşuluklarındaki daha iyi çözümlerle geliştirilebilir. Ancak bu yerel arama yöntemlerinin karmaşıklık, işlem yükü ve performans faktörleri de göz önünde bulundurulmalıdır.

İleriki çalışmalar için kullanılan PSO-S yöntemindeki global en iyi parçacıktan kopyalanan derslerin boyutu artırılarak ve azaltılarak yöntem üzerindeki etkileri incelenebilir.

İleriki çalışmalar için YAKA-S'e daha iyi bir global arama kabiliyeti kazandırmak için PSO-S'dekine benzer bir global en iyi mekanizması eklenebilir. Ancak bu işlemin global arama mekanizmasına yapacağı etkiler göz önünde bulundurulmalıdır.

İleriki çalışmalarda bu yöntemler bir veri tabanı ve ara yüzle desteklenecektir.

Yapılan çalışmada kullanılan parametrelerin yöntemlerin başarısındaki etkisi incelenmemiştir. İlerideki çalışmalarda bu parametrelerin etkileri de araştırılabilir.

6. KAYNAKLAR

Abdelhalim EA, El Khayat GA, 2016. A utilization-based genetic algorithm for solving the university timetabling problem (uga). Alexandria Engineering Journal, 55, 2, 1395-409.

Ahandani MA, Baghmisheh MTV, Zadeh MAB, Ghaemi S, 2012. Hybrid particle swarm optimization transplanted into a hyper-heuristic structure for solving examination timetabling problem. Swarm and Evolutionary Computation, 7, 21-34.

Ahmed H, Glasgow J, 2012. Swarm intelligence: concepts, models and applications. School Of Computing, Queens University Technical Report.

Aickelin U, Burke EK, Li J, 2009. An evolutionary squeaky wheel optimization approach to personnel scheduling. IEEE Transactions on evolutionary computation, 13, 2, 433-43.

Al-Betar MA, Khader AT, Doush IA, 2014. Memetic techniques for examination timetabling. Annals of Operations Research, 218, 1, 23-50.

Alzaqebah M, Abdullah S, 2011. Artificial bee colony search algorithm for examination timetabling problems. International Journal of Physical Sciences, 6, 17, 4264-72.

Alzaqebah M, Abdullah S, 2014. An adaptive artificial bee colony and late-acceptance hill-climbing algorithm for examination timetabling. Journal of Scheduling, 17, 3, 249-62.

Arogundade OT, Akinwale AT, Aweda OM, 2010. A genetic algorithm approach for a real-world university examination timetabling problem. International Journal of Computer Applications, 12, 5, 0975-8887.

Back T, Hammel U, Schwefel H-P, 1997. Evolutionary computation: Comments on the history and current state. IEEE transactions on Evolutionary Computation, 1, 1, 3-17.

Bansal JC, Sharma H, Jadon SS, 2013. Artificial bee colony algorithm: a survey. International Journal of Advanced Intelligence Paradigms, 5, 1-2, 123-59.

Berrada I, Ferland JA, Michelon P, 1996. A multi-objective approach to nurse scheduling with both hard and soft constraints. Socio-Economic Planning Sciences, 30, 3, 183-93.

Bilgin B, Özcan E, Korkmaz EE. An experimental study on hyper-heuristics and exam timetabling. International Conference on the Practice and Theory of Automated Timetabling, 394-412.

Bodenhofer U, (2003). Genetic algorithms: theory and applications, Lecture notes, Fuzzy Logic Laboratorium Linz-Hagenberg, Winter.

Boizumault P, Delon Y, Péridy L, 1996. Constraint logic programming for examination timetabling. The Journal of Logic Programming, 26, 2, 217-33.

Bolaji ALa, Khader A, Al-Betar M, Awadallah M, Thomas J. The effect of neighborhood structures on examination timetabling with artificial bee colony. Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling,(PATAT 2012), 29-31.

Burke E, Bykov Y, Newall J, Petrovic S, 2004. A time-predefined local search approach to exam timetabling problems. Iie Transactions, 36, 6, 509-28.

