

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**MÜFREDAT TABANLI ÜNİVERSİTE DERS ÇİZELGELEME PROBLEMİ İÇİN
BİR SÜRÜ ZEKÂSI YAKLAŞIMI**

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Saeid AGAHİAN

HAZİRAN 2012

TRABZON

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**MÜFREDAT TABANLI ÜNİVERSİTE DERS ÇİZELGELEME PROBLEMİ İÇİN
BİR SÜRÜ ZEKASI YAKLAŞIMI**

Bilgisayar Müh. Saeid AGAHIAN

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
"BİLGİSAYAR YÜKSEK MÜHENDİSİ"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih : 25.05.2012
Tezin Savunma Tarihi : 15.06. 2012**

Tez Danışmanı : Yrd. Doç. Dr. Hüseyin PEHLİVAN

Trabzon 2012

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalında

Saeid AGAHIAN tarafından hazırlanan

**MÜFREDAT TABANLI ÜNİVERSİTE DERS ÇİZELGELEME PROBLEMİ İÇİN
BİR SÜRÜ ZEKASI YAKLAŞIMI**

**başlıklı bu çalışma, Enstitü Yönetim Kurulunun 29 / 05 / 2012 gün ve 1458/3 sayılı
kararıyla oluşturulan jüri tarafından yapılan sınavda**

YÜKSEK LİSANS TEZİ

olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Yrd. Doç. Dr. Hüseyin PEHLİVAN

Üye : Yrd. Doç. Dr, Mustafa ULUTAŞ

Üye : Yrd. Doç. Dr. Halil İbrahim OKUMUŞ

Prof. Dr. Sadettin KORKMAZ

Enstitü Müdürü

ÖNSÖZ

Bu çalışma Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tezi olarak hazırlanmıştır.

“MÜFREDAT TABANLI ÜNİVERSİTE DERS ÇİZELGELEME PROBLEMİ İÇİN BİR SÜRÜ ZEKASI YAKLAŞIMI” konulu bu tez çalışmasını, bana öneren ve çalışmanın her aşamasında gerek bilgi ve tecrübelerini gerekse maddi ve manevi desteğini benden esirgemeyen, önerileriyle ufkumu açan, görevlerindeki yoğunluğuna rağmen değerli zamanını bana ayıran danışmanım ve saygıdeğer hocam sayın Yrd. Doç. Dr. Hüseyin PEHLİVAN'a en içten teşekkürlerimi sunarım.

Çalışma süresinde değerli görüş ve bilgilerini benimle paylaşan ve çalışmayı inceleyen değerli hocalarım Yrd. Doç. Dr. Orhan KESEMEN ve Mohammad Reza ALAVİ MİLANI'ye çok teşekkür ederim.

Bilgi ve tecrübelerini hiçbir zaman esirgemeyen çok kıymetli arkadaşlarım, Bahram SAMADİ VAİGHAN, Rahim DEHKHARGHANI ve Türkçe yazım düzenlemesinde yardım eden Merve KESİM, Müşra DEĞİRMEN, Hatice BAŞOL, ve destekleyen diğer arkadaşlarıma da şükranlarımı sunarım.

Anadilim Türkçe'de ilk olarak öğretim almak için çok mutluyum ve bu ortamı sağlayan herkese teşekkür ederim.

Son olarak hayatım boyunca beni sabır ve şefkatle destekleyen anne ve babama müteşekkir olduğumu belirtir, çalışmanın faydalı olmasını dilerim.

Saeid AGAHİAN
Trabzon 2012

TEZ BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “MÜFREDAT TABANLI ÜNİVERSİTE DERS ÇİZELGELEME PROBLEMİ İÇİN BİR SÜRÜ ZEKÂSI YAKLAŞIMI” başlıklı bu çalışmayı baştan sona kadar danışmanım Yrd. Doç. Dr. Hüseyin PEHLİVAN‘ın sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuarlarda yaptığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 25/05/2012

Saeid AGAHİAN

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	III
TEZ BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VIII
SUMMARY	IX
ŞEKİLLER DİZİNİ.....	X
TABLolar DİZİNİ.....	XI
SEMBOLLER DİZİNİ	XII
1. GENEL BİLGİLER.....	1
1.1. Optimizasyon Nedir?.....	1
1.1.1. Optimizasyonun Matematiksel Tanımı	3
1.1.2. Optimizasyon Problemlerinin Sınıflandırılması.....	4
1.2. Zaman Çizelgeleme	6
1.2.1. Zaman Çizelgelemenin Formal Tanıtımı ve Türleri.....	6
1.2.2. Eğitimsel Zaman Çizelgeleme.....	7
1.2.2.1. Okul Çizelgeleme (School Timetabling).....	8
1.2.2.2. Üniversite Ders Çizelgeleme (University Course Timetabling).....	8
1.2.2.3. Sınav çizelgeleme (Exam Timetabling).....	8
1.3. Üniversite Ders Çizelgeleme	9
1.3.1. Problemin Matematiksel Açıklaması.....	10
1.3.2. Örnek (Benchmark) Problemlerin Üniversite Ders Çizelgelemesi	12
1.3.2.1. Uluslararası Zaman Çizelgeleme (ITC-2002)	14
1.3.2.2. Socha'nın Örnek Problemi	16
1.3.2.3. Müfredat Tabanlı Üniversite Ders Çizelgelemesi	19

1.4.	Literatür Tarama	22
1.4.1.	Sıralı Yöntemler.....	23
1.4.2.	Kümeleme Yöntemleri.....	26
1.4.3.	Kısıt Tabanlı Yöntemler	26
1.4.4.	Meta Sezgisel Yöntemler.....	27
1.4.4.1.	Lewis Meta Sezgisel Yöntemlerin Sınıflandırılması	27
1.4.4.1.1.	Tek Aşamalı Optimizasyon Algoritmalar.....	27
1.4.4.1.2.	İki Aşamalı Optimizasyon Algoritmalar	28
1.4.4.1.3.	Gevşemeye İzin Veren Algoritmalar	28
1.4.4.2.	Obit Meta Sezgisel Yöntemlerin Sınıflandırılması	29
1.4.4.2.1.	Popülasyon Tabanlı Meta Sezgisel Yaklaşımlar	29
1.4.4.2.2.	Tek Bir Çözüm Tabanlı Meta Sezgisel Yaklaşımlar	31
1.4.5.	Melez Evrimsel ve Meta Sezgisel Algoritmalar	32
1.4.6.	Çok Ölçütlü Yaklaşım	32
1.4.7.	Durum-Tabanlı Düşünme Yaklaşımları	32
1.4.8.	Üst sezgisel ve Uyarlamalı Yaklaşımlar.....	32
1.5.	Sürü Zekâsı ve Temelleri.....	33
1.5.1.	Kendi Kendine Organize Olabilme	34
1.5.1.1.	Pozitif Geri Besleme.....	35
1.5.1.2.	Negatif Geri Besleme	35
1.5.1.3.	Dalgalanmalar.....	35
1.5.1.4.	Çoklu Etkileşimler.....	35
1.5.2.	İş Bölümü	36
1.6.	Doğada Arılar	36
1.7.	Gerçek Arıların Yiyecek Arama Davranışları	37
1.7.1.	Yiyecek Kaynakları	37
1.7.2.	Görevli İşçi Arılar.....	37
1.7.3.	Görevsiz İşçi Arılar.....	37

1.8.	Yapay Arı Koloni Algoritması	40
1.8.1.	Başlangıç Yiyecek Kaynağı Bölgelerinin Üretilmesi.....	42
1.8.2.	Görevli Arıların Yiyecek Kaynağı Bölgelerine Gönderilmesi	42
1.8.3.	Görevli Arıların Seleksiyonda Kullanacakları Olasılık Değerlerinin Hesaplanması (Dans Benzetme).....	44
1.8.4.	Gözcü Arıların Yiyecek Kaynağı Bölgelerini Seçmeleri	44
1.8.5.	Kaynağı Bırakma Kriteri: Limit ve Kaşif Arı Üretimi	45
1.8.6.	ABC'nin Temel Özellikleri	47
2.	YAPILAN ÇALIŞMALAR.....	48
2.1.	Giriş	48
2.2.	Problem Tanıma.....	50
2.3.	Yöntem	56
2.3.1.	Komşuluk Arama Mekanizmaları	59
2.3.2.	Kullanılan Veri Yapıları	62
2.3.3.	İlk Aşama.....	66
2.3.3.1.	İlk Aşama Literatür Analizleri.....	66
2.3.3.2.	İlk Aşama İçin Önerilen Yöntem.....	70
2.3.4.	İkinci Aşama	76
2.3.4.1.	İkinci Aşama Literatür Analizleri.....	76
2.3.4.2.	İkinci Aşama İçin Önerilen Yöntem.....	77
3.	BULGULAR VE SONUÇLAR.....	82
3.1.	Önerilen Yaklaşımının comp15.ctt Problemin Veriler Üzerinde Denemesi	82
3.2.	ITC-2007 (CB-CTT) Örnek Problem Üzerinde Önerilen Yaklaşımının Sonuçları.....	90
4.	ÖNERİLER.....	93
5.	KAYNAKLAR	94

ÖZGEÇMİŞ

Yüksek Lisans

ÖZET

MÜFREDAT TABANLI ÜNİVERSİTE DERS ÇİZELGELEME PROBLEMİ İÇİN BİR
SÜRÜ ZEKÂSI YAKLAŞIMI

Saeid AGAHIAN

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Yrd. Doç. Dr. Hüseyin PEHLİVAN
2012, 102 Sayfa

Üniversite ders çizelgeleme problemi popüler zaman çizelgeleme problemlerinden biri olduğundan çok sayıda araştırmacının ilgisini çekmiştir. NP-Zor problemler grubunda yer alan bu problemin deterministik yöntemlerle çözümü verimli değildir ve bu nedenle sezgisel ve meta-sezgisel yöntemler önerilmektedir. Sürekli uzaydaki optimizasyon problemlerinin çözümü için önerilen meta-sezgisel yöntemlerinden birisi olan ABC algoritması bal arılarının yiyecek aramada kullandığı sürü zekasından ilham alır. Bu algoritma sürekli uzayda diğer meta sezgisel yöntemlerden daha iyi sonuçlar üretmektedir. Ayrık uzay problemlerinin, örneğin üniversite ders çizelgeleme probleminin çözümü için ABC'nin uyarlanması işleminde komşuluk kavramının tanımlanmasına ve kullanılmasına ihtiyaç duyulur.

Bu tezde ABC algoritması üniversite ders çizelgeleme problemi türlerinden birisi olan müfredat tabanlı üniversite ders çizelgeleme probleminin (CB-CTT) çözümü için kullanılmıştır. ITC-2007 yarışmasının üçüncü bölümünde yer alan bu problemin çözümünde, temel ABC algoritmasının komşu üretimine değişiklikler yapılmış ve yerel arama yöntemi eklenmiştir. Problem için geliştirilen standart veri setleri üzerinde farklı araştırmacıların farklı yöntemlerle elde ettikleri sonuçları karşılaştırabilmesi sağlandığından dolayı, ITC-2007 probleminin 21 örnek veri seti üzerinde elde ettiğimiz sonuçlar, yarışmada dereceye giren en iyi 5 sonuç ile kıyaslanabilir.

Anahtar Kelimeler: Meta sezgisel yöntem, Yapay arı kolonisi, üniversite ders çizelgeleme

Master Thesis

SUMMARY

A Swarm Intelligence Approach to Curriculum-based Course Timetabling Problem

Saeid AGAHIAN

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Assist. Prof. Hüseyin PEHLIVAN
2012, 102 Pages,

The University Course Timetabling Problem is one of the most famous scheduling problems, which has attracted many researchers to solve the problem. This problem is located in NP-Hard problems group and so deterministic methods are not efficient to solve it and heuristic and meta-heuristic methods are suggested to do so. One of the meta-heuristic methods suggested to solve optimization problems in continuous environments that is inspired by the swarm intelligence of honey bees in search for the food, is called ABC algorithm. This algorithm could achieve good results compared to other meta-heuristic methods in a continuous environment. The task of adapting ABC to solve problem in discrete environments, e.g. University Course Timetabling, needs defining the neighborhood concept and applying this concept to find the better solutions. In this thesis, ABC has been used to solve one of popular types of University Course Time-Tabling problems which is Curriculum Based University Course (CB-CTT) Time-Tabling problem. It was the third type of Time-Tabling problems in ITC-2007 competition. The adaptation of ABC to solve this problem gets the advantage of some improvements in neighbor generation and using local search technique, compared to the original ABC algorithm.

