

Sezgisel Algoritmalar Yardımıyla Ders Programı Optimizasyonu*

Araştırma Makalesi /Research Article

Ahmet YURTSAL¹
Oğuz KAYNAR²

ÖZ: Günümüzde çoğu eğitim kurumunda hazırlanan ders programı her dönem için yeniden yapılmaktadır. Bu işlemin her dönem tekrardan yapılması ve çoğu kurumda elle hazırlanıyor olması bu olayı zahmetli ve zaman alıcı bir iş haline getirmektedir. Bu çalışma için bir fakültenin gerçek verileri kullanılmış ve fakültenin bölümleri için uygun bir haftalık ders programı çizelgesi oluşturulmaya çalışılmıştır. Çalışmada problemin çözümü noktasında evrimsel hesaplama teknikleri olarak kabul edilen Genetik Algoritma, Parçacık Sürü Optimizasyonu ve Yapay Arı Kolonisi yöntemleri kullanılmış ve üç yöntem için de aynı veriler kullanılarak, mevcut yöntemlerin problemin çözümü üzerindeki performansları analiz edilmiştir. Çalışmada öğretim elemanı, öğrenci ve fakülte personelinin memnun edecek şekilde bütün kısıtlar dikkate alınmıştır. Geliştirilen sistemde kullanılan yöntemlerin parametreleri üzerinde değişiklikler yapılarak algoritmalar optimize edilmiştir. Yapılan deneyler sonucunda elde edilen ders programları kontrol edilerek fakülte için uygun ders programları elde edildiği görülmüştür. Ayrıca kullanılan algoritmalar, çalışma zamanı ve çözüme yakınsama açısından değerlendirilerek performansları karşılaştırılmıştır.

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

JEL Kodları: C61, C63, C88

Course Scheduling Optimization with the Help of Heuristic Algorithms

ABSTRACT: Nowadays, the curriculum prepared in most educational institutions is carried out for each term. The fact that this process is repeated every semester and that it is prepared manually in most institutions makes this event a laborious and time consuming task. The actual data of a faculty were used for this study and an appropriate weekly course schedule was prepared for the departments of the faculty. In the study, the methods of Genetic Algorithm, Particle Swarm Optimization and Artificial Bee Colony, which are accepted as evolutionary calculation techniques at the point of solution of the problem, were used and the performance of the existing methods on the solution of the problem was analyzed by using the same data for all three methods. In the study, all constraints were taken into consideration in a way that would satisfy the teaching staff, students and faculty staff. The algorithms are optimized by making changes on the parameters of the methods used in the developed system. It was observed that the curriculum obtained as a result of the experiments were controlled and appropriate curriculum for the faculty was obtained. In addition, the algorithms used were evaluated in terms of runtime and convergence, and their performance was compared.

Keywords: Genetic Algorithm, Particle Swarm Optimization, Artificial Bee Colony,

JEL Codes: C61, C63, C88

Geliş Tarihi / Received: 23/03/2021

Kabul Tarihi / Accepted: 18/01/2022

* Bu çalışma Ahmet YURTSAL'ın "Sezgisel Algoritmalar Yardımıyla Ders Programı Optimizasyonu" adlı Yüksek Lisans tezinden türetilmiştir.

¹ Arş. Gör., Hatay Mustafa Kemal Üniversitesi, İİBF, Yönetim Bilişim Sistemleri, Ahmet_yurtsal@hotmail.com, orcid.org/ 0000-0003-0523-3519.

² Prof. Dr., Sivas Cumhuriyet Üniversitesi, İİBF, Yönetim Bilişim Sistemleri, okaynar@cumhuriyet.edu.tr, orcid.org/ 0000-0003-2387-4053.

1. Giriş

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

Bir optimizasyon problemi olan zaman çizelgeleme problemleri kaynakların etkin dağılımı ile ilgilenen problemlerdir. Programlama sürecinde birçok kısıt dikkate alınmaktadır. Kaynaklar genellikle sınırlıdır ve iki görev aynı anda belirli bir kaynağı içermemelidir. Zaman çizelgeleme problemlerinin çoğu NP-zor problemler oldukları için bu problemler polinom zamanda çözülememektedir. (Sigl vd., 2003). Bu problemlerin çözümünde yapay zeka optimizasyon algoritmaları olarak adlandırılan Genetik Algoritma (Genetic Algorithm, GA), Tabu Araştırma (Tabu Search, TA), Karınca Kolonisi Optimizasyonu (Ant Colony Optimization, ACO), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO), Yapay Arı Kolonisi (Artificial Bee Colony, ABC) gibi sezgisel yöntemler yaygın olarak kullanılmakta ve problemlerin çözümünde başarılı sonuçlar vermektedir (Karaboğa, 2017).

Evrimsel Algoritmalar (EA), bilgisayar bilimlerinde, Yapay Zeka (YZ) çalışmalarının altında geçen bir konudur. Evrimsel bir algoritma, çözüm popülasyonundaki uygun olmayan çözümleri yok eden, uygun olan çözümlerin ise hayatta kalmasına izin veren ve istenilen çözüm bulununcaya kadar devam eden bir çalışma yapısına sahiptir. Zamanla başarılı çözümler en uygun çözümü bulmak için evrim geçirmektedir. Başka bir deyişle, EA Darwin'in evrim teorisinden esinlenilerek geliştirilmiş, karmaşık problemleri çözmek için biyolojik süreçleri taklit eden bilgisayar uygulamalarıdır (Goldberg, 1989).

Çalışmada kullanılan GA yöntemi, optimizasyon problemlerinde sıklıkla kullanılan EA'nın en popüler türüdür. GA biyolojiden ilham alınarak geliştirilen bir arama tekniğidir. Evrimsel biyolojiden gelen kavramları karmaşık optimizasyon problemlerini çözmek için bir teknik olarak kullanır. GA, potansiyel çözümlerin bulunduğu popülasyona, çaprazlama ve mutasyon işlemleri uygular ve üretilen yeni nesil çözümlerle tekrardan bir çözüm popülasyonu oluşturur. Çözümlerden biri problemi çözerse, süreç tamamlanır. Aksi takdirde, iyi performans gösteren çözümler kullanılarak yeni çözümler üretilir ve bu çözüm geliştirme süreci algoritma optimal çözüme ulaşana kadar veya belirlenen durdurma kriteri sağlanana kadar devam eder (Ryder, 2006).