Burke E, Bykov Y, Petrovic S. A multicriteria approach to examination timetabling. International Conference on the Practice and Theory of Automated Timetabling, 118-31.

Burke EK, Eckersley AJ, McCollum B, Petrovic S, Qu R, 2010. Hybrid variable neighbourhood approaches to university exam timetabling. European Journal of Operational Research, 206, 1, 46-53.

Burke EK, Newall JP, 1999. A multistage evolutionary algorithm for the timetable problem. IEEE transactions on evolutionary computation, 3, 1, 63-74.

Burke EK, Qu R, Soghier A, 2014. Adaptive selection of heuristics for improving exam timetables. Annals of Operations Research, 218, 1, 129-45.

Bykov Y, Petrovic S, 2016. A Step Counting Hill Climbing Algorithm applied to University Examination Timetabling. Journal of Scheduling, 19, 4, 479-92.

Caramia M, Dell'Olmo P, Italiano GF, 2000. New algorithms for examination timetabling. Algorithm Engineering, 1982, 230-42.

Casey S, Thompson J. GRASPing the examination scheduling problem. International Conference on the Practice and Theory of Automated Timetabling, 232-44.

Cayirli T, Veral E, Rosen H, 2006. Designing appointment scheduling systems for ambulatory care services. Health care management science, 9, 1, 47-58.

Chu S-C, Chen Y-T, Ho J-H. Timetable scheduling using particle swarm optimization. Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on, 324-7.

Coley DA, 1999. An introduction to genetic algorithms for scientists and engineers, World Scientific Publishing Co Inc, p.

Colijn AW, Layfield C. Conflict reduction in examination schedules. 1995). Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling. 30th Aug-1st Sep, 297-307.

Cooper TB, Kingston JH. The complexity of timetable construction problems. International Conference on the Practice and Theory of Automated Timetabling, 281-95.

Corne D, Ross P, Fang H-L. Evolutionary timetabling: Practice, prospects and work in progress. In Proceedings of the UK Planning and Scheduling SIG Workshop, Strathclyde.

Cuupic M, Golub M, Jakobovic D. Exam timetabling using genetic algorithm. Information Technology Interfaces, 2009. ITI'09. Proceedings of the ITI 2009 31st International Conference on, 357-62.

Dahiya T, Kapil E, Goyal EA, 2014. Exam Timetabling Problem Using GA.

Daskalaki S, Birbas T, 2005. Efficient solutions for a university timetabling problem through integer programming. European Journal of Operational Research, 160, 1, 106-20.

David P. A constraint-based approach for examination timetabling using local repair techniques. International Conference on the Practice and Theory of Automated Timetabling, 169-86.

de Werra D, 1985. An introduction to timetabling. European journal of operational research, 19, 2, 151-62.

Duong T-A, Lam K-H. Combining Constraint Programming and Simulated Annealing on University Exam Timetabling. RIVF, 205-10.

Erden C, Kökçam AH, Demir Hİ, 2016. A Two-Phase Modeling with Genetic Algorithm for Solving Exam-Invigilator Timetabling Problem. International Journal of Engineering and Technology Research, 1, 1, 13-22.

Ersoy E, Özcan E, Uyar Ş. Memetic algorithms and hyperhill-climbers. Proc. of the 3rd Multidisciplinary Int. conf. on scheduling: theory and applications, P. Baptiste, G. Kendall, AM Kordon and F. Sourd, Eds, 159-66.

Fong CW, Asmuni H, McCollum B, 2015. A hybrid swarm-based approach to university timetabling. IEEE Transactions on Evolutionary Computation, 19, 6, 870-84.

Ganguli R, Roy S, 2017. A Study on Course Timetable Scheduling using Graph Coloring Approach. International Journal of Computational and Applied Mathematics, 12, 2, 469-85.

Gao W-f, Liu S-y, 2012. A modified artificial bee colony algorithm. Computers & Operations Research, 39, 3, 687-97.

Ghatei S, Panahi FT, Hosseinzadeh M, Rouhi M, Rezazadeh I, Naebi A, Ghatei Z, Khajei RP, 2012. A new hybrid algorithm for optimization using PSO and GDA. Journal of Basic and Applied Scientific Research, 2, 3, 2336-41.