Having a benchmark case study for this problem helps researchers to work on a standard dataset and makes it easy to compare different results obtained by different researchers with different approaches. Our achieved results on 21 datasets of ITC-2007 is comparable to five winners of this competition.

Key Words: Meta-heuristic method, Artificial bee colony, University course timetabling

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. Gerçek dünya problemlerin optimizasyon yaklaşımlarla çözüm modeli [1].....	3
Şekil 2. Tek boyutlu amaç fonksiyon [3].....	4
Şekil 3. Doğrusal ve doğrusal olmayan programlama problemi [2], (a) doğrusal amaç fonksiyonu; (b) doğrusal olmayan amaç fonksiyonu.....	5
Şekil 4. Rulet Tekerlik üzerinde dizilerin seçme olasılığı.	30
Şekil 5. Sürü zekası.....	34
Şekil 6. Yiyecek arama çevrimi [100].	39
Şekil 7. ABC algoritmanın temel adımları [100].....	41
Şekil 8. ABC Algoritmasının akış diyagramı [3, 100].....	46
Şekil 9. Önerilen yöntemin akış diyagramı.....	58
Şekil 10. Kemp chain tanıtımı[114].....	61
Şekil 11. Çözümü matrisi.....	62
Şekil 12. UNAVAILABILITY_CONSTRAINTS matrisi	64
Şekil 13. Room_of_course matrisi.....	65
Şekil 14. Proof_time matrisi	65
Şekil 15. Group_time matrisi.....	66
Şekil 16. Graf renklendirmekle mümkün ders çizelgeleme oluşturma [26].	67
Şekil 17. Mümkün çözümler üreten yöntemin adımları [63].....	68
Şekil 18. İlk aşama için SN1 tane başlangıç çözüm üretimi	72
Şekil 19. İlk aşama için ABC algoritmasının kullanımı	75
Şekil 20. İkinci aşama için ABC'nin uygulaması	81
Şekil 21. En kaliteli başlangıç çözümün değerlendirilmesi	84
Şekil 22. İkinci aşamanın ABC'sinin başlangıç çözümlerin en kalitelisinin değerlendirmesi.....	85
Şekil 23. İkinci aşamanın ABC'sinin 100 saniyede çözümlerin en kalitelisinin değerlendirilmesi.....	87
Şekil 24. Comp15.ctt üzerinde önerilen yaklaşımın sonuçları	88
Şekil 25. İkinci aşamanın ABC'sinin 468 saniyede çözümlerin en kalitelisinin değerlendirilmesi.....	89

TABLolar DİZİNİ

	<u>Sayfa No</u>
Tablo 1. Örnek problem verilerinin özellikleri.....	15
Tablo 2. Üretilen örnek verilerin özellikleri.....	17
Tablo 3. Örnek verilerin bilgileri.....	21
Tablo 4. ToyExample örneğın ders özellikler	24
Tablo 5. Çakışma matrisi.....	24
Tablo 6. Derslere verilen dereceler.....	24
Tablo 7. Bireylerin bilgileri	30
Tablo 8. Örnek problem için verilen dosya	51
Tablo 9. Örnek verilerin dosya da bilgilerin biçimi	52
Tablo 10. ToyExample.ctt problem için bir aday çözüm dosyası (Output.out)	55
Tablo 11. Aday çözümün verilen uygulamayla değerlendirmesi	56
Tablo 12. Birinci aşamanın en kaliteli başlangıç çözümü (Z.P zaman periyod).....	83
Tablo 13. İkinci aşamanın başlangıç mümkün çözümlerin en kalitelisi.....	85
Tablo 14. İkinci aşamanın 100. saniye en kaliteli çözüm.....	87
Tablo 15. İkinci aşamanın sonunda çözümlerin en kaliteli çözümü.....	89
Tablo 16. İlk aşamanın sonuçları (saniye biriminden)	91
Tablo 17. ITC-2007 örnek verilerin üzerinde önerilen yaklaşımların sonuçları	92

SEMBOLLER DİZİNİ

CB-CTT	Curriculum-based Course Timetabling
ABC	Artificial Bee Colony
Σ	Toplama
$f(x)$	x Üzerine amaç fonksiyonu
ACO	Ant Colony Optimizasyonu
PATAT	Practice and Theory on Automated Timetabling
NP	Nondeterministic Polynomial time
LDF	Largest Degree First
Z.P	Zaman Periyodu
CSP	Constraint Satisfaction Problems
f_i	i çözümün amaç fonksiyon değeri
$fitness_i$	i çözümün uygunluk (kar) fonksiyon değeri
p_i	i çözümün olasılık değeri
abs	Mutlak Değer
$fitnes_i$	i amaç foksiyon değeri üzerinde kalite değeri
SN	ABC'nin Yiyecek kaynak sayısı
ϵ	Doğruluk foksiyonu
l_i	i dersin oturumlar sayısı
$wd(c)$	C dersin oturumlarının hafta boyu atanan günü
C	Dersler sayısı (courses)
L	Derslerin oturumların sayısı (total Lecture)
R	Derslikler sayısı (Room)
PPD	Gün içerisinde zaman dilim sayısı (periods per day)
Cu	Müfredat sayısı (curricula)
MML	Her gün için her müfredata düşen en az ve en çok oturum sayısını,(Min and Max Lectures per day per curriculum)
CL	Her müfredat için günlük ortalama oturum sayısını, (average number of lectures per curriculum per day):
RO	Her zaman dilimi için uygun olan ortalama (Average Room Occupation)

1. GENEL BİLGİLER

1.1. Optimizasyon Nedir?

Gerçek anlamda optimizasyon, kıt kaynakların en iyi biçimde kullanılmasını sağlayan bir matematik bilimdir. Optimizasyon tekniklerinin en çok uygulandığı alanlar, endüstriyel planlama, kaynak ayırma, zaman planlama, karar verme, vb. gibi sayılabilir. Örnek olarak, küresel bir petrol rafineri işletmesi, ham petrolün nereden alınması, işlenmesi için nereye gönderileceği, hangi tür ürüne dönüştürüleceği, elde edilen ürünün nerede satılacağı ve fiyatının ne olacağı gibi sorulara cevap aramaktadır. Bu sorunlara çözmek için maksimum kar optimizasyon modeli kullanılabilir. Bir havayolu işletmesi, belli aralıklarla uçaklarının bakımlarını yaptırmaktadır. Bu bakımlar için görevlendirilen çalışanları daha az çalıştırmak için uçaklarının uçuş zamanlarının nasıl ayarlanacağı bilmek istenmektedir. Burada önemli kısıtlamalardan birisi her uçağın bakım yapıldıktan sonra maksimum bir uçuş süresinin bulunmasıdır. Bu problemde minimum maliyetli bir optimizasyon modeli kullanılır [1].

İnsanoğlu dünyada var olduğu süre boyunca birçok alanda mükemmelliğe ulaşmak için çaba göstermiştir. En az çaba ile en yüksek başarıya ulaşmak isteği insanoğlunun gelişme yolunda gösterdiği çabalardan birisidir. Ekonomik alanda ise işletmeler minimum maliyet ile maksimum kar elde etmeye çalışmaktadırlar. Bundan dolayı optimizasyon bilimi, en eski bilimlerden biri sayılmasına rağmen günümüze kadar araştırma güncelliğini korumaya devam etmektedir [2].

Günümüzde kullanılan büyük ölçekteki optimizasyon tekniklerinin çoğu II. Dünya savaşı sırasında geliştirilen yöntemlere dayanmaktadır. Savaş sırasında milyonlarca askere ve makineye sahip orduların kıt kaynakların kullanımı ile lojistik ihtiyacının karşılanmasındaki zorluklar optimizasyon tekniklerinin gelişmesini sağlamıştır. Büyük ölçekteki optimizasyon yöntemlerinin temeli savaş sırasında geliştirilen simplex yöntemine dayanmaktadır [1].

Belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilir. Optimizasyon işleminde ilk adım olarak karar parametreleri veya karar değişkenleri ya da tasarım parametreleri olarak da adlandırılan parametreler kümesinin tanımlaması gerekir.

Sonra, bu parametrelere baęlı olarak en kk yapılacak (minimize edilecek) bir maliyet fonksiyonu veya en byk yapılacak (maksimize edilecek) bir kar fonksiyonu ve problemle ilgili sınırlama fonksiyonu (constraints) tanımlanmalıdır. Maliyet fonksiyonu, daha iyi zm temsil eden parametre deęerlerinin kullanılması durumunda daha dk bir numerik deęer retirken, kar fonksiyonu ise daha yksek bir nmerik deęer retmektedir. Sınırlama, parametrelerine baęlı olarak ifade edilmektedir. Bazı sınırlamalar eitsizlikler, bazıları ise eitlikler biiminde olabilir [3].

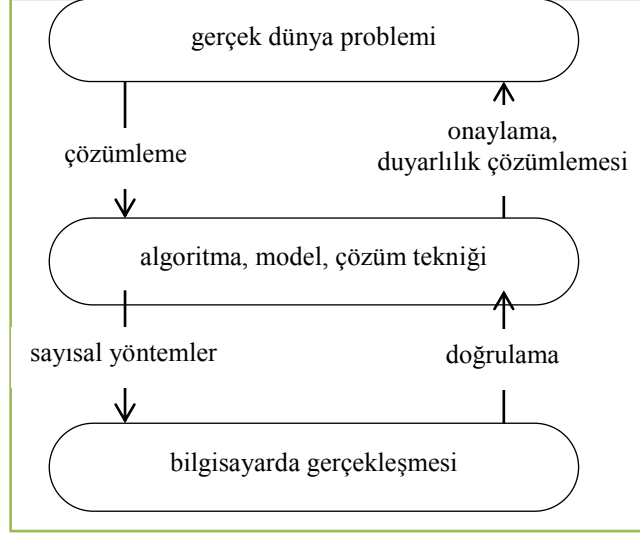
Maliyet fonksiyonunun tanımlanması, problemin doęasına ve yapısına baęlıdır. rnek olarak taımacılık optimizasyon probleminde, yolculuk sresi veya yolculuk masrafı (maliyeti) en yaygın amalarındandır. Bylelikle uygun maliyet (ama) fonksiyonunun seimi optimizasyon ilemlerinin en nemli adımlarından birisidir.

Bazen optimizasyon ileminde aynı zamanda birden fazla amacın saęlanması hedeflenebilmektedir. Yani birden fazla maliyet fonksiyonu bulunmaktadır, bu tr optimizasyon problemlerin ok amalı optimizasyon problemleri olarak adlandırılır. Bu tr problemlerde en basit yaklaım bir ama fonksiyonu retmektir. Bu ama fonksiyonu dięer ama fonksiyonlarının doęrusal birleimidir. Her biri farklı aęırlık deęerleriyle ama fonksiyonuna katılırlar.

Her optimizasyon problemi birkaç baęımsız deęikenden oluabilir. Bunlara tasarım deęikenleri denir ve bir N uzunluęunda, X listesiyle gsterilir.

Optimizasyon ileminin amacı, ama fonksiyonun maksimum veya minimum yapan tasarım deęikenlerinin belirlenmesidir.

Gerek dnya problemlerine uygulanan optimizasyon yntemleri, Őekil 1'de Őematik olarak verilmektedir. Bu Őemada grldęi gibi optimizasyon ynteminin uygulanmasında matematiksel tanım nemli bir yer tutmaktadır.



Şekil 1. Gerçek dünya problemlerin optimizasyon yaklaşımlarla çözüm modeli [1].

1.1.1. Optimizasyonun Matematiksel Tanımı

Bir fonksiyonun minimum veya maksimum değerinin bulunması, gerçel veya tamsayı parametrelerini tanımlı bir aralıkta seçip fonksiyona yerleştirerek sistematik olarak bir problemi inceleme yada çözüme işlemi optimizasyonun matematiksel temelini oluşturur.