Çalışmada kullanılan diğer bir yöntem ise PSO algoritmasıdır. Söz konusu bu algoritma temelde kuş ve balık sürülerindeki bireylerin birbirini geliştirmesine dayanan optimizasyon yöntemidir. Bu yöntem 1995 yılında Kennedy ve Eberhart

tarafından geliştirilen ve popülasyon temelli bir yöntemdir. PSO, kuş ve balık sürüleri gibi doğada etkileşim ve iletişim halinde olan canlıların davranışlarından ilham alınarak geliştirilmiştir. (Uysal, 2006).

Çalışmada problemin çözümü için kullanılan son yöntem ABC algoritmasıdır. Bu yöntem PSO yöntemine benzer şekilde sürü zekasını temel alan bir optimizasyon yöntemidir. Derviş Karaboğa tarafından geliştirilen bu yöntem doğada sürüler halinde hareket eden arıların yiyecek arama davranışları modellenerek geliştirilmiştir (Karaboga, 2005).

Bu çalışmada ilk olarak GA, PSO, ABC ve ders programı çizelgeleme problemi üzerine literatürde yapılan çalışmalar kısaca anlatılacak, sonrasında Ders Çizelgeleme Problemi (DÇP)'nin çözümünde kullanılan GA, PSO ve ABC yöntemlerinin özellikleri, terminolojisi ve nasıl çalıştığı üzerinde durulacaktır. Son olarak DÇP'nin çözümü aşamasında GA, PSO, ABC yöntemlerinin nasıl kullanıldığından bahsedilmiş ve yapılan deneylerin sonuçları değerlendirilecektir.

2. İlgili Çalışmalar

Bu çalışma ile ilgili daha önceden yapılmış çalışmaların literatür özeti Tablo 1'de verilmiştir.

Tablo 1: Literatür Tablosu

Yazar	Problem	Yöntem	Sonuç
Ghaemi vd. (2007)	Ders çizelgeleme problemi	Değiştirilmiş GA	Önerilen yöntemin problemin çözümünde iyi sonuçlar verdiği görülmüştür.
Küçüksille ve Tokmak (2011)	Ders çizelgeleme problemi	ABC algoritması	Geliştirilen algoritma ile sert kısıtlar üzerinde %100 başarı elde edilirken, esnek kısıtlarda %92,5 oranında başarı elde edilmiştir.
Bolaji vd. (2011)	ITC-2007 veri seti	Değiştirilmiş ABC algoritması	Önerilen ABC algoritmasının, iyi kalitede sonuçlar verebilmesine rağmen diğer yöntemlerle elde edilen en iyi sonuçlarla karşılaştırılmayacağı belirtilmiştir.
Oner vd. (2011)	Ders çizelgeleme problemi	Grafik düğüm renklendirme algoritması ve ABC algoritmasından oluşan hibrit bir algoritma	DeneySEL sonuçlar, önerilen hibrit algoritmanın verimli çözümler verdiğini göstermiştir.
Chen ve Shih (2013)	Ders çizelgeleme problemi	Yerel arama algoritması ve PSO algoritmasından oluşan hibrit bir algoritma	Hibrit yöntem standart PSO algoritmasına göre daha iyi sonuçlar vermiştir.
Kanović vd. (2014)	Gemi kilit kontrolü optimizasyonu	GA, PSO ve ABC	Elde edilen sonuçlara göre üç yöntemin de benzer sonuçlar gösterdiği görülmüş ve operasyon performansı iyileştirilmiştir.
Çolak (2015)	Ders çizelgeleme problemi	GA	Yapılan tez çalışmasında farklı popülasyon sayısı, çaprazlama oranı, mutasyon oranı ve iterasyon sayıları ile deneyler yapılmış ve uygun sonuçlar elde edilmiştir.
Bourouba ve Ladaci (2016)	Kesirli PID denetleyicinin parametrelerinin belirlenmesi	GA, PSO, ABC ve Kültür Algoritması	Deney sonuçlarına göre Kültür algoritması yönteminin diğer tekniklerden daha iyi sonuçlar verdiği görülmüştür.

Tablo 1: Literatür Tablosu (Devam)