Goldberg DE, 1989. Genetic algorithms in search, optimization, and machine learning, 1989. Reading: Addison-Wesley.

Goltz H-J, Matzke D. University timetabling using constraint logic programming. PADL, 320-34.

Grosan C, Abraham A, Chis M, 2006. Swarm intelligence in data mining. Swarm Intelligence in Data Mining, 1-20.

Grönkvist M, 2006. Accelerating column generation for aircraft scheduling using constraint propagation. Computers & Operations Research, 33, 10, 2918-34.

Haupt RL, Haupt SE, 2004. Practical genetic algorithms, John Wiley & Sons, p.

Heppner F, Grenander U, 1990. A stochastic nonlinear model for coordinated bird flocks. The ubiquity of chaos, 233-8.

Irene SFH, Deris S, Zaiton MHS. A study on PSO-based university course timetabling problem. Advanced Computer Control, 2009. ICACC'09. International Conference on, 648-51.

Jha SK, 2014. Exam timetabling problem using genetic algorithm. International Journal of Research in Engineering and Technology, 3, 649-54.

Jindal R, Rani M, 2016. Graph Coloring and its Implementation.

Kalaycı CB, 2008. Öğrenci başarısına odaklı sınav çizelgeleme modeli ve yazılım uygulaması, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü.

Karaboga D, (2005). An idea based on honey bee swarm for numerical optimization, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Karaboga D, Basturk B, 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization, 39, 3, 459-71.

Karaboga D, Gorkemli B, Ozturk C, Karaboga N, 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artificial Intelligence Review, 42, 1, 21-57.

Karaboğa D, 2014. Yapay Zeka Optimizasyon Algoritmalari.

Kaveh A, 2017. Particle Swarm Optimization. In: Advances in Metaheuristic Algorithms for Optimal Design of Structures. Eds. Cham: Springer International Publishing, p. 11-43.

Kendall G, Hussin N, 2005. An investigation of a tabu-search-based hyper-heuristic for examination timetabling. Multidisciplinary Scheduling: Theory and Applications, 309-28.

Kennedy R. J. and Eberhart, Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks IV, pages.

Kolonias V, Goulas G, Gogos C, Alefragis P, Housos E, 2014. Solving the examination timetabling problem in gpus. Algorithms, 7, 3, 295-327.

Lee S-Y, Kim Y-D, Lee H-J, 2009. Heuristic algorithm for a military training timetabling problem. Asia Pacific Management Review, 14, 3, 289-99.

Lovelace AL, 2010. On the complexity of scheduling university courses.

Lü Z, Hao J-K, 2010. Adaptive tabu search for course timetabling. European Journal of Operational Research, 200, 1, 235-44.

MacKinnon B, Sowden R, Dudley S, 2001. Sharing the skies: an aviation guide to the management of wildlife hazards. Transport Canada, Ottawa, Ontario, Canada.

Malkawi M, Hassan MA-H, Hassan OA-H, 2008. A New Exam Scheduling Algorithm Using Graph Coloring. International Arab Journal of Information Technology (IAJIT), 5, 1.

Man K-F, Tang K-S, Kwong S, 1996. Genetic algorithms: concepts and applications [in engineering design]. IEEE transactions on Industrial Electronics, 43, 5, 519-34.

Mansour N, Timani M, 2007. Stochastic search algorithms for exam scheduling. International Journal of Computational Intelligence Research, 3, 4, 353-61.

Matsushita H, Nishio Y, 2009. Network-structured particle swarm optimizer with various topology and its behaviors. Advances in Self-Organizing Maps, 163-71.

McCollum B, McMullan P, Parkes AJ, Burke EK, Qu R, 2012. A new model for automated examination timetabling. Annals of Operations Research, 194, 1, 291-315.

Mendes R, Kennedy J, Neves J, 2004. The fully informed particle swarm: simpler, maybe better. IEEE transactions on evolutionary computation, 8, 3, 204-10.

Mitchell M, 1998. An introduction to genetic algorithms, MIT press, p.