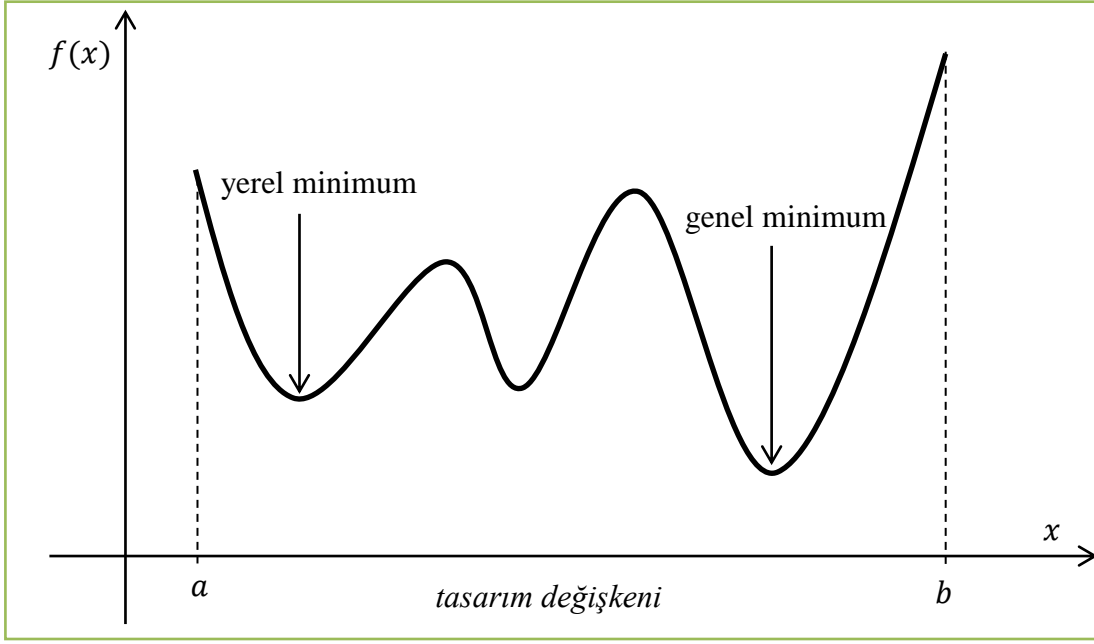
Genel (Global) optimizasyonun amacı mümkün olan en iyi x parametresini, X kümesinin içerisinde sınırlı sayıda kısıtlar kümesini ($F = \{f_1, f_2, \dots, f_n\}$) sağlayacak şekilde bulunmasıdır. Bu kısıtlar matematiksel fonksiyonla gösterilir.

Problem için sınırlamalar sağlayan mümkün tüm çözümlerin oluşturduğu bölge araştırma yapılabilecek uygun (feasible) çözüm bölgesi olarak adlandırılır.

$f(x)$ maliyet fonksiyonu, $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ çözümü için denklem (1) ile verilen şartı uygun bölge (S) içindeki tüm muhtemel çözümler (x) için sağlıyor ise $f(x)$ fonksiyonu x^* 'de genel minimuma sahiptir (x n boyutlu bir vektör).

$$f(x^*) \leq f(x) \quad (1)$$

Eğer denklem (1) uygun bölge içerisinde, x^* 'in bir komşuluğun (N) içerisinde tüm muhtemel çözümler için sağlanıyorsa $f(x)$ fonksiyonu x^* 'da yerel minimuma sahiptir [3].

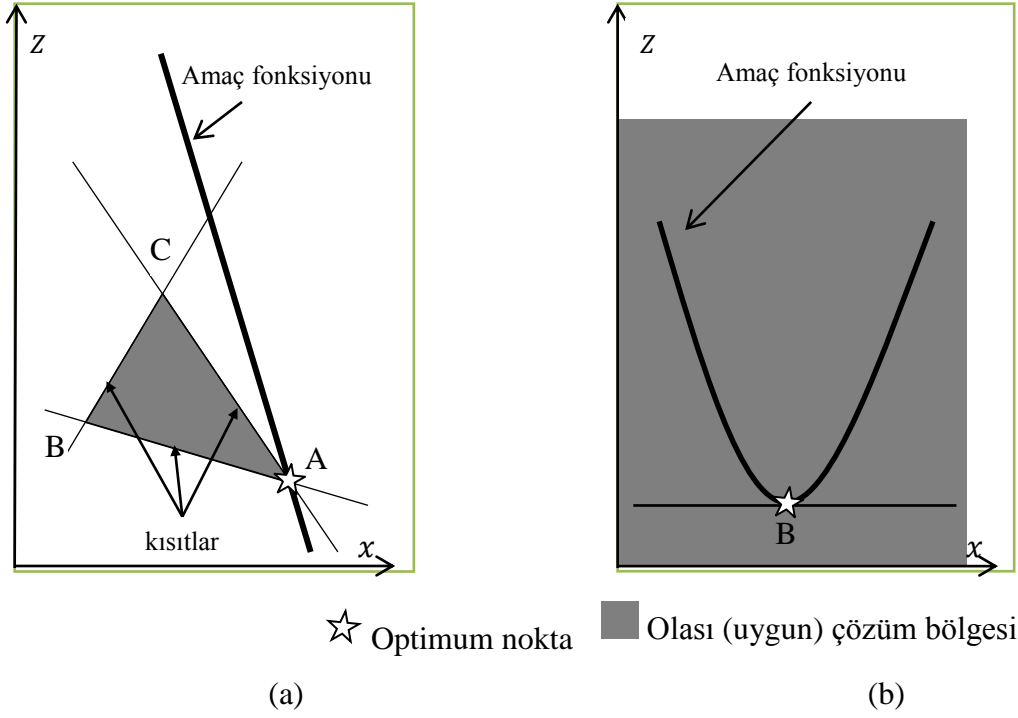


Şekil 2. Tek boyutlu amaç fonksiyon [3].

1.1.2. Optimizasyon Problemlerinin Sınıflandırılması

Optimizasyon problemlerinde $f(x)$ amaç fonksiyonundaki x değerleri için bir sınırlama varsa sınırlamalı optimizasyon problemi yoksa sınırlamasız optimizasyon problemi olarak adlandırılır. Eğer sınırlamalı fonksiyonlar doğrusal ise doğrusal programlama problemi değilse doğrusal olmayan programlama denir.

Şekil 2’de tek boyutlu doğrusal, kısıtlı amaç fonksiyonu ve Şekil 2(b)’de ise kısıtsız ve doğrusal olmayan tek boyutlu amaç fonksiyonu gösterilmektedir. Burada optimum nokta, kısıtlamayı sağlayan bölgede amaç fonksiyonunu minimum yapan noktadır. Bu problemin sayısal bir optimizasyon problemi olduğu açıkça görülmektedir.



Şekil 3. Doğrusal ve doğrusal olmayan programlama problemi [2], (a) doğrusal amaç fonksiyonu; (b) doğrusal olmayan amaç fonksiyonu.

Diğer bir sınıflandırma, tasarım değişkenlerin aldığı değerlere göre sürekli (continuous) veya kombinyasyonel (combinatorial) optimizasyon olarak yapılmaktadır.

Sürekli değişkenli problemlerde genellikle gerçel sayıların bir kümesi üzerinde tanımlı bir fonksiyon araştırılır. Yani tasarım değişkenleri değerlerini sürekli bir uzaydan alırlar. Kombinyasyonel problemlerde ise sonlu bir kümeden bir nesne ya da sayılabilir sonsuz bir küme (tipik olarak bir tamsayılar kümesi, permutasyonlar kümesi yada graflar) üzerinde tanımlanmış bir fonksiyon araştırılır. Bu problemde sadece değerleri belirlemek değil aynı zamanda nesnelerin optimal düzenlemesi, gruplanması veya sıralanması söz konusu olur. Bu problemler oldukça farklı özelliklere sahip olup çözümleri içinde kullanılan yöntemler farklıdır [4].

Mühendislikte birçok optimizasyon problemi oldukça zor ve karmaşık bir yapıya sahiptir. Bu problemlerin çözümü matematiksel programlama veya benzer optimizasyon yöntemleriyle yapılamamaktadır [5].

Günümüzde karmaşık optimizasyon problemlerinin çoğu NP-ZOR (NP-HARD) problemler türünde olup, problemin boyutu büyüdükçe deterministik bir yöntemle çözümü bulmak için gereken hesaplama zamanı ve bellek kullanımı yönünden polinomial olmayan bir şekilde büyümektedir. Bu nedenle bilgisayar destekli yaklaşık çözüm yöntemleri

kullanılmaya başlanmıştır. Örnek olarak sezgisel, sezgisel üstü, meta sezgisel yöntemler olarak sıralanabilir.

1.2. Zaman Çizelgeleme

1.2.1. Zaman Çizelgelemenin Formal Tanıtımı ve Türleri

Günlük yaşamın birçok alanında zaman çizelgesi bulunmaktadır. Örneğin iş yönetimi, eğitim öğretim, taşımacılık veya bir tur organizasyonundan da zaman çizelgeleme problemi çıkmaktadır. Düzenli ve modern bir toplumun ilerlemesi için işlerini çözerken zaman çizelgeleme yapması kaçınılmaz bir zorunluluktur.

Halen günlük yaşamın birçok alanında, özellikle kaynakların (kişi, yer ve zaman) kısıtlı olması durumlarında işe yarayan ve pratik zaman çizelgeleme üretimi, bu işle uğraşanların büyük bir emek ve zaman harcamasına neden olmaktadır.

Zaman içerisinde kaynaklar, istekler ve kısıtların değişimi zaman çizelgeleme problemi hala güncel ve sürekli bir iş olmasını sağlamaktadır.

Zaman çizelgeleme problemi ilk olarak Wren tarafından [6], “Kısıtları göz önünde bulundurarak mümkün olduğunca istenileni karşılayabilecek bir şekilde, verilen kaynakların belirli zaman aralıklarında paylaşılmasıdır” olarak biçimsel yönden tanımlanmıştır.

Wren [6] her probleme ait kısıtları genelde iki kategoriye ayırır. Bunlar katı kısıtlar (hard constraints) ve esnek kısıtlar (soft constraints) olarak verilmiştir. Katı kısıtlar için şartları ne olursa olsun çözümü bulma sürecinde değişmeden yerine getirilmesi gerekmektedir. Esnek kısıtların karşılanması veya karşılanmaması çözümün doğru veya yanlış olmasında etkisi yoktur ama çözümün kalitesini etkileyebilir. Yani esnek kısıtlar ne kadar fazla göze alınırsa o kadar çözümün kalitesi artmış olur ve tersi durumunda ise çözümün kalitesi düşmektedir.

Katı ve esnek kısıtların artması ile zaman çizelgeleme problemi büyür ve bir NP problemine dönüşür. Bu çeşit NP problemlerini çözmek için kullanılan en yaygın ve eski yöntemlerden birisi sezgisel yöntemlerdir. Bir sezgisel yöntem kullanımına örnek olarak zaman çizelgeleme problemini bir graf renklendirme problemine dönüştürme ve sezgisel yöntemlerle bu grafi renklendirme verilebilir [7].

Zaman çizelgelemede en temel örnek, önceki bölümlerde söz edildiği gibi gündelik hayatta birçok farklı türde zaman çizelgesi problemi gözlenebilmektedir. Örneğin tren zaman çizelgesi oluştururken tren, günlük yolculuğun nerede başlayıp nerede biteceği, tren sürücülerinin kullanabildiği zamanlar ve sürücüler her nöbete kaç saat çalışacakları göz önüne alınmaktadır [8].

Diğer örnek otobüslerin zaman çizelgelemesidir. Bu problemin tren çizelgesiyle birçok ortak yönü olmasına rağmen farklı yönleri de bulunmaktadır. Spor müsabakaları zaman çizelgelemede başka bir örnek olabilir. Çok çeşitli zaman çizelgeleme problemleri arasından pratik ve en yaygın olanlardan birisi eğitimsel zaman çizelgelemesidir (Educational Timetabling).

Zaman çizelgeleme problemi araştırmacıların ilgisini çeken yaygın konulardan birisidir ve dünyada bu konuda konferanslar düzenlenmektedir. Bu konferanslardan birisi düzenli bir şekilde yapılan PATAT (Practice and Theory on Automated Timetabling) [9] adlı bir konferanstır. Bu konferans zaman çizelgeleme yönteminin gelişmesine oldukça yararlı olmuştur ve zaman çizelgeleme çalışma gruplarının buluşmasına katkı sağlamıştır. Çok sayıda makale bu konferans aracılığıyla yayınlanmış ve halen de yayınlanmaya devam etmektedir. Euro zaman çizelgeleme çalışma grubunun kurulması da zaman çizelgeleme probleminin önemini göstermektedir [10].

1.2.2. Eğitimsel Zaman Çizelgeleme

Bir eğitim-öğretim kurumunun, mevcut derslikleri ve bu dersliklerin kapasitelerini, tüm öğretmen ve öğrencilerin istek ve ihtiyaçlarını da aynı anda göz önünde bulundurarak sağlayabilecek bir ders çizelgelemesinin oluşturulması, bu iş ile uğraşan personel için oldukça zor ve zaman alıcı bir iştir [11].