Yazar	Problem	Yöntem	Sonuç
Yazdani vd. (2017)	Ders çizelgeleme problemi	Yapay bağıklık, GA ve benzetilmiş tavlama algoritmaları	Deneysel, yapay bağıklık algoritmasının diğer algoritmalarından daha iyi performans gösterdiğini göstermiştir.
Özkan (2019)	Ders çizelgeleme problemi	Farklı aşamalar için, tamsayılı doğrusal programlama, açgözlü arama, ileri arama algoritması, tabu arama ve benzetim tavlama	Problemin karmaşıklığı arttıkça tamsayılı doğrusal programlama problemi çözmekte zorlanmış, açgözlü aramanın ileri arama algoritmasından daha hızlı olduğu ve benzetim tavlama yönteminin tabu arama yönteminden daha iyi sonuçlar verdiği görülmüştür.
Muthiah vd. (2016)	İş atölyesi çizelgeleme	ABC ve PSO algoritmasından oluşan hibrit bir algoritma	Hibrit yöntem ile standart ABC ve PSO'dan daha iyi sonuçlar elde edilmiştir.
Kulkarni ve Desai (2016)	Kıyaslama Fonksiyonları	ABC ve PSO	ABC algoritmasının PSO'dan daha kesin optimizasyon sağladığı görülmüştür; ancak gecikmeli yakınsama problemi vardır. Öte yandan, PSO daha hızlı; ancak optimale ulaşma olasılığı nispeten daha düşük olduğu görülmüştür.
Akkan ve Güçlü (2018)	ITC-2007 veri seti	Standart GA ve tepe tırmanışı ve simüle tavlama algoritmalarını kullanan hibrit GA	Önerilen hibrit algoritma ile iyi sonuçlar elde edilmiştir.
Şamdan ve Yetgin (2018)	Ders çizelgeleme problemi	Değiştirilmiş ABC algoritması	Deneysel sonuçlar, önerilen yaklaşımın standart ABC yaklaşımından daha başarılı olduğunu göstermiştir.
Le vd. (2019)	Akıllı şehirlerdeki enerji verimliliği optimizasyonu	Yapay sinir ağları temelli GA, ABC PSO ve yayılımçı rekabet algoritması	Sonuçlar, GA-yapay sinir ağları modelinin en yüksek performansı sağladığı göstermiştir.
Hossain vd. (2019)	Ders çizelgeleme problemi	Değiştirilmiş PSO algoritması, GA ve harmoni arama algoritması	Deneysel sonuçlar, önerilen yöntemin GA ve harmoni arama algoritmasına kıyasla etkinliğini ve üstünlüğünü ortaya koymuştur.
Ba vd. (2019)	Ders çizelgeleme problemi	Değiştirilmiş GA	Deneysel, bu algoritmanın ders programı planlamasının tüm gereksinimlerini etkili bir şekilde karşılayabildiğini göstermiştir.
Chaudhari ve Thakkar (2019)	Gezgin satıcı problemi	ACO, PSO, ABC, GA ve ateş böceği algoritması	Deneysel sonuçların karşılaştırmalarına göre, ACO ve GA'nın ABC, PSO ve ateş böceği algoritmasından daha iyi performans gösterdiği görülmüştür.
Dener ve Calp (2019)	Merkezi sınav organizasyonları	GA	Yöntem problemin çözümünde etkili sonuçlar elde etmiştir.
Ramdania vd. (2019)	Ders çizelgeleme problemi	GA ve PSO	PSO uygunluk değerinin GA'dan daha iyi performans gösterdiğini, ancak GA yürütme süresinin PSO algoritmasından daha hızlı olduğu görülmüştür.
Naik vd. (2020)	Hidrotermal çizelgeleme problemi	Değiştirilmiş Sosyal grup optimizasyonu, GA, PSO, ABC ve diferansiyel evrim	Simülasyon sonuçları, değiştirilmiş sosyal grup optimizasyonu yönteminin daha iyi bir çözüm elde etme yeteneğine sahip olduğu görülmüştür.

3. Materyal ve Yöntem

Bu çalışmada deneysel sonuçlar, Intel (R) Core (TM) i5-4200U, 1.60-2.30 GHz işlemciye ve 4 GB RAM'e sahip dizüstü bilgisayar kullanılarak elde edilmiştir. Yöntemlerin probleme uygulanması Matlab programında gerçekleştirilmiştir.

3.1 Uygulanan Yöntemler

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

3.1.1 Genetik Algoritma

GA ilk olarak Michigan Üniversitesi'nde psikoloji ve bilgisayar bilimi uzmanı olan John Holland tarafından 1970'li yıllarda bulunmuş ve daha sonra çalışma arkadaşları ve öğrencileri ile birlikte geliştirilmiştir. Holland 1975 yılında yaptığı çalışmaları "Adaptation in Natural and Artificial Systems" adlı kitabında bir araya getirmiştir. Makine öğrenmesi üzerine çalışmalar yapan Holland, Darwin'in evrim teorisinden etkilenerek bunu bilgisayar ortamında kullanmayı düşünmüştür. Tek bir mekanik yapının öğrenme yeteneğini geliştirmek yerine, bu tür yapılardan oluşan topluluğun çiftleşme, çoğalma ve değişim vb. gibi genetik süreçlerden geçerek başarılı yeni bireyler (öğrenebilen) oluşturulabildiğini göstermiştir (Şahin, 2009). Holland'ın bu çalışmalarından sonra onun doktora öğrencisi olan David E. Goldberg, 1989 yılında alanında klasik olarak kabul edilen "Genetic Algorithms in Search, Optimization & Machine Learning" adlı kitabını yayınlamıştır. Goldberg kitabında yaptığı çalışmaları ile GA'nın daha iyi anlaşılmasını sağlamış ve birçok alanda uygulanabileceğini göstermiştir. Günümüze kadar bu yöntemle ilgili çeşitli çalışmalar yapılmış ve birçok alana uygulanmıştır.

3.1.2 Parçacık Sürü Optimizasyonu

PSO, evrimsel optimizasyon yöntemlerinden biridir. İlk olarak 1995 yılında Kennedy ve Eberhart tarafından geliştirilen popülasyon tabanlı bir optimizasyon yöntemidir. PSO, kuş sürüsü ve balık eğitimi gibi sosyal etkileşim ve iletişim benzetmesine dayanmaktadır. Temel olarak sürü zekâsına dayanan bir algoritma olduğu için, bir PSO algoritmasındaki üyeler en iyi performansla sahip olan grup liderini takip etme eğilimindedir (Kennedy ve Eberhart, 1995).

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

3.1.3 Yapay Arı Kolonisi Algoritması

ABC algoritması, bal arısı kolonilerinin kendilerine has olarak sürdürdükleri zeki tutumlarını örnek alarak, arıların besin ararlarken kullanmış oldukları yöntemlerden esinlenerek meydana getirilmiş bir optimizasyon algoritması olarak kabul görmektedir. Sürünün kendine özgü zekasına dayanan söz konusu algoritma, doğada sürü olarak hareket edebilen arıların besin bulmalarında sergilemiş oldukları tutumları temel alarak problemlere çözüm aramaktadır. Arı sürülerinin zeki davranış tutumları ve besin arama süreci içerisindeki tutumlarını modelleyen Karaboğa, ABC algoritmasını geliştirmiştir (Karaboga, 2005).

3.2 Problemin Çözümü

DÇP’de, yerine getirilmesi gereken bazı kısıtlar mevcuttur. Bu kısıtlar zorunlu ve esnek olarak ikiye ayrılır. DÇP’de zorunlu ve esnek kısıtlar belirlenirken dikkat edilmesi gereken nokta uygun bir ders programının hazırlanması için zorunlu kısıtların kesinlikle sağlanması, esnek kısıtların ise ders programın kalitesini artırmak için mümkün olduğunca sağlanmasıdır. Esnek kısıtların yeteri kadar sağlanmaması durumunda öğrenci ve öğretim elemanını memnun etmeyen bir ders programı ortaya çıkacaktır. Bu çalışmada belirlenen kısıtlar uygulamanın yapıldığı fakültenin özelliklerine göre hazırlanmıştır.