Moscato P, 1989. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report, 826, 1989.

Pillay N, Banzhaf W, 2010. An informed genetic algorithm for the examination timetabling problem. Applied Soft Computing, 10, 2, 457-67.

Qu R, K. Burke E, McCollum B, Merlot L, Y. Lee S, 2006. A Survey of Search Methodologies and Automated Approaches for Examination Timetabling, p.

Qu R, Pham N, Bai R, Kendall G, 2015. Hybridising heuristics within an estimation distribution algorithm for examination timetabling. Applied Intelligence, 42, 4, 679-93.

Reynolds CW, 1987. Flocks, herds and schools: A distributed behavioral model. ACM SIGGRAPH computer graphics, 21, 4, 25-34.

Ross P, Marín-Blázquez JG, Hart E. Hyper-heuristics applied to class and exam timetabling problems. Evolutionary Computation, 2004. CEC2004. Congress on, 1691-8.

Saharan S, Kumar R, 2016. Graph Coloring based Optimized Algorithm for Resource Utilization in Examination Scheduling. Appl. Math, 10, 3, 1193-201.

Sani H, Yabo M, 2016. Solving Timetabling problems using Genetic Algorithm Technique. International Journal of Computer Applications, 134, 15.

Sastry K, Goldberg DE, Kendall G, 2014. Genetic algorithms. In: Search methodologies. Eds: Springer, p. 93-117.

Shiau D-F, 2011. A hybrid particle swarm optimization for a university course scheduling problem with flexible preferences. Expert Systems with Applications, 38, 1, 235-48.

Smith KA, Abramson D, Duke D, 2003. Hopfield neural networks for timetabling: formulations, methods, and comparative results. Computers & industrial engineering, 44, 2, 283-305.

Tereshko V, Loengarov A, 2005. Collective decision making in honey-bee foraging dynamics. Computing and Information Systems, 9, 3, 1.

Welsh DJ, Powell MB, 1967. An upper bound for the chromatic number of a graph and its application to timetabling problems. The Computer Journal, 10, 1, 85-6.

Weng FC, Bin Asmuni H, 2013. An automated approach based on bee swarm in tackling university examination timetabling problem. Int J Electr Comput Sci, 13, 02, 8-23.

White GM, Xie BS. Examination timetables and tabu search with longer-term memory. International Conference on the Practice and Theory of Automated Timetabling, 85-103.

White GM, Xie BS, Zonjic S, 2004. Using tabu search with longer-term memory and relaxation to create examination timetables. European Journal of Operational Research, 153, 1, 80-91.

Wren A. Scheduling, timetabling and rostering—a special relationship? International Conference on the Practice and Theory of Automated Timetabling, 46-75.

Yaldır A, Baysal C, 2012. Evrimsel Hesaplama Tekniği Kullanarak Sınav Takvimi Otomasyon Sistemi Geliştirilmesi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 18, 2, 105-22.

Yang X-S, 2010. Engineering optimization: an introduction with metaheuristic applications, John Wiley & Sons, p.

7. ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	GÖKHAN SEYFİ
Uyruğu	**:**	T.C
Doğum Yeri ve Tarihi	**:**	YERKÖY/29.05.1987
Telefon	**:**	05537396090
Faks	**:**	-
e-mail	**:**	gokhanseyfi@selcuk.edu.tr

EĞİTİM

Derece		Adı, İlçe, İl	**Bitirme Yılı**
Lise	:	Mamak yabancı dil ağırlıklı lisesi	**2005**
Üniversite	:	Kırgızistan Türkiye manas üniversitesi	**2012**
Yüksek Lisans	:	Selçuk Üniversitesi Bilgisayar Mühendisliği	**Devam ediyor**
Doktora	:	-	

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2014-2015	Gümüşhane Üniversitesi	Araştırma Görevlisi
2015-(devam ediyor)	Selçuk Üniversitesi	Araştırma Görevlisi

UZMANLIK ALANI
Optimizasyon

YABANCI DİLLER
Rusça: Orta Seviye
Kırgızca : İleri Seviye
İngilizce İleri Seviye