Her eğitim kurumu kendine özel kısıtlara ve amaçlara sahip olduğu için bütün durumlar için kullanılacak genel bir model oluşturulabilmesi mümkün değildir [11]. Bu nedenle çok yaygın ve pratik bir araştırma alanına dönüşen zaman çizelgeleme araştırmacıların ilgisini (yararına ve ihtiyacına göre) çekmiştir.

Schaerf 1999 'da yayınladığı makalede [12], bu problemi üç ana başlıkta toplar: okul zaman çizelgelemesi (School Timetabling); ders çizelgelemesi (Course Timetabling) ve sınav çizelgelemesidir (Exam timetabling). Bunların birbiriyle ortak yönlerinin olmasına

rağmen çok önemli farkları bulunmaktadır. Bu farklılıklar aşağıdaki bölümlerde ayrıntılı bir biçimde incelenecektir.

1.2.2.1. Okul Çizelgeleme (School Timetabling)

Bir okulun bütün sınıflarının haftalık ders programı yapılmak isteniyor. Ancak dikkat edilmesi gereken bir noktada, bir öğretmenin aynı anda iki farklı sınıfa atanmamasıdır. Aynı şekilde iki öğretmen de aynı anda aynı sınıfa atanmamalıdır. Normalde okul ders çizelgeleme probleminde öğrenciler önceden atanır, çizelgelemede sadece öğretmen ve dersliklerin çizelgelemede atanması gerekir. Bu problemde katı kısıtlar, aynı zamanda bir öğretmenin birden fazla sınıfa atanmaması, öğretmenlerin atanan periyotlarda mevcut olması ve her öğretmen-sınıf ikilisine doğru sayıda periyot (gün, zaman dilimi) ayırılması gerekir [7]. Bu problem için daha ayrıntılı bilgi, [13, 14] çalışmasında bulunabilir.

1.2.2.2. Üniversite Ders Çizelgeleme (University Course Timetabling)

Üniversitedeki tüm derslerin tüm oturumlarının hafta içinde mümkün olduğunca ortak öğrencileri olan derslerin çakışmasını minimize ederek atanmasını sağlayan bir ders çizelgelemesidir [15]. Katı kısıtlara örnek olarak öğrencileri ortak olan derslerin aynı periyota atanmaması, öğretim elemanlarının atanan periyotta mevcut olması, öğretmenleri ortak olan derslerin aynı periyota atanmaması verilebilir.

1.2.2.3. Sınav Çizelgeleme (Exam Timetabling)

Bir dizi üniversite dersinin sınavlarından ortak öğrencileri olan sınavların çakışmasından kaçınarak ve mümkün olduğu kadar öğrenciler için sınavların yayılmasını sağlayarak çizelgenmesidir. Konuyla ilgili ayrıntılı bilgiler Carter ve Laporte [16, 17] Qu, ve diğerleri [18] bulunmaktadır.

Sınav ve ders çizelgeleme problemlerinin birçok ortak noktası olmasına rağmen, birçok önemli farklılıklara da sahiptir. Örnek olarak sınav çizelgelemede, eğer bir dersliğin sandalye sayısı yetiyorsa farklı derslerin sınavı aynı zamanda aynı dersliğe atanabilir. Oysa, ders çizelgelemede bu mümkün değildir. Yani aynı dersliğe aynı zamanda sadece bir

ders atanabilmektedir. Diğer önemli farklılıklardan birisi de her ders için sadece bir sınav vardır ve çakışma koşulları çok sıkı bir kısıtlamaya sahiptir. Eğer bir öğrenci bazı derslerini alttan alıyorsa, bu derslerin çakışma olasılığı olabilir. Devam zorunluluğu olmayan derslere girmeyebilir. Ancak bu öğrenciler de her ne olursa olsun sınava girmek zorundadırlar. Dolayısıyla sınavların çakışması söz konusu olamaz. Öte yandan, her sınıf için aynı günde en çok bir sınav konmalıdır. Bu sınav tüm sınav dönemine yayılırsa daha iyi olur. Ders çizelgelemenin aksine bir sınıfın sınavları art arda gelmemelidir. Sınav çizelgesinde, ders çizelgesinin aksine sınav zamanı esnek bir yapıya sahiptir.

Ders çizelgelemesinin, Okul çizelgelemesine göre en önemli farkı, üniversitelerde olan derslerin ortak öğrenciye sahip olabilir. Ancak okulda, sınıflar ayrık öğrencilerden oluşurlar. Eğer iki dersin ortak öğrencileri olursa o dersler çakışıyor demektir. Bu dersler aynı zaman dilimine atanamazlar.

Buna ek olarak üniversite probleminde dersliklerin kullanılabilirliği(derslik kapasitesi) önemli bir rol oynar. Oysa okul probleminde bu durum genellikle göz ardı edilir. Çünkü çoğu durumda her sınıfın kendi dersliği olduğunu varsayılır [15]. Okul çizelgelemesi, üniversite ders çizelgesine göre daha sıkı oluyor. Bir gün içerisinde zaman periyotları arasında boşluk olmamalıdır. Okuldaki bütün öğrenciler üniversitenin aksine derslere aynı zamanda başlayıp aynı zamanda bitirirler. Bu öğrenciler farklı gruplara bölünüp her gurubun farklı dersleri olamaz. Üniversite çizelgeleme probleminde ise okula göre çok sayıda ders, program ve farklı grup öğrenciler bulunmaktadır. Bu durumlar üniversite ders çizelgeleme probleminin, okul çizelgeleme problemine göre daha karmaşık bir probleme dönüşmesine neden olmaktadır [7].

1.3. Üniversite Ders Çizelgeleme

Yaklaşık 50 yıldır ders çizelgeleme probleminin üzerinde araştırmalar yapılmaktadır. Eğitim kurumlarının farklı istek ve kısıtları nedeniyle çok farklı çalışmalar ortaya çıkmış ve halen de devam etmektedir. Bunun nedeni yeni kısıtların eğitim kurumları tarafından eklenmesi, bölümlerin müfredatı, kaynakların kullanılabilirliği, isteklerin güncellenmesidir. Üniversite ders çizelgeleme problemi çok yönlü NP-ZOR problem sınıfına girmektedir [13, 19]. Eğer C tane dersimiz olursa ve bu dersleri T tane periyot ve R tane dersliğe atanması gerekirse, toplam $(T * R)^C$ tane farklı çözüm adayını ortaya çıkar. Bu çözüm adayları boyut büyüdükçe daha da büyümektedir.

Tüm bu çizelgeleme problemlerinin ortak birçok yönleri bulunmaktadır. Ders çizelgeleme, derslerin, sınıflarda olan öğrenciler ve öğretmenlere atanması; Ders oturumlarının, derslik ve zaman periyotlarına atandığı çok yönlü bir atama problemi olarak tanımlanmıştır [20]. Araştırmacılar genellikle bu tanımlamayı standart bir tanım olarak kabul edip bunu kullanmaya başlamışlardır. Zaman çizelgeleme problemi gibi, ders çizelgeleme probleminde de kısıtlar iki kategoriye ayrılmaktadır. İzlenmesi gereken katı kısıtlar: Öğrenciler aynı zamanda birden fazla derse atanamaz; derslik dersin gerektirdiği özellikleri sağlamalıdır; derse katılan öğrenci sayısı sınıf kapasitesinden az yâda eşit olmalıdır; bir zaman periyodunda bir derslikte birden fazla ders olmamalıdır. Eğer tüm çizelgeleme uzayındaki bir çizelgelemede bütün katı kısıtlar bozulmaz durumda ise çizelgeye bir mümkün (feasible) çözüm adayı denir. Esnek kısıtlar: Bir öğrencinin her günün son zaman periyodunda bir dersinin olmaması; bir öğrencinin ikiden fazla ardışık dersinin olmaması; bir öğrencinin gün içerisinde tek bir dersinin olmamasıdır [21]. Diğer esnek kısıtlardan, bazı dersler özel zaman periyotlarına atanmalıdırlar, bazı dersler diğerlerinden önce veya sonra gelmelidir.

1.3.1. Problemin Matematiksel Açıklaması

Werra [22] ve Schaerf [12] üniversite ders çizelgeleme programını aşağıdaki gibi tanımlamıştır,

- q ders sayısı ($C_1, C_2, \dots, C_i, \dots, C_q$) (courses),
- k_i, l_i inci dersin (C_i) haftadaki oturum sayısı (lectures),
- r , müfredat (işleme katılan sınıf) sayısı, ($S_1, S_2, \dots, S_m, \dots, S_r$) (curricula), yani ortak öğrencilere sahip bir gruptur. Her müfredat derslerinin farklı zaman periyotlarına atanması gerekmektedir.
- p , zaman periyot sayısı (her periyot gün ve saat ikilisinden oluşuyor)
- l_j, j 'nci periyoduna atanabilen en fazla ders (lectures) sayısı, yani j zaman periyodunda mevcut olan derslik (rooms) sayısıdır.

Bu girdilerle üniversite ders çizelgeleme problemi şöyle tanımlanıyor:

- y_{ij} , ($i = 1, \dots, q; j = 1, \dots, p$) değerleri aşağıdaki koşulları sağlayacak biçimde bulunmalıdır,

$$\circ \sum_{j=1}^p y_{ij} = k_i, \quad (i = 1, \dots, q)$$

Bu kısıt, çizelge oluşturulurken her ders için haftada kaç oturum yapılacağı koşulunu karşılamak zorundadır.

$$\circ \sum_{i=1}^q y_{ij} \leq l_j, \quad (j = 1, \dots, p)$$

Her derslikte bir zaman periyodunda sadece bir ders yapılabilir.

$$\circ \sum_{i \in S_m} y_{ij} \leq 1, \quad (m = 1, \dots, r; \quad j = 1, \dots, p)$$

Aynı zaman periyoduna bir müfredat derslerinin atanması ve çakışması önlenmelidir.

$$y_{ij} = 0 \text{ veya } 1, \quad (i = 1, \dots, q; \quad j = 1, \dots, p)$$

$y_{ij} = 1$ ise C_i dersin bir oturumu j zaman periyoduna atanmış ve $y_{ij} = 0$ ise atanmadığı anlamına gelmektedir.

Werra [22] 1985'de ders çizelgeleme problemi için aşağıdaki amaç fonksiyonunu tanımlayarak problemi bir optimizasyon problemine dönüştürmüştür. Bu amaç fonksiyonu,

$$J = \max \left(\sum_{i=1}^q \sum_{j=1}^p d_{ij} y_{ij} \right) \quad (2)$$

biçiminde verilmektedir. Burada d_{ij} , C_i dersin j zaman periyoduna atanmasının istenilme derecesidir. Bu derece öğretim elemanları veya bölüm yönetimi tarafından belirlenmekte olup, öğretim elemanının i . dersi j . periyotta ne derecede vermek istediğini gösterir.

Bu formül, ilk ortaya atılan formül olmakla birlikte, daha sonraki çalışmalarda başka kısıtlar ve istekler nedeniyle farklı formüller ve farklı d değerleri kullanılmıştır [12].

Bu problem için daha ayrıntılı bilgi Burke ve diğerleri [23, 24, 25]; Lewis [26] çalışmalarında bulunabilir.

1.3.2. Örnek (Benchmark) Problemlerin Üniversite Ders Çizelgelemesi

Önceki bölümlerde sözü edilen ders çizelgeleme problemlerinin yapısı ve doğası, bir üniversiteden diğer bir üniversiteye farklılık göstermektedir. Bunun nedeni her üniversitenin kendine özel istekleri, farklı esnek ve katı kısıtlarının olması değişik değerlendirme ölçütlerinin kullanılmasına neden olmaktadır. Bu probleme ait ortak ve standart bir tanımın eksikliği, bir belirsizliğe neden olmaktadır. Sonuçta problemlerin farklı türde tanımlanması, bu problemler için önerilen çözüm yöntemlerinin farklılaşmasına neden olmaktadır. Bu yöntemlerin sadece ele aldığı problemi çözmesi ve diğer yöntemlerle aynı problemin çözüm sonuçlarının karşılaştırılmamasına neden olmaktadır.

Bu alanda yapılan çalışmalar, bu boşluğu doldurmak için standart bir problemin geliştirilmesini sağlamıştır. Esnek ve katı kısıtların ortak istekler çerçevesinde belirlenmesi, bu belirsizliği ortadan kaldırmıştır. Bu konudaki ilk çalışanlar Carter ve Laporte [20] ve Burke ve diğerleri [25], problem kısıtlarının iki bölüme ayrılmasında ve her birisinin açıklamasında büyük katkıları olmuştur.