DÇP’nin çözümü için yapılan çalışmada GA, PSO ve ABC algoritması ayrı ayrı kodlanmıştır. Bazı yöntemlerin bazı problemlerin çözümünde daha uygun bir yapıda olduğu bilinmekte ve bazı yöntemlerin bir probleme uygulanabilmesi için yöntem üzerinde değişiklikler yapılması gerekmektedir. Çalışmamızda kullandığımız yöntemler literatüre uygun şekilde probleme uygulanabilir hale getirilmiştir. Her ne kadar üç algoritma birbirlerinden farklı gibi gözükseler de her üç algoritmanın da popülasyon temelli olmaları, iteratif bir yapıda var olan çözümleri sürekli iyileştirerek en iyi çözüme yakınsama gibi ortak çalışma prensiplerine sahip olmaları nedeniyle çoğu yönden benzerlik göstermektedir. Kullanılan algoritmalarından bağımsız olarak ders çizelgeleme problemin kodlanması ve uygunluk fonksiyonlarının belirlenmesi öncelikle ele alınmıştır.

3.2.1 Problemin Kodlaması

Bu çalışmada her üç yöntem içinde kullanılan kodlama yapısı, derslikler satır ve zaman dilimleri sütun olacak şekilde matris formunda oluşturulmuştur. Oluşturulan bu matrisin boyutu derslik*zaman periyodları şeklinde olacaktır.

A = Toplam kodlama boyutu

Z = Günlük zaman periyodu sayısı

C = Toplam derslik sayısı

G = Ders yapılan gün sayısı

$A = Z \times C \times G$

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

$40 \times 2 \times 5 = 400$ boyutunda bir kod yapısı oluşturulmuştur. Oluşturulan kodlama yapısında zaman periyodları blok şeklinde düşünölmüştür.

1. Blok: 08.00-12.00 saatleri arası

2. Blok: 13.00-17.00 saatleri arası

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

Derslik-Zaman matrisinin hücreleri ders numaralarını tutmaktadır. Ayrıca bir derslikte herhangi bir zaman periyodunda ders yapılmadığı zaman 0 değeri ile temsil edilmektedir. Oluşturulan kodlama yapısının ilk hali 40 satır ve 10 sütundan oluşan 0 matrisi şeklindedir. Sonrasında ders numaraları bu kodlama yapısına aktarılmaktadır. Bu ders numaralarının ve 0 değerlerinin her biri GA'da bir gen olarak düşünölür ve bir araya gelerek kromozom yapısını oluşturur. Bu kodlama yapısı sayesinde bazı kısıtlar ortadan kaldırılmıştır. Çözümü gösteren kod yapısının bir örneğı Şekil 1'de gösterilmiştir.

Şekil 1: Kodlama Yapısı

Derslikler \ Zaman periyodları	1. Blok		2. Blok		1. Blok		2. Blok		1. Blok		2. Blok	
	1. Blok	2. Blok	1. Blok	2. Blok	1. Blok	2. Blok	1. Blok	2. Blok	1. Blok	2. Blok		
1	12	14	33	21	44	0	0	131	19	42		
2	0	33	8	38	46	52	4	0	40	0		
3	0	76	0	0	0	0	91	0	1	220		
4	11	21	46	0	189	0	55	0	93	0		
5	61	0	33	6	0	71	0	111	0	0		
6	49	0	0	0	2	0	0	0	18	88		
7	22	8	0	219	0	0	64	0	199	0		
...	0	127	0	139	0	67	0	66	7	0		
...	79	0	0	77	0	0	124	0	0	3		
...	51	0	94	9	0	86	92	0	99	0		
40	0	28	99	0	5	0	101	0	14	0		

3.2.2. Uygunluk Değeri

Uygunluk değeri bir çözümün olası diğer çözümlere oranla ne derecede iyi ve uygun olduğunun bir göstergesidir. Uygunluk değeri, zaman çizelgeleme probleminin çözümünde kullanılan zorunlu ve esnek kısıtların ne kadar ihlal edildiğinin göstergesidir. Problemlerin çözümünde bireylerin uygunluk değerlerini hesaplayan bir uygunluk fonksiyonu vardır. Uygunluk fonksiyonu, her bireyin uygunluk değerini zorunlu ve esnek kısıtlara göre ayrı ayrı hesaplamaktadır.

Zaman çizelgeleme problemlerinde bireyin uygunluk değeri, bireyin ihlal ettiği kısıtların ceza puanları toplanarak hesaplanır. Bundan dolayı bireyin uygunluk değeri ne kadar küçükse çözüm için o kadar uygun olacaktır. Aşağıda bir bireyin uygunluk değerinin hesaplanması ve zorunlu ve esnek kısıtlara verilen ceza puanları Tablo 2 ve Tablo 3’de gösterilmiştir.

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

b = Birey

f(b) = Bireyin uygunluk değeri

n = Kısıt sayısı

C_j = j. kısıtın ceza puanı

Tablo 2: Zorunlu Kısıtlar

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

Tablo 3: Esnek Kısıtlar

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

3.2.3. Genetik Algoritma Yönteminin Kullanımı

Çalışmada GA'nın kodlama tipi olarak Değer Kodlama kullanılmıştır. Bu kodlama tipine uygun olarak da kromozom yapısı oluşturulmuş ve uygun başlangıç popülasyonu, seçim operatörü, çaprazlama operatörü ve mutasyon operatörü belirlenmiştir. Çalışmada oluşturulan kromozom yapılarının her biri bir ders programını temsil etmektedir. GA problemin çözümünde kromozomlardan oluşan bir başlangıç popülasyonu oluşturmaktadır. Algoritma başlamadan önce belirlenen popülasyon sayısı kadar başlangıç popülasyonunun kromozom sayısı olur ve algoritma çözüm arama sürecini bitirene kadar bu sayı sabit kalmaktadır. Başlangıç popülasyonu oluşturulduktan sonra çözüm arama süreci sonlanana kadar GA yöntemi ile kromozomlar üzerine uygunluk fonksiyonu ve 3 temel GA operatörü uygulanmıştır. Uygunluk fonksiyonu ile kromozomların yani ders programlarının her iterasyonda fakülte için ne kadar uygun oldukları değerlendirilmiş ve yaptığı kısıt ihlallerine göre ceza puanı verilmiştir. Bu değerlendirmeden sonra seçim operatörü ile hangi kromozomlara çaprazlama ve mutasyon operatörü uygulanacağına karar verilmiştir. Son olarak önceden belirlenen oranlara göre seçim operatörü ile seçilen kromozomlara, çaprazlama ve mutasyon operatörleri uygulanmıştır. Bu süreç optimal çözüme, belirlenen iterasyona veya durdurma kriterine ulaşıncaya kadar tekrarlanmıştır.