Lewis [26] yaptığı doktora çalışmasında ders çizelgeleme probleminin çözümü için önerdiği algoritma, gerçek bir üniversite problemi için değil bilinen örnek problemlerin çözümü için denenmiştir. Lewis niçin gerçek bir üniversite problemini değil de bilinen örnek bir problemi ele aldığını aşağıdaki şekilde açıklamıştır,

- Bu problemin standartlaştırılması için standart bir tanım geliştirilmiş ve test için veriler üretilmiştir. Böylelikle araştırmacıların bu problemin için önerdiği çözüm yöntemlerini diğer araştırmacıların önerdiği çözüm yöntemleriyle karşılaştırmasına imkânı vermiştir. Aynı zamanda araştırmacılara kendi yöntemlerinin başarısını değerlendirmek için bir ölçüt sağlamıştır. Problemlerin çözüm başarısını bir ölçüt olmadan karşılaştırma imkânı yoktu. Yani yöntem başarılı veya başarısız (kalite bakımından) olduğuna karar verilemiyordu.
- İkinci seçme nedeni, gerçek üniversitelerde çok fazla sayıdaki esnek ve katı kısıtlar nedeniyle çözüm için önerilen algoritmanın programa dönüştürülmesi, yani uygulanması çok fazla zaman gerektiriyordu. Aynı zamanda bu işlem bir araştırmacı için zahmetli bir işe dönüşüyordu.
- Gerçek problemde bulunan kısıtlar, çok sayıda özel duruma bağlı olurken az sayıda üniversite için geçerli olmaktadır. Dolayısıyla bu özel problemlere girmek

genel olarak zaman çizelgeleme problemi için bir ölçüt geliştirmeye izin vermiyordu.

Bu çalışmada ders çizelgeleme problemi için önerilen algoritmada kullanılan ölçüt Lewis'in önerdiği ölçütler doğrultusunda yapılmıştır. Ancak Lewis'in 2006 yılındaki çalışmasındaki benchmark problemler [21, 27] tam olarak olgunlaşmamış ve test verileri yapay olarak üretiliyordu. Bu çalışmalardan sonra ölçüt belirleme problemlerinde yeni gelişmeler oldu. Bu gelişmelerin en önemlisi problemler gerçek dünya problemlerine oldukça yaklaşırken pratikte ve teorideki olan boşluk doldurulmaya çalışıldı. Bunun sonucu olarak International Timetabling Competition-2007 (ITC-2007) [28] zaman çizelgeleme yarışması için oluşturulan örnek problem ve test için oluşturulan gerçek üniversite verileri bir ölçüt olarak kabul edildi.

Çalışmada ele alınan ölçüt problem ITC-2007 problemini üçüncü bölümün Curriculum based Course Timetabling (CB-CTT). temel almıştır. Bu problem sonraki bölümlerde detaylı şekilde açıklanacaktır. Bu ölçüt problemin üzerinde çok sayıda çalışmalar yapılmış ve makaleler yayınlanmıştır. Bu ölçütün önerilen yöntem için iyi bir değerlendirme aracı olacağı düşünülmüştür.

Üniversite ders çizelgeleme problemi üzerindeki ilk ayrıntılı çalışma Rossi-Doria ve arkadaşları [29] tarafından 2002 yılında yapılmıştır. Bu çalışmada bir problem ele alınarak farklı yöntemlerle çözümü ortaya konmuş ve sonuçları birbiriyle dengeli bir şekilde karşılaştırılıp değerlendirilmiştir. Bu beş farklı üst sezgisel (meta-heuristic) yöntem, Evrimsel (Evolutionary Algorithm) Algoritma, Karınca Koloni Optimizasyonu (Ant Colony Optimization), Yinelemeli Yerel Arama (Iterated Local Search), ısı işleme (Simulated Annealing) ve Tabu Araması'dır. Bu algoritmalarda esnek ve katı kısıtlar aynı anda ele alınmış ve katı kısıtların ihmal edilmesinin maliyeti, yumuşak kısıtların maliyetinden daha fazla olmaktadır.

Araştırma için ele alınan algoritmalar, çizelgelemeyi oluşturmak için iki aşamalı bir algoritmadan oluşmaktadır. İlk aşamada esnek kısıtları göz ardı edip yalnızca katı kısıtların sağlanmasını amaçlayan mümkün (feasible) bir çözüm üretilmektedir. İkinci aşamada ise esnek kısıtların maliyetinin azaltılmasına çalışılmıştır. Bu araştırmanın sonucu olarak iki aşamalı yöntemlerin daha iyi çözümler verdiği gözlenmiştir.

Bu çalışmadan sonra 2002 yılında Rossi-Doria ve arkadaşları, Avrupa Meta Sezgisel Ağlarda (European Metaheuristic Networks) [30] bulunan diğer araştırmacılarla birlikte,

birinci uluslararası zaman çizelgeleme yarışmasını (International Timetabling Competition, ITC-2002) 2002-2003 yılında düzenlediler.

Zaman çizelgeleme probleminin çözümü için düzenlenen bu yarışmanın amacı, belirli zaman içerisinde ortak ölçüt problemine ve verilerine, yarışmacıların geliştirdiği algoritmaların uygulanması ile elde edilen sonuçların karşılaştırılmasıdır.

Ortak ölçüt verilerine ihtiyaç duyması nedeniyle, bu yarışmayı düzenleyenler yapay şekilde örnek test verileri üretmek için, İngiltere Napier Üniversitesinde Ben Paechter tarafından [31] önerilmiş ve uygulanmış örnek üretici programından yararlanmıştır. Bu program gerçek üniversite problemine benzer, ancak basitleşmiş halde olan test verileri üretmek için kullanılmaktaydı. Yarışma için farklı özelliklere sahip 20 tane örnek test verisi oluşturulmuştur. Bu veriler her birisinin en az bir mümkün çözümü olması koşuluyla üretilmiştir. Bu örneklerin detaylı açıklaması ve yarışmada olan kurallar ve problemin tanımı için yarışmanın resmi sitesinden [27] yararlanılabilir.

1.3.2.1. Uluslararası Zaman Çizelgeleme (ITC-2002)

Rossi-Doria ve arkadaşları, 2002 [29] tarafından tanımlanmış olan örnek problem:

- n tane faaliyet $E = \{e_1, e_2, \dots, e_n\}$,
- 45 tane zaman dilimi (haftada 5 gün ve her gün için 9 zaman dilimi)
 $T = \{t_1, t_2, \dots, t_{45}\}$,
- m tane mevcut dersliği $R = \{r_1, r_2, \dots, r_m\}$,
- k tane öğrenci kümesi $S = \{s_1, s_2, \dots, s_k\}$,
- Her derslik özelliği $F = \{f_1, f_2, \dots, f_1\}$, faaliyete göre dersliğin uygunluğu,

Parametrelerinden oluşmaktadır. Katı ve esnek kısıtlar aşağıda açıklanmaktadır:

Katı kısıtlar:

1. Bir öğrenciye bir zaman diliminde sadece bir faaliyet atanmalıdır.
2. Her derslik atan öğrenci sayısı kadar büyük olmalıdır.
3. Her derslik atan dersin istenilen özelliklerini karşılayabilmelidir.
4. Bir dersliğe aynı zaman diliminde sadece bir faaliyet atanabilir.

Esnek kısıtlar:

1. Öğrencilere her gün için sadece bir ders atanmamalıdır.
2. Öğrencilere her gün için ikiden fazla birbirini izleyen saatlerde ders atanmamalıdır.
3. Öğrenciler bir gün için sadece bir derse ve son zaman dilimine atanmamalıdır.

Normalde yarışma için örnek ölçüt verilerinde 10 tane derslik, faaliyetleri atamak için 450 tane boş yer, 350-400 arasında faaliyet, 5-10 arasında özellik, 200-300 arasında öğrenci bulunmaktadır.

Tablo 1. Örnek problem verilerinin özellikleri

Örnek	Faaliyet Sayısı (<i>n</i>)	Öğrenci Sayısı (<i>s</i>)	Derslik Sayısı (<i>m</i>)	Derslik /Faaliyet	Faaliyet /Öğrenci	Öğrenci /Faaliyet
com01	400	200	10	1.96	17.75	8.88
com02	400	200	10	1.92	17.23	8.62
com03	400	200	10	3.42	17.70	8.85
com04	400	300	10	2.45	17.43	13.07
com05	350	300	10	1.78	17.78	15.24
com06	350	300	10	3.59	17.77	15.23
com07	350	350	10	2.87	17.48	17.48
com08	400	250	10	2.93	17.58	10.99
com09	400	220	11	2.58	17.36	8.68
com10	400	200	10	3.49	17.78	8.89
com11	400	220	10	2.06	17.41	9.58
com12	400	200	10	1.96	17.57	8.79
com13	400	250	10	2.43	17.69	11.05
com14	350	300	10	3.08	17.42	17.42
com15	350	220	10	2.19	17.58	15.07
com16	440	300	11	3.17	17.75	8.88
com17	350	300	10	1.11	17.67	15.15
com18	400	200	10	1.75	17.56	8.78
com19	400	300	10	3.94	17.71	13.28
com20	350	300	10	3.43	17.49	14.99

Üsteki kümeler arası bağlantılar 5 ayrı matris ile gösterilir. Birinci matris $A_{k,n}$ Öğrenci/Faaliyet matrisi olarak adlandırılıyor. Bu matris her faaliyetin hangi öğrencilere verilmesi gerektiğini göstermektedir. $A_{k \times n}$ 'de, $a_{i,j}$ 'nin değeri 1 ise i . ($i \in S$) öğrenci kümesine, j . ($j \in E$) faaliyetin verilmesi zorunludur. 0 olması durumunda ise böyle bir zorunluluk bulunmamaktadır.

İkinci matris $B_{n,n}$, Faaliyet/Çakışma matrisi olarak adlandırılıyor. Bu matris, iki dersin aynı zaman dilimine atanıp atanamamasını göstermektedir.

Üçüncü matris $C_{m,l}$, Derslik/Özellik matrisi olarak adlandırılıyor. Her dersliğin hangi özelliklere sahip olduğunu göstermektedir. $c_{i,j}$ 'nin değeri 1 ise i . ($i \in R$) dersliğin, j . ($j \in F$) özelliğe sahip olduğunu gösterir. 0 olması durumunda ise dersliğin o özelliğe sahip olmadığını gösterir.

Dördüncü matris $D_{n,l}$, Faaliyet/Özellik matrisi olarak adlandırılıyor. Her dersin hangi özelliklere ihtiyaç duyduğunu göstermektedir. Eğer $d_{i,j} = 1$ ise i . ($i \in E$) faaliyetin, j . ($j \in F$) özeliğine ihtiyacı var demektir.

Sonuncu matris $G_{m,n}$, Faaliyet/Derslik matrisi olarak adlandırılmaktadır. Her faaliyet için atanmaya mümkün olan derslikleri göstermektedir. Yani her ders için istenen özellikleri ve sandalye sayısının uygun olup olmadığını denetler.

Bu yarışmayı düzenleyenler belirlenen zaman içinde yarışmacılardan elde edilen sonuçları değerlendirip karşılaştırdıktan sonra bu yarışmayı kazanan kişilerin isimlerini ve önerdikleri algoritmaları açıklamışlardır. Bu açıklamalar yarışmanın resmi sitesinde bulunmaktadır [27].

ITC-2002 zaman çizelgeleme probleminin üzerinde çalışan grupların düşüncelerini gerçekleştirmek için ortak bir temeller üretmek adına çok başarılı olmuştur.

1.3.2.2. Socha'nın Örnek Problemi

Socha ve arkadaşları [21] tarafından ele alınan problem aynen ITC-2002 deki ölçüt problemi gibidir. Yalnızca diğerinden farkı önerdiği örnek test verileridir. Socha ve arkadaşları bu örnek verileri üretmek için aynen diğeri gibi Ben Paechter tarafından [31] önerilen programdan yararlanmışlardır. Üretilen örnek veriler üç ayrı kategoride tanımlamıştır. Bunlar küçük (small), orta (medium) ve büyük (large) örneklerdir. Tablo 2'de örnek verilerin kategorik özellikleri verilmektedir.