3.2.4. Parçacık Sürü Optimizasyonu Yönteminin Kullanımı

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

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

3.2.5. Yapay Arı Kolonisi Yönteminin Kullanımı

DÇP'nin çözümü için kullandığımız üçüncü yöntem bal arılarının nektar arayışlarında gösterdikleri işbirliğinden ilham alınarak geliştirilen ABC algoritmasıdır. Amaç, en iyi nektarı (fakülte için en uygun ders programını) bulmaktır.

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

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

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

ABC algoritmasının parametrelerinden biri limit parametresidir. Bu parametre algoritma çalışmaya başlamadan önce belirlenmektedir. Bir çözümün iyileştirilememe değeri belirlenen limit değerinden büyükse kâşif arı aşaması başlar. Bu aşama iyileştirilememe ve limit değerlerini dikkate alarak hafızada tutulan çözümün silinmesine ve yeni çözümün oluşturulmasına karar vermektedir. Bu aşamadan sonra popülasyonun en uygun çözümü hafızaya alınır. İterasyon sayısına veya durdurma kriterine ulaşılmadıysa algoritma işçi arı aşamasına geri dönmektedir.

4. Bulgular ve Tartışma

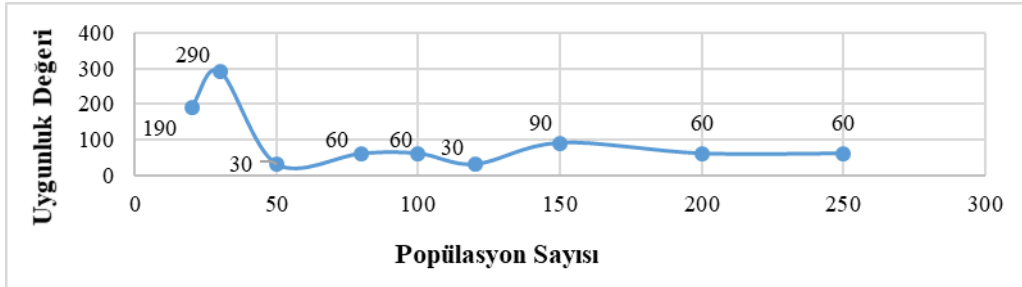
Uygulama için hazırlanan yazılımın çözüm algoritmasında standart GA, PSO ve ABC yöntemleri yazılmıştır. DÇP'nin çözümü için hangi yöntemin daha iyi performans gösterdiğini görmek için bu üç sezgisel yöntem karşılaştırılmıştır.

Tüm algoritmaların düzgün çalışması için ayarlanacak algoritma parametrelerine ihtiyacı vardır. Bu algoritmalar ile optimum çözümün aranması noktasında uygun parametrelerin seçimi oldukça önemlidir. Algoritma parametrelerindeki bir değişiklik, algoritmanın etkinliğini değiştirmektedir. Bu çalışmada tüm parametreler deneysel olarak belirlenmiş ve parametre değerleri üzerinde farklı kombinasyonlar denenmiştir. Üç yöntem içinde çok sayıda deney yapılmış sadece uygun çözüme ulaşan sonuçların bir kısmı aşağıdaki deney tablolarında verilmiştir.

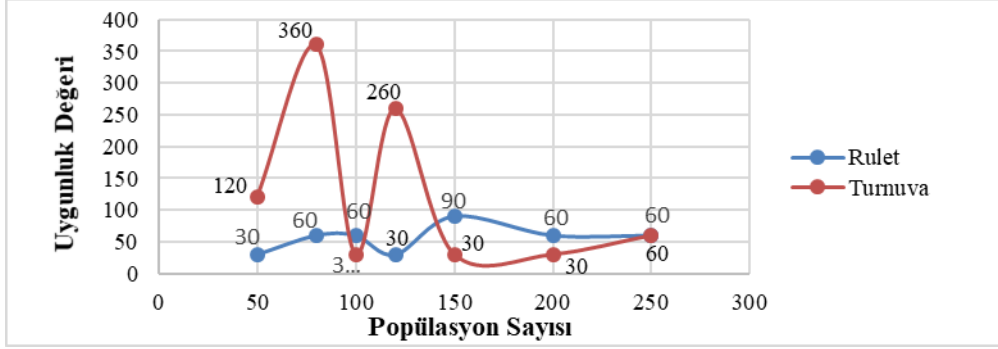
4.1. GA Yönteminin Test Sonuçları

Bu çalışmada popülasyon sayısının 50-100 arasında iyi sonuçlar verdiği 50'nin altına inildikçe uygunluk değerinin kötüleştiği ve bütün zorunlu kısıtların sağlanmadığı görülmüş, 100'ün üstüne çıkıldıkça uygunluk değerinde iyileşme yaşanmadığı görülmüştür. Diğer parametreler sabit tutularak farklı popülasyon sayıları ile yapılan deneylerin uygunluk değeri sonuçları Şekil 2'de gösterilmiştir.

Şekil 2: GA Yönteminde Farklı Popülasyon Sayılarına Göre Uygunluk Değeri Sonuçları



Seçilim operatörü olarak Rulet Tekerleği Yöntemi ile yapılan deneyler Turnuva Seçilim Yöntemine göre daha iyi sonuçlar verdiği gözlenmiştir. Tüm parametreler sabit tutularak seçilim yöntemlerine bakıldığında, genellikle Rulet Tekerleği Yöntemi ile daha hızlı çözüme ulaşıldığı görülmüştür. Şekil 3'de diğer parametreler sabit tutularak farklı popülasyon sayıları ile Rulet Tekerleği ve Turnuva yöntemlerinin uygunluk değeri üzerine etkileri gösterilmiştir. Ayrıca yapılan deneyler Tablo 4'te paylaşılmıştır.

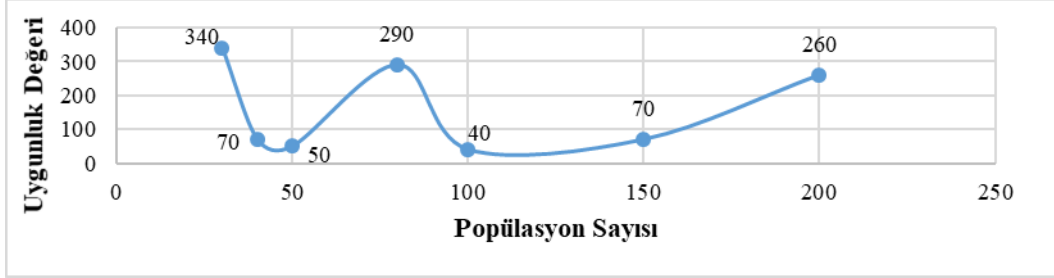
Şekil 3: GA Yöntemi ile Rulet Tekerleği ve Turnuva Yöntemlerinin Karşılaştırma Sonuçları**Tablo 4:** GA Yöntemine Ait Deney Tablosu

Popülasyon Sayısı	İterasyon Sayısı	Seçilim Operatörü	Uygunluk Değeri	Zaman (Dk)
20	1000	Rulet	90	4.9240
50	1000	Rulet	60	7.1777
50	1000	Rulet	30	8.0109
50	1000	Turnuva	60	7.2749
100	1000	Turnuva	30	12.6741
100	1000	Rulet	30	12.9415
150	1000	Turnuva	30	18.9699
150	1000	Rulet	90	20.7375
200	1000	Turnuva	60	25.4325
200	1000	Rulet	60	24.0480
500	1000	Rulet	90	57.4138
500	1000	Turnuva	30	59.9206

4.2. PSO Yönteminin Test Sonuçları

PSO yöntemi oldukça sade bir algoritma yapısına sahip olmasına rağmen zor bir problem olan DÇP'nin çözümünde oldukça etkili olmuştur. PSO tabanlı algoritma ABC yöntemine kıyasla kullanılabilir bir çözüm için daha az zaman harcadığını, GA yöntemine kıyasla ise daha fazla zaman harcadığını göstermiştir. Bununla birlikte bulduğu en uygun ve en hızlı çözüm için harcadığı zamanın diğer iki yöntemden fazla olduğu görülmüştür. Algoritma 40 ve üzeri popülasyon sayısı ve yeterli sayılabilecek 1000 ve 2000 iterasyon sayıları ile uygun çözümler üretmiştir. Fakat 40'ın üstünde bazı popülasyon sayıları ile uygun olmayan sonuçlar bulunmuştur. Uygun olmayan sonuçlar şekillere eklenmiş, çözüm tablosuna eklenmemiştir. PSO yönteminde farklı popülasyon sayıları ile uygunluk değerinin değişimi Şekil 4'de gösterilmiştir. Ayrıca PSO yöntemi kullanılarak yapılan deneyler sonucunda elde edilen bazı uygun çözümler Tablo 5'de verilmiştir.

Şekil 4: PSO Yönteminin Popülasyon Sayısına Göre Uygunluk Değeri Dağılımı



Tablo 5: PSO Yöntemine Ait Deney Tablosu

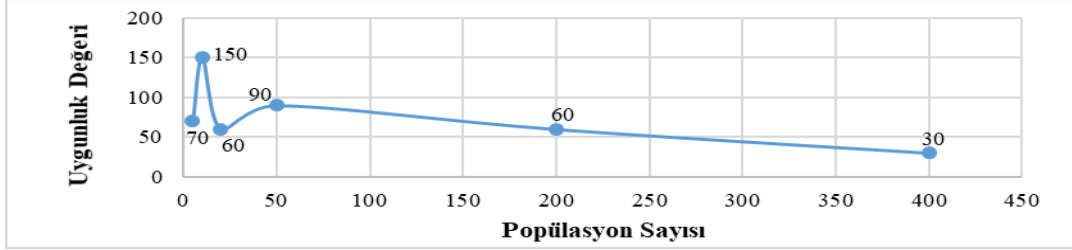
Popülasyon Sayısı	İterasyon Sayısı	Uygunluk Değeri	Zaman (Dk)
40	2000	60	14.7385
40	1000	70	7.6933
50	1000	50	9.2392
50	2000	30	17.9886
100	1000	40	17.7989
100	2000	100	35.4299
150	1000	70	26.5973
150	500	40	13.4041
300	2000	60	107.7465
500	2000	30	189.8092
750	2000	30	277.9001

4.3. ABC Yönteminin Test Sonuçları

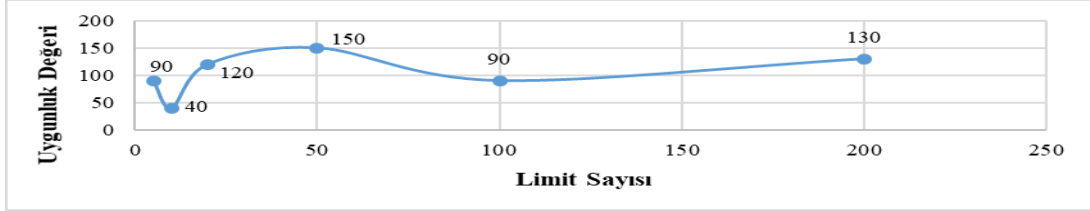
ABC algoritmasının performansını değerlendirmek için farklı popülasyon sayısı, iterasyon sayısı ve limit sayısı ile deneyler yapılmıştır. İterasyon sayısı diğer yöntemlerde olduğu gibi genellikle 1000 veya 2000 olarak sabit tutulmuştur. 2000 iterasyon sayısı ve düşük popülasyon sayısı ile hızlı ve başarılı sonuçlar elde edilmiştir.