Tablo 2 Üretilen örnek verilerin özellikleri

Sınıflar	Küçük	Orta	Büyük
Faaliyet sayısı	100	400	400
Derslik sayısı	5	10	10
Özellik sayısı	5	5	10
Her dersin yaklaşık özelliği	3	3	5
Özellik kullanım yüzdesi	70	80	90
Öğrenci sayısı	80	200	200
Öğrenci başına düşen en fazla faaliyet sayısı	20	20	20
Faaliyet başına düşen en fazla öğrenci sayısı	20	50	100

Lewis 2006 da yaptığı doktora tez çalışmasında [26] ve Obit [7] 2010da yaptığı doktora tezinde ders çizelgeleme probleminin çözümü için önerdikleri algoritmaları bu örnek problemler üzerinde denemişlerdir.

McCollom 2006 da altıncı serisi düzenlenen PATAT konferansında yayınladığı bildiride [32] eğitimsel çizelgeleme probleminin her iki dalında (sınav çizelgeleme ve ders çizelgeleme) çalışan araştırmacıların ilgisini teorik problemler ile gerçek dünyadaki eğitim kurumların ihtiyaçları arasındaki boşluğa çekmeye çalıştı. Yani gerçek dünyada olan problemlerin ihtiyaçları araştırmacıların yaptığı çalışmalarda tam olarak göz önüne alınmadığını vurgulamıştır.

McCollum bu çalışmadaki amacı üniversitelerde yapılan teorik araştırmalar ile gerçek dünyadaki pratik uygulamalar arasında bir köprü oluşturmaktır. Bunlara ek olarak 2006 yılına kadar iki dalda üretilen bütün örnek problemler üniversitelerin gerçek isteklerini karşılamada yetersiz olduğunu ve yeni bir problem için standart tanıtım ve örnek ölçüt verilerinin oluşturulma gerekçesini vurgulayarak araştırmacıların ilgisini bu noktaya çekmeyi başardı.

Bu tarihten sonra çok sayıda gruplar ve komiteler, teorik ve pratik çalışma arasında olan boşluğu azaltmak amacıyla oluşturuldu. Bunun sonucu ikinci kez düzenlenen uluslararası zaman çizelgeleme problemi yarışması ITC-2007 kendisini gösterdi.

Bu yarışma Udine üniversitesi, Napier üniversitesi, Queen üniversitesi, ve Cardiff üniversitesinin katılımıyla, Nottingham üniversitesi ev sahipliğinde İngiltere’de düzenlendi.

Bu yarışmayı düzenleyenlerin amacı ilk yarışmada olduğu gibi eğitimsel çizelgeleme probleminin üzerinde yapılan araştırmaların gelişmesi için iyi bir ortam hazırlamak, katılımcılar sayesinde yeni yöntemler üretmek ve ek olarak McCollum [32] söylediği yapay zekâ araştırmalarında ve yöneylem araştırmacıların arasında olan boşluğu daraltmaktır. Bu yarışmanın ayrıntılarına yarışmanın resmi sitesinden ulaşmak mümkündür [28].

2007 yılının sonuna kadar üniversite zaman çizelgeleme problemi için çok sayıda farklı formüller vardı. Bu formüllerin her birisi farklı gerçek üniversiteleri ele almıştı. ITC-2007'nin formülleştirmesinde gerçek üniversitelerin ortak istek ve ihtiyaçları ve yarışmada olan isteklerin arasında bir denge yaratmasına önem verilmiştir.

Birinci yarışmanın (ITC-2002) başarılı olması nedeniyle ITC-2007 de konu daha derin ve karmaşık ele alındı. Yarışma yalnızca bir dal için değil üç farklı dalda (tracks) düzenlendi. Katılımcıların bir, iki veya her üç dalda da yarışmaya katılmaları mümkün olmuştur. Bu dalların bazı ortak yönleri olmasına rağmen, pratikte ve teorik araştırmalarda problemi farklı açılardan ele almışlardır.

Bu dallar (tracks) sınav çizelgeleme problemi ve iki türlü üniversite ders çizelgeleme probleminden oluşmaktadır. Ders çizelgeleme problemi kendi içinde iki farklı dala ayrılmaktadır. Bu dallar, farklı üniversitelerde, ders çizelgesi üretme yöntemleri, ders kaydı sonrası ders çizelgeleme ve müfredat tabanlı çizelgeleme için kendilerine ait farklı örnek ölçüt verileri tanımlanmıştır.

Çok sayıda teknik raporlar bu yarışmanın her bölümü (track) için yayınlanmıştır. Bu raporlar, her bölümün ayrıntılı şekilde açıklamasını ve problemin nasıl formülleştirilmesi gerektiğini açıklamıştır. Bu örnek raporlardan birkaçı McCollom ve arkadaşları [33], Lewis ve arkadaşları [34], Di Gaspero ve arkadaşları [35] ve yarışmanın resmi sitesinde bulunmaktadır [28].

ITC-2002 ve ITC-2007 de olan farklılıklar bazı makalelerde ayrıntılı bir şekilde açıklanmıştır. Bunlardan birisi 2007 yılında Abdullah ve arkadaşları [36] tarafından yayınlanmıştır.

Bu farklılıklardan en önemlisi, ITC-2007 eğitimsel çizelgeleme probleminin üç dala ayrılmasıdır. İkincisi, problemin her dalında olan formülleştirmenin gerçek üniversitedeki problemlere oldukça benzer olmasıdır. Üçüncü fark, ITC-2002 de sadece mümkün (feasible) çözümler kabul edilip değerlendirmeye alınırken, ITC-2007 bazı örnek test verileri için mümkün olmayan (infeasible) çözümler de değerlendirmeye alınmıştır.

Bu çalışmada ITC-2007'nin üçüncü bölümü (track), yani Müfredat Tabanlı Üniversite Ders Çizelgeleme Problemi (Curriculum-based timetabling problem, CB-CTT) ele alındığı için aşağıdaki bölümlerde sadece bu problemi ayrıntılı bir şekilde açıklanmaktadır.

1.3.2.3. Müfredat Tabanlı Üniversite Ders Çizelgelemesi

Müfredat tabanlı üniversite ders çizelgeleme problemi, birçok üniversite derslerinin hafta içerisinde, belirli sayıda dersliklere belli zaman periyotları arasında, derslerin birbiriyle çakışmamasını sağlayacak şekilde atanmasıdır. Dersler arasında çakışmalar derslere yazılan öğrencilere göre değil, her üniversitenin yayınlamış olduğu müfredat üzerinden belirlenir.

İtalya'nın Udine üniversitesinde ve diğer birçok İtalyan ve uluslararası üniversiteler problemin formüllemesini ve tanımını yaparak kullanmışlardır. Bu çalışmada daha gerçekçi olan Udine üniversitesinin kullandığı örnek problemin basitleştirilmiş hali kullanılmıştır.

Bu basitleşmiş problem nedenleri ve hangi bölümlerinde olmasının açıklaması McCollum ve diğerleri [33] tarafından yayınlanan makalede geniş şekilde ele alınmıştır.

Bu problem aşağıdaki kısımlardan oluşmaktadır:

- **Gün, Zaman Dilimleri ve Periyotları:** Hafta içerisinde belirlenen öğretim günleri (normalde 5 ya da 6 gündür). Her gün belirlenen sayıda zaman dilimlerine ayrılır ve bu bütün günler için eşittir. Bir periyot, gün ve zaman dilimi ikiliğinden oluşur. Çizelgede atama için zaman periyotlarının toplam sayısı, günlerin sayısı ile zaman dilimlerinin sayısının çarpımıdır.
- **Dersler ve Öğretim Elemanları:** Her ders haftada belirlenen sayıda yapılmalıdır ve her biri farklı periyotlara atanmalıdır. Her ders için dersi alan öğrenci sayısı ve dersi veren öğretim elemanı dosya halinde verilmektedir. Her dersin haftada minimum kaç günde yapılması gerektiği veri olarak verilmektedir.
- **Derslikler:** Her derslik belli bir kapasiteye sahiptir. Bu kapasite derslikte olan sandalye sayısını gösterir. Her dersliğin kapasitesi yeterince büyük ise her ders için uygundur.
- **Müfredat:** Bir müfredat bir ders grubundan oluşmaktadır. Her grup içerisinde ele alınan herhangi iki dersin ortak öğrencileri bulunabilmektedir. Dersler arasındaki çakışmaların oluşturduğu katı kısıtlar ve esnek kısıtlar müfredat üzerinden belirlenmektedir.

Bu problemin çözümü için her dersin tüm oturumları bir dersliğe ve bir periyoda atanmalıdır.

Problemden bulunan katı kısıtlar:

- **Ders Oturumları:** Bir dersin oturumlarının hepsinin farklı periyotlara atanması gerekir. Her oturumun aynı periyoda atanması veya hiç birine atanmaması hata oluşmasına neden olur.
- **Derslik Kullanımı:** Aynı zaman periyodunda bir dersliğe birden fazla ders atanamaz. Aynı zaman periyodunda aynı dersliğe iki dersin atanması bir hata oluşmasına yol açar ve aynı periyoda aynı derslikte her fazla dersin olması bir artık hataya sebep olur.
- **Çakışmalar:** Bir müfredatta olan derslerin oturumları farklı periyotlara atanmalıdır. Öte yandan bir öğretim elemanının verdiği derslerin oturumları da farklı periyotlara atanmalıdır. Her periyoda olan iki oturumun çakışması bir hata oluşmasına neden olur.
- **Uygunluk:** Eğer bir periyod için dersi veren öğretim elemanı müsait değil ise bu dersin hiç bir oturumu bu periyotta atanamaz. Bu şekilde yapılan her atama bir hata oluşmasına neden olur.

Esnek kısıtlar aşağıda verilmektedir:

- **Derslik kapasitesi:** Her oturuma ait öğrenci sayısı, o oturumun atandığı dersliğin sandalye sayısına eşit veya küçük olmalıdır. Dersliğin kapasitesinden fazla olan her bir öğrenci bir ceza puanı alınmasına neden olur.
- **Minimum İş Günü:** Verilen her dersin oturumlarının günleri o dersin minimum verilme gün sayısına göre dağıtılmalıdır. Her eksik gün için 5 ceza puanı alınmasına neden olur.
- **Müfredat Yoğunluğu:** Aynı müfredat içerisinde aynı güne atanan oturumlar peş peşe olması istenir. Her müfredatın herhangi bir gün içindeki bir oturumunun komşusu yoksa 2 ceza puanı alınmasına neden olur.
- **Sabit Derslik:** Her dersin tüm oturumları aynı dersliğe atanmalıdır. Aynı dersliğe atanmayan her ders bir ceza puanı alınmasına neden olur.

Bu problemde 21 tane örnek test verisi ele alınmaktadır. Udine Üniversitesinden alınan bu örnekler gerçek verilerdir. Hepsi için en az bir tane mümkün çözüm bulunmaktadır. Bu çözümlerin hepsi tüm katı kısıtları karşılamaktadır. Bu problemdeki örneklerin haftadaki iş günleri farklı olabilir. Tablo 3'da problemin tüm örneklerinin bilgileri ayrıntılı bir şekilde verilmektedir.

Tablo 3. Örnek verilerin bilgileri

Instance	C	L	R	PPD	D	Cu	MML	CL	RO
comp01	30	160	6	6	5	14	2-5	3.24	88.9
comp02	82	283	16	5	5	70	2-4	2.62	70.8
comp03	72	251	16	5	5	68	2-4	2.36	62.8
comp04	79	286	18	5	5	57	2-4	2.05	63.6
comp05	54	152	9	6	6	139	2-4	1.8	46.9
comp06	108	361	18	5	5	70	2-4	2.42	80.2
comp07	131	434	20	5	5	77	2-4	2.51	86.8
comp08	86	324	18	5	5	61	2-4	2	72
comp09	76	279	18	5	5	75	2-4	2.11	62
comp10	115	370	18	5	5	67	2-4	2.54	82.2
comp11	30	162	5	9	5	13	2-6	3.94	72
comp12	88	218	11	6	6	150	2-4	1.74	55.1
comp13	82	308	19	5	5	66	2-3	2.01	64.8
comp14	85	275	17	5	5	60	2-4	2.34	64.7
comp15	72	251	16	5	5	68	2-4	2.36	62.8
comp16	108	366	20	5	5	71	2-4	2.39	73.2
comp17	99	339	17	5	5	70	2-4	2.33	79.8
comp18	47	138	9	6	6	52	2-3	1.53	42.6
comp19	74	277	16	5	5	66	2-4	2.42	69.2
comp20	121	390	19	5	5	78	2-4	2.5	82.1
comp21	94	327	18	5	5	78	2-4	2.25	72.7
Toy	4	16	3	4	5	2	2-3	2.1	26.7

Burada;

- C (Courses): Derslerin sayısını,
- L (Total Lecture): Tüm derslerin toplam oturumlarını,
- R (Room): Dersliklerin sayısını,
- PPD (Periods Per Day) : Bir gündeki zaman dilimini,
- D (Days): Haftadaki iş günü sayısını,
- Cu (Curricula): Müfredat sayısını,

- MML (Min and Max Lectures per day per curriculum): Her gün için her müfredata düşen en az ve en çok oturum sayısını,
- CL (average number of lectures per curriculum per day): Her müfredat için günlük ortalama oturum sayısını,
- RO (average room occupation): Her zaman dilimi için uygun olan ortalama derslik sayısını,

vermektedir.

1.4. Literatür Tarama

Eğitimsel çizelgeleme otomatikleştirme konusunda ilk çalışmalardan birisi 1963 Gotlieb [37] tarafından yöneylem araştırma teknikleriyle yapılmıştır. Yaklaşık 50 yıl önceden zaman çizelgeleme probleminin otomatikleştirilmesi araştırmacıların ilgisini çekmiştir. Çok sayıda çalışma ve yöntem gerçek eğitim kurumlarının problemini çözmek için yapılmış ve halen yapılmaya devam etmektedir. Bunun nedeni; günlük yaşamda eğitim kurumları tarafından istenen kısıtların ve isteklerin gün geçtikçe artması, yüksek derecede başarının elde edilmesi, zamandan (kısıtlı kaynaklardan) tasarruf yapılması ve en önemlisi modern dünyadaki işlerin daha karmaşık hale gelmesidir.

Yöneylem araştırmasındaki klasik yöntemler bu konuda yetersiz kalmaktadır. Bilgisayarların yaygınlaşmasıyla birlikte yapay zeka tekniklerinin gelişmesi, yeni hesaplama tekniklerini ortaya çıkarmış ve problem bu tekniklerle de ele alınmıştır.

Carter ve Laporte 1998 [16] yılına kadar ders çizelgemesi probleminin üzerinde yapılan çalışmalarda kullanılan yöntemleri 4 sınıfa ayırmış ve şu ana başlıklar altında toplamıştır. Sıralı Yöntemler (Sequential Methods), Küme Yöntemleri (Cluster Methods), Kısıt Tabanlı Yöntemler (Constraint Based Approach) ve Meta Sezgisel Yöntemlerdir (Meta-Heuristic Methods).

Yapılan sınıflama sonraki yıllarda yapılan çalışmalar için kaynak olmuştur. 2004 yılında Petrovic ve Buruke [38], yapılan sınıflandırmaya 4 sınıf daha ekleyerek sınıflandırmayı tamamlamıştır.

Eklenen sınıflar; Melez Sezgisel Yöntemler ve Meta Sezgisel Yöntemler (Hybrid Heuristic and Meta-Heuristic), Çok Ölçütlü Yöntemler (A Multicriteria Approach), Durum Tabanlı Muhakemeli Yaklaşımlar (Case-based reasoning approaches), Uyarlamalı Yöntemler (Adaptive Methods) dir.

1.4.1. Sıralı Yöntemler

Bu yöntem, faaliyetleri ortamsal sezgisel kullanarak bulunan değerler üzerinden azalan sıralama yapar. Daha sonra bu faaliyetler sırasıyla alınıp periyottaki diğer faaliyetlerle çakışmama koşuluna göre boş olan zaman periyotlarına atanır. Bu yöntemler genellikle zaman çizelgeleme problemini bir grafla göstermektedir. Faaliyetler grafin düğümleridir. Eğer bir zaman periyodunda iki faaliyet çakışiyorsa, faaliyetleri temsil eden düğümler arasında bir bağlantı var demektir. Bu problemin çözümü ise oluşturulan grafin renklenmesidir.

Çizelgedeki her bir zaman periyodu grafta bir renge denk gelir. Eğer faaliyetler arasında çakışma varsa çakışan faaliyetleri temsil eden düğümler farklı renklenir.

Kromatik sayısı (Chromatic number, χ), graf renkleme probleminde kullanılan en az sayıdaki renktir. Bu değer çakışmasız çizelgeleme oluşturmak için ihtiyaç duyulan en az zaman periyoduna karşılık gelmektedir. Graflarda bu değer belirlenmesi zor bir işlemdir ve Zor NP Sınıfı problemidir.

Graf renkleme problemine paralel bir problem ise grafta Klik (Clique) ile tanımlanan özelliğin belirlenmesidir. Klik; bir grup düğüm arasında birbiriyle bağlantısı olan düğümlerin oluşturduğu gruptur. Bu grupta olan düğümlerin sayısı Klik'in boyutunu oluşturur.

Grafa dönüştürülen faaliyet çizelgeleme problemindeki her Klik'in üyeleri aynı zaman periyoduna atanamaz. Graf renklemede her bir Klik düğümü farklı renktedir.

Oluşturulan grafta bulunan Kliklerin en büyük boyutu C olursa grafi renklendirmek için en az C tane renge ihtiyaç duyulur. ($\chi \geq C$) Graflarda en büyük klik boyutunu belirlemenin kendini de bir NP-Zor problemidir [26].

Literatürde çakışmayan zaman çizelgeleri oluşturmak için graf renklendirmede farklı sezgisel yöntemler bulunmaktadır. Bu sezgisel yöntemler (event), faaliyetleri atama zorlukları tahminine göre sıralar. Sıkça kullanılan sezgisel yöntemler şunlardır;

- Önce en büyük derece (Largest degree first):
Diğer faaliyetlerle çakışması fazla olan faaliyetlerin atanması önce yapılır. Bunun nedeni atanması en zor olan faaliyetler, çakışması çok olan faaliyetlerdir. Örnek: Bu işlem ITC-2007 yarışmasının üçüncü bölümünde olan ToyExample örneği üzerinde uygulanır. ToyExample örneğinin ders özellikleri Tablo 4'de verilmektedir.

Tablo 4. ToyExample örneğinin ders özellikleri

Derslerin ismi	SceCosC	ArcTec	TecCos	Geotec
Hocasının ismi	Ocra	Indaco	Rosa	Scarlati

Bu örnekte iki tane müfredat bulunmaktadır. Bunlar,

Cur1: (SceCosC, ArcTec, TecCos)

Cur2: (TecCos, Geotec) ‘dır.

Müfredat ve derslerin öğretim elemanları üzerinden çakışma matrisi Tablo 5’de görüldüğü gibi (clash matrix) oluşturulur.

Tablo 5. Çakışma matrisi

Dersler	Dersler			
	SceCosC	ArcTec	TecCos	Geotec
SceCosC	False	True	True	False
ArcTec	True	False	True	False
TecCos	True	True	False	True
Geotec	False	False	True	False

Çakışma matrisi (clash matrix) üzerinden her dersin derecesi hesaplanır. Bu dereceler (LDF) Tablo 6’de verilmiştir.

Tablo 6. Derslere verilen dereceler

Ders	Derece değeri
SceCosC	2
ArcTec	2
TecCos	3
Geotec	1

Derslerin sıralaması bu değerlerin azalmasına göre [TecCos, ArcTec, SceCosC, Geotec] şeklinde olur.

Buna göre ilk olarak TecCos dersinin oturumları atanmalıdır sonra diğer oturumlar aldığı değer sırasıyla atanmalıdır.

- Önce en büyük ağırlıklı derece (Largest weighted degree):
Bu yöntemde önce en büyük derece yöntemine benzetilmektedir. Aralarındaki fark bu yöntemde çakışmalar öğrenci sayısına göre ağırlıklandırılır.
- Satürasyon derecesi (Saturation degree):
Zaman çizelgesi oluştururken her adımda, o ana kadar oluşturulmuş zaman çizelgesinde bulunan atanma için uygun en az sayıda zaman periyoduna sahip faaliyet seçilir.
- Renk derecesi (Colour degree):
Bu sezgisel yöntem atanmamış faaliyetler içerisinde atanmış faaliyetlerle çakışması fazla olanı seçer.

Bu sezgisel yöntemlerden birini kullanıp faaliyetleri (event) sıraladıktan sonra boş periyot seçiminde her faaliyetin atanması için bir kaç yaklaşımdan faydalanılabilir. Örneğin, en erken bulduğumuz uygun boş zaman periyodu veya amaç fonksiyonu değerine göre en iyi uygun periyot seçilir [23].

Üniversite ders çizelgeleme probleminin çözümü için üstte açıklanan sıralı yöntemler, uygulanan ilk ve en yaygın yöntemlerden birisidir. Bu konu ayrıntılı şekilde [30] kaynağında ele alınmaktadır. Burke ve diğerleri [39, 40], L. Kiaser ve diğerleri [41] çalışmalarında bu yöntemden faydalanmışlardır.

Açıklanan sıralı yöntemler, ders çizelgeleme probleminin boyutu büyüdüğünde problemin çözümü için yetersiz kalır. Bu nedenle araştırmacılar bu yöntemi diğer yöntemlerle birleştirip melez yöntem şeklinde kullanmaktadırlar [42].

Üniversite ders çizelgeleme probleminin çözümünde yöneylem araştırmacı tarafından çok kullanılmış ve iyi başarılar elde edilmiş, diğer sıralı yöntemlerden birisi matematiksel programlama tekniğidir. Bu tekniklerden birisi doğrusal tamsayı programlama (Integer Linear Programing, ILP) tekniğidir. Bu yöntemin ders çizelgeleme problemi üzerinde kullanımı 1969 yılına kadar uzanır [44].

Literatürde bu yöntemle ders çizelgeleme problemin üzerinde yapılmış çalışmalar [45, 46, 47] 'dır. En son çalışmalardan birisi G. Lach ve M.E. Lübbecke 2010 [48] ve Hao, Benlic 2011 [49]' dır.

Matematiksel programlama tekniklerinden bir diğeri ise, Karma Tamsayılı Programlama'dır. Burke ve diğeri 2008 [50] Burke ve arkadaşları 2010 [51] bu tekniğı kullanarak ders çizelgeleme problemini çözenlere örnek olarak verilebilir.

1.4.2. Kümeleme Yöntemleri

Bu yöntemde faaliyetler, katı kısıtları sağlayan gruplar şeklinde bölünürler. Sonra bu gruplar esnek kısıtları sağlamak şartıyla zaman periyotlarına atanır. White ve Chan 1979 [52] bu yöntemle eğitimsel çizelgeleme problemini ilk ele alan araştırmacılardandır. Faaliyet gruplarını zaman periyotlarına atamak için farklı optimizasyon tekniklerinden yararlanılır. Bu yöntem ile yapılan çalışmalara [53] kaynağı örnek gösterilebilir.

1.4.3. Kısıt Tabanlı Yöntemler

Bu yöntemlerde zaman çizelgeleme problemi bir küme değişkenlerle yani faaliyetlerle modellenir. Değişkenler belirli sayıda kısıtları sağlamak koşuluyla zaman periyodu, dersi gibi değerler almaktadır.

Kaynakların faaliyetlere atanması için genellikle belirli sayıda kurallar tanımlanır. Eğer hiç bir kural çözümün şimdiki durumuna uygun değilse geri dönüş yapılır ve uygun bir çözüm bulunana kadar, yani bütün kısıtlar sağlanana kadar işleme devam edilir.

Kısıt Sağlama Problemleri (Constraint Satisfaction Problems, CSP) yöntemi kısıtları sağlayarak kaynakları dağıtım problemlerin çözümünde kullanılan en önemli yöntemlerden sayılır.

Eğer bir problem CSP problem türüne dönüşürse elde edilen problemin çözümü için Kısıt Mantıklı Programlama (Constraint Logic Programming) kullanılabilir. Ders çizelgeleme problemini çözmek için [54] kaynak örnek gösterilebilir. Bu çalışmada Aten Üniversitesinin ders çizelgesinin oluşturulması ele alınmıştır.

Muller [55] 2007 yılında düzenlenen ITC-2007 yarışmasının üçüncü bölümü için CSP yöntemi kullanarak birinci oldu. Aynı yarışmada üçüncü çıkan kişilerden Mitsunori Atsuta ve diğeri [56] CSP yönteminin farklı türünü kullanmışlardır.

1.4.4. Meta Sezgisel Yöntemler

Son 20 yıl içerisinde Meta Sezgisel yaklaşımların üniversite ders çizelgeleme probleminin çözümü için araştırmalarda kullanımı artış göstermektedir. Örneğin, Tabu Araştırma (Tabu Search), Isıl İşlem (Simulated Annealing), Genetik Algoritma (Genetic Algorithm) ve bunların melez halidir. Bunlardan birkaçı serisi düzenlenen PATAT [9] konferansından çıkan yayın kitapçıklarında bulunmaktadır [20, 57].