Popülasyon sayısı olarak 5 ve 20 arasında ABC algoritması ile uygun ve hızlı çözümler elde edilmiştir. Bu çalışmada limit sayısına 5 ve 500 arasında değerler verilmiştir. Limit sayısının düşük olması problem için uygun çözümün bulunmasını zorlaştırmakta ve problemin çözüm süresini artırmakta, limit sayısının fazla olması durumunda ise algoritmanın daha kaliteli bir çözüm bulma noktasında sorun yaşadığı görülmüştür. Bu çalışmada 10 limit sayısı ile hızlı ve uygun çözümler elde edilmiştir. ABC yönteminde iterasyon sayısı 1000 ve limit değeri 50 olarak sabit tutulmuş, 6 deney ile popülasyon sayısının uygunluk değeri üzerine etkisi gözlemlenmiştir. Bu sonuçlara göre, 0-50 arasındaki popülasyon sayıları ile uygun çözümlere ulaşılmıştır. Şekil 5'de bu deneylerin sonuçları verilmiştir. ABC'nin diğer bir parametresi olan limit sayısı üzerine yapılan deneylerde 10 popülasyon sayısı ve 1000 iterasyon sayısı sabit tutulmuştur. Yapılan deneylerin sonuçları Şekil 6'da verilmiştir.

Şekil 5: ABC Yönteminde Farklı Popülasyon Sayılarına Göre Uygunluk Değeri



Şekil 6: ABC Yönteminde Farklı Limit Sayılarına Göre Uygunluk Değeri



Problemin çözümünde kullanılan diğer iki yöntem ile kıyaslandığında ABC algoritmasının daha başarılı olduğu görülmüştür. Tablo 6'da ABC algoritması ile yapılan bazı deneylerin parametre ve sonuçları paylaşılmıştır.

Tablo 6: ABC Yöntemine Ait Deney Tablosu

Popülasyon Sayısı	İterasyon Sayısı	Limit Sayısı	Uygunluk Değeri	Zaman (Dk)
5	2000	10	30	5.8840
5	2000	100	30	5.7062
5	1000	50	70	2.8968
10	1000	5	90	7.5448
10	1000	10	40	6.7891
10	1000	20	120	6.8564
10	1000	50	150	7.0625
10	1000	100	90	5.8318
10	1000	200	130	5.3061
10	2000	100	30	11.9237
10	3000	10	30	21.4441
20	1000	10	90	11.8228
20	2000	10	30	27.9280
20	1000	50	60	11.3658
50	1000	10	60	28.9996
50	1000	50	90	28.3071
50	1000	100	50	27.7390
100	1000	10	30	62.4398
100	1000	50	30	58.7479
200	1000	50	60	114.863
400	1000	50	30	224.215
500	1000	100	30	299.366

4.4 Yöntemlerin Kıyaslanması

Çalışmamızda önerilen üç yaklaşımın performansları elde edilen en iyi ve en hızlı çözüme göre karşılaştırıldığında üç yöntemde makul sürede oldukça başarılı sonuçlar elde ettiği söylenebilir. Ancak üç yöntem ile elde edilen en iyi ve en hızlı çözümlere göre ABC yöntemi diğer iki yöntemle göre daha hızlı çözüme ulaşmıştır. Söz konusu üç deney Tablo 7’de verilmiştir. Bunun yanı sıra 50 popülasyon sayısı ve 1000 iterasyon sayısı ile algoritmaların en hızlı çözümleri kıyaslanmıştır. Elde edilen bulgulara göre ABC algoritmasının diğer iki yöntemle göre daha yavaş çalıştığı gözlemlenmiştir. GA’nın problemin çözümüne diğer iki yöntemden daha hızlı ulaştığı görülmüştür. Elbette bu çözüm süreleri algoritmaların diğer parametreleri ile doğrudan ilişkilidir. Örneğin; ABC algoritmasının limit değeri veya GA’nın çaprazlama oranı çözüm süresini etkiler. Çalışmamızda her yöntem farklı parametre değerleri ile test edilmiş, uygun çözüme ulaşan parametre değerleri tablolara eklenmiş ve bu üç yöntemin karşılaştırılması da tablolardaki çözümler üzerinden yapılmıştır. Tablo 7’de üç yöntemin kıyaslanan çözümleri verilmiştir.

Tablo 7: GA, PSO ve ABC Yöntemlerini Kıyaslama Tablosu

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

5. Sonuç ve Öneriler

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

Literatürde kullanılan klasik yöntemlerin çözülmesi zor olan çizelgeleme problemlerinin optimizasyonu aşamasında yetersiz kaldığı görülmektedir. Bu çalışmamızda sezgisel yöntemler olan GA, PSO ve ABC’nin farklı parametreleri kullanılarak yapılan deneyler sonucunda zor bir optimizasyon problemi olan DÇP’nin uygun çözümü sunulmuş ve söz konusu yöntemler kıyaslanmıştır. Bu çalışmada yöntemler düzgün bir şekilde kullanarak zor bir problem olan DÇP’nin çözümünde iyi sonuçlar elde edilmiştir. Böylelikle hem bizim fakültemizin hem de farklı fakültelerin kullanabileceği ve bütün bölümlerin ders programını kısa sürede ve tek tabloda hazırlayan algoritmalar geliştirilmiştir.

İleriki çalışmalarda, üç yöntemin de algoritma yapısında kullandığımız algoritmanın performansı açısından oldukça önemli olan genetik operatörlerin iyileştirilmesine odaklanarak, daha kaliteli ve hızlı çözüm veren algoritmaların geliştirilmesi düşünülmektedir.

Bu çalışmanın literatür kısmında bahsedildiği üzere farklı problemlerin çözümünde sezgisel yöntemler ve sezgisel yöntemlerin birlikte kullanımıyla oluşturulan hibrit yaklaşımlar kullanılmıştır. Araştırmacıların gelecekteki çalışmalarında GA, PSO ve ABC yaklaşımları ile birlikte yardımcı yöntemler kullanarak problemlerin çözüm sürelerini kısaltacak algoritmaların elde edilmesini hedefleyebilir. Ayrıca kullanılan bu üç yöntemin dışında ACO, TA, Tavla Benzetimi, Diferansiyel Gelişim gibi farklı yöntemlerle hiyerarşik ve hibrit yapıların ele alınması planlanabilir.

Kaynakça

Akkan, C. ve Gülcü, A. (2018). A bi-criteria hybrid genetic algorithm with robustness objective for the course timetabling problem. *Computers & Operations Research*, 90, 22-32.

Ba, M., Hu, Y., Xu, C. ve Zhong, Y. (2019). A Course Scheduling Algorithm in Secondary Vocational School Based on Genetic Algorithm. In *2019 6th International Conference on Systems and Informatics (ICSAI)* (pp. 1543-1547). IEEE.

Bolaji, A. L. A., Khader, A. T., Al-Betar, M. A. ve Awadallah, M. A. (2011). An improved artificial bee colony for course timetabling. in *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011 Sixth International Conference*, 9-14.

Bourouba, B., ve Ladaci, S. (2016). Comparative performance analysis of ga, pso, ca and abc algorithms for fractional $p_i \lambda d \mu$ controller tuning. In *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, 960-965.

Chaudhari, K.ve Thakkar, A. (2019). Travelling salesman problem: An empirical comparison between aco, pso, abc, fa and ga. In *Emerging Research in Computing, Information, Communication and Applications*, 397-405.

Chen, R. M. ve Shih, H. F. (2013). Solving university course timetabling problems using constriction particle swarm optimization with local search. *Algorithms*, 6(2), 227-244.

Çolak, R. (2015). Sezgisel Algoritmalarla ders programı çizelgeleme problemi çözümü. *Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi, Isparta*.

Dener, M. ve Calp, M. H. (2019). Solving the exam scheduling problems in central exams with genetic algorithms. *Mugla Journal of Science and Technology*, 4(1), 102-115.

Ghaemi, S., Vakılı, M. T. ve Aghagolzadeh, A. (2007). Using a genetic algorithm optimizer tool to solve university timetable scheduling problem. *In Signal Processing and Its Applications, ISSPA 9th International Symposium*, 1-4.

Goldberg, D. E. (1989). *Genetic algorithm in search, optimization & machine learning*, USA: Addison-Wesley Longman Publishing.

Hossain, S. I., Akhand, M. A. H., Shuvo, M. I. R., Siddique, N. ve Adeli, H. (2019). Optimization of university course scheduling problem using particle swarm optimization with selective search. *Expert systems with applications*, 127, 9-24.

Kanović, Ž., Bugarski, V. ve Bačkalić, T. (2014). Ship lock control system optimization using GA, PSO and ABC: a comparative review. *Promet-Traffic & Transportation*, 26(1), 23-31.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization, technical report-tr06. *Erciyes University, Engineering Faculty, Computer Engineering Department*, 200.

Karaboğa, D. (2017). *Yapay zeka optimizasyon algoritmaları*. Ankara: Nobel Yayıncılık.

Kennedy, J. ve Eberhart, R. C. (1995). Particle swarm optimization. *Conference on Neural Networks*, 4, 1942-1948.

Kulkarni, V. R. ve Desai, V. (2016). ABC and PSO: A comparative analysis. *In 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCCIC)*, 1-7.

Küçüksille, E. ve Tokmak, M. (2011). Yapay arı kolonisi algoritması kullanarak otomatik ders çizelgeleme. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 15(3), 203-210.

Le, L. T., Nguyen, H., Dou, J. ve Zhou, J. (2019). A comparative study of PSO-ANN, GA-ANN, ICA-ANN, and ABC-ANN in estimating the heating load of buildings' energy efficiency for smart city planning. *Applied Sciences*, 9(13), 2630.

Muthiah, A., Rajkumar, A. ve Rajkumar, R. (2016). Hybridization of artificial bee colony algorithm with particle swarm optimization algorithm for flexible job shop scheduling. *In 2016 international conference on energy efficient technologies for sustainability (ICEETS)*, 896-903.

Naik, A., Satapathy, S. C. ve Abraham, A. (2020). Modified social group optimization-a meta-heuristic algorithm to solve short-term hydrothermal scheduling. *Applied Soft Computing*, 95, 106524.

Oner, A., Ozcan, S., ve Dengi, D. (2011). Optimization of university course scheduling problem with a hybrid artificial bee colony algorithm. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, 339-346.

Özcan, M. (2016). Atölye tipi çizelgeleme problemlerinde evrimsel algoritmalar ile yapay arı kolonisi algoritmasının bütünleşik bir yaklaşımı. *Doktora Tezi, Sakarya Üniversitesi, Sakarya*.

Özkan, A. (2019). Üniversite ders çizelgeleme probleminin tamsayı doğrusal programlama ve sezgisel yaklaşımlar ile çözümü. *Doktora Tezi, Hacettepe Üniversitesi, Ankara*.

Ramdania, D. R., Irfan, M., Alfarisi, F. ve Nuraiman, D. (2019). Comparison of genetic algorithms and Particle Swarm Optimization (PSO) algorithms in course scheduling. In *Journal of Physics: Conference Series*, 1402 (2), 022079.

Ryder, J. (2006). A Genetic Algorithm for Scheduling.

Sigl, B., Golub M. ve Mornar, V. (2003). Solving timetable-scheduling problem using genetic algorithms, in information technology interfaces, *ITI 2003. Proceedings of the 25th International Conference*, 519-524.

Şahin, Y. (2009). Depo operasyonlarının planlanması için genetik algoritma esaslı bir model. *Yüksek Lisans Tezi, Pamukkale Üniversitesi, Denizli*.

Şamdan, M. ve Yetgin, Z. (2018). Multi-objective cost function optimization using artificial bee colony algorithm with enhanced local search for course scheduling problem. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, 1-6.

Uysal, Ö. (2006). Comparison of Genetic Algorithm And Particle Swarm Optimization Algorithm for bicriteria permutation flowshop scheduling problem, *Doktora Tezi, Marmara Üniversitesi, İstanbul*.

Yazdani, M., Naderi, B. ve Zeinali, E. (2017). Algorithms for university course scheduling problems. *Tehnicki vjesnik/Technical Gazette*, 24.

Zhang, R. ve Wu, C. (2011). An artificial bee colony algorithm for the job shop scheduling problem with random processing times. *Entropy*, 13(9), 1708-172.