Bu yöntemlerin hepsi bir veya birden fazla başlangıç çözümlerinin üzerinden bölgesel optimuma takılmayan arama tekniklerinden yararlanarak iyileşmeyapmaya çalışır. Bu yöntem kaliteli sonuçlar üretmenin yanı sıra yüksek hesaplama maliyetine de sahiptir.

Lewis 2006 yılında yaptığı doktora tezinde [25], 2008 de yayınladığı makalede [58] üniversite ders çizelgeleme probleminin üzerinde Meta sezgisel yöntemlerin kullanımını ayrıntılı bir şekilde araştırmıştır. Bu yöntemlerin ayrıntılı açıklamasına sonraki bölümde yer verilmiştir.

1.4.4.1. Lewis Meta Sezgisel Yöntemlerin Sınıflandırılması

Lewis [25] çalışmasında bu yöntemleri katı ve esnek kısıtları sağladığı zaman üzerinde üç ana sınıfa bölmüş ve literatürde bulunan üniversite ders çizelgeleme problemi üzerinde yapılan çalışmaları ve yöntemlerin hepsini alt başlıklar altında toplamıştır.

1.4.4.1.1. Tek Aşamalı Optimizasyon Algoritmalar

Bu tür algoritmalarda esnek ve katı kısıtların aynı zamanda sağlanması ve karşılanması ele alınır. Genellikle tek aşamalı optimizasyon algoritmaları katı ve esnek kısıtların çizelge içerisinde ihmaline izin verir. Daha sonraki amaç ise her iki kısıtı sağlayacak biçimde uygun bir çözüm aramaktır. Normalde bu ağırlıklı toplam fonksiyonu kullanılarak katı kısıtların ihmaline yüksek ceza, esnek kısıtların ihmaline de düşük ceza verilir. Aşağıdaki fonksiyon buna bir örnektir. Burada k tane kısıtla bir problem göz önüne alınır ve her i . kısıt için ağırlık cezası w_i tanımlanır, $v_i(T)$ i . kısıtın T zaman çizelgesindeki ihlal sayısı temsil etsin. T zaman çizelgesinin kalitesi şu formül kullanılarak hesaplanabilir:

$$f(T) = \sum_{i=1}^k w_i v_i(T) \quad (3)$$

Bu yöntem iki avantaja sahiptir. Birincisi; amaç fonksiyonunu minimum yapma işlemi bir aday çözüm üretimine karşılık gelir. İkincisi ise bu yaklaşımın uygulaması kolay ve esnektir. Her ek kısıt mantıklı ceza değeriyle formüle eklenebilir.

Bu nedenle literatürde bu tipte çalışan çok sayıda zaman çizelgeleme algoritması önerilmiştir [59, 60].

1.4.4.1.2. İki Aşamalı Optimizasyon Algoritmalar

Çizelgeleme problemi için iki aşamalı optimizasyon algoritmalarının operasyonel özellikleri aşağıdaki gibi özetlenebilir: Genellikle ilk aşamada esnek kısıtlar göz önüne alınmaz sadece optimizasyon problemindeki katı kısıtlar ele alınarak mümkün çizelge aranır. İkinci aşamada bulunan mümkün çizelgenin üzerinde esnek kısıtların ihlal ceza sayılarının mümkün olan bölgelerde arama teknikleri kullanılarak azaltılmasına çalışılır.

Literatürde bu yöntemle yapılan çalışmalarda, özellikle ITC-2007 ve ITC-2002 benchmark örnek probleminin üzerinde kullanılmıştır. Bunlara Socha [21], Arntzen H, Løkketangen [61], Lewis R, Paechter B [62], Salwani Abdullah ve diğerleri [63], Kostuch P ITC-2002 [64] örnektir. Aynı zamanda Lewis 2006 [26] ve Obit'in 2010 [7] da doktora tezi için önerdiği algoritma bu türdendir.

1.4.4.1.3. Gevsemeye İzin Veren Algoritmalar

Baştan beri katı kısıtların her birisinin sağlanmaması için oluşan hatalar problemin diğer kısıtları rahatlatıcı tarafından iptal edilip ve sonra esnek kısıtların sağlamasına çalışılıyor. Bu yaklaşımlar için ayrıntılı bilgiler ve örnekler [58] verilmektedir.

Obit [7] ders çizelgeleme probleminde kullanan meta sezgisel yöntemleri farklı şekilde sınıflandırmıştır. Sınıflandırmaların ayrıntılı açıklaması aşağıda verilmektedir:

1.4.4.2. Obit Meta Sezgisel Yöntemlerin Sınıflandırılması

Obit meta sezgisel yöntemleri, başlangıçta ele aldığı çözümlerin sayısına bağlı olarak iki sınıfa ayırmıştır.

1.4.4.2.1. Popülasyon Tabanlı Meta Sezgisel Yaklaşımlar

Bu yöntemlere bazen çeşitli nokta (multiple-point) yöntemleri de denilir. Bunlar bir popülasyonda farklı çözümlerden oluşur. Bu yöntem, her iterasyonda uygun bir seçme mekanizmasından yararlanılarak popülasyon içerisinde bir çözüm seçilip onu güncelleştirmek ve şimdiki çözümden yeni bir çözüm üretmek için kullanılır. Çoklu sayıda seçme mekanizmi vardır.

- Seçme yöntemleri:

- Delete all: Popülasyondaki bütün çözümleri siler ve her birini yeni bir çözümle değiştirir.
- Steady-state: Popülasyon içerisinde n tane çözümü seçer ve onları yeni üretilmiş çözümle değiştirir.
- Tournament selection: Popülasyon çözümleri içerisinde rastgele olarak bir alt küme seçer ve küme içerisinde bir sezgisel kullanarak sezgisele ait en büyük değerli olanı seçer.
- Bias selection: Çözüm listesinden baştaki ilk n tanesini rastgele olarak seçer.
- Rulet Tekerleği Seçimi (Roulette Wheel Selection): “Bu yöntemde seçilme işlemi bireylerin uygunluk değerlerine göre yapılır. Bireylerin uygunluk değerinin büyük olmasına rağmen seçilmesi garanti edilemez sadece seçilme şansı daha fazla olur.

Bu yöntemde tüm bireylerin uygunluk değerleri bir tabloya yazılır ve toplanır. Uygunluk değerleri toplama bölünerek bireylerin $[0, 1]$ aralığında seçilme olasılığı belirlenir [65].

Daire şeklinde olan rulet tekerleğinin üzerinde popülasyondaki her bireyin kalite değerine popülasyonun ortalama kalite değerinin oranına göre bir alan verilir. Dolayısıyla değeri fazla olanın alanı da büyüktür. Böylece rulet tekerleği çevrildiğinde topun bu alana düşme olasılığı fazla olacaktır. Bu

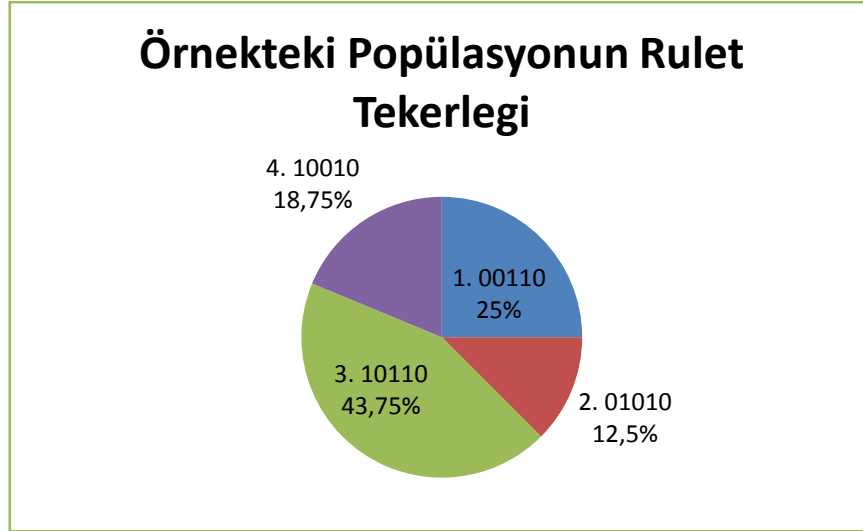
durumda Tablo 7’de verilen bilgilere sahip dört tane bireyden oluşan bir popülasyon için şekil 4’deki grafikte gösterilmektedir [3].

Popülasyondaki i . bireyin olasılığı 4 formülüne göre hesaplanmaktadır.

$$p_i = \frac{fitness_i}{\sum_{i=1}^n fitness_i} \quad (4)$$

Tablo 7. Bireylerin bilgileri

Sayı	Dizi	Kalitesi	Toplam(%)
1	00110	100	% 25
2	01010	50	% 12,5
3	10110	175	% 43,75
4	10010	75	% 18,75
Toplam		400	100,00



Şekil 4. Rulet Tekerlik üzerinde dizilerin seçme olasılığı.

Popülasyon tabanlı algoritmalarından en yaygın olanı genetik algoritmadır [66]. Zaman çizelgeleme ve özellikle ders çizelgeleme probleminin çözümü için temel genetik algoritmadan veya genetik algoritmanın melez halinden yararlanılmıştır. Bu çalışmalardan ilki [67] ile yapılmıştır çalışmadır. [68] de bu problemin çözümü için iki aşamalı bir algoritma önerilmiştir. Ayrıca [69, 70] çalışmaları socha benchmark problemin üzerinde test edilmiştir. [71]’de önerilen yöntem ICT-2002 benchmark problemi üzerinde test

edilmiştir. [72]'de olan yöntem bir melez genetik algoritmadan yararlanarak, ITC-2007'inin üçüncü bölümünün çözümü için uygulanmıştır.

Bu problemin çözümü için diğer popülasyon tabanlı yöntemlerden birisi karınca koloni algoritmasıdır (Ant Colony Optimisation - ACO) [73]. Bu çalışmalarda [21, 74] karınca koloni algoritması kullanılmıştır. Başka popülasyon tabanlı yöntemlerden birisi memetik algoritmadır (memetic algorithm) [75]. Ayrıca [76]'da bu yöntem kullanılmıştır. Yeditepe üniversitenin mühendislik ve mimarlık fakültesinin ders çizelgelemesinin oluşturulması için [77]'de bu yöntem kullanmış ve iyi sonuçlar elde edilmiştir.

1.4.4.2.2. Tek Bir Çözüm Tabanlı Meta Sezgisel Yaklaşımlar

Bu yöntemler isminden belli olduğu gibi optimum çözümü bulması için tek bir başlangıç çözümünden yola çıkar. Bundan dolayı bu yöntemlere tek nokta (single point) da denilir. Bu yöntemin farkı diğer popülasyon tabanlı yöntemlere bütün optimum arama sürecinin başından sonuna kadar sadece bir çözüm üzerinden yapılır. Bu yöntem bütün arama uzayının aranabilmesi için farklı komşu arama mekanizmalarından yararlanır ve yerel optimumda saklanıp kalmasını önler. Bu yöntemlerden birisi tabu arama [78] tekniğidir (Tabu search). Buradaki [79, 80, 81, 82] çalışmalarda ders çizelgeleme probleminin çözümü için bu yöntemden yararlanılmıştır. ITC-2007 yarışmasının üçüncü bölümünde ikinci olan kişi çalışmasında [83] bu yöntemin bir melez halini kullanmıştır.

Diğer bir yöntem ise değişken komşu arama (Variable Neighbourhood Search-VNS) [84] yöntemidir. Salwani Abdullah ve diğerleri [85] ders çizelgeleme probleminin çözümü için bu yöntemi kullanmışlardır.

Tavlama benzetme diğer bir (Simulated Annealing) [86] tek çözüm tabanlı meta sezgiseldir ve ilk olarak Davis ve Ritter [87] tarafından zaman çizelgeleme probleminin çözümü için ele alınmıştır. Daha sonra [88, 89] çalışmalarında bu yöntemi ders çizelgeleme probleminin çözümü için kullanmışlardır.

Great Deluge [90] algoritması bu sınıftaki diğer bir yöntemdir. Bu yöntemle ders çizelgeleme probleminin çözümü için [91] çalışmasında ele alınmıştır. Bu yöntem benzer şekilde [92] ITC-2002 problemi üzerinde de kullanılmıştır. Problemin çözümü için yapılan çalışmalar sonucunda Great Deluge yönteminin tavlama benzetme yönteminden daha iyi olduğu görülmüştür. [92, 93] ITC-2002 ve Socha örnek problemleri üzerinde yapılan çalışmalardandır.

1.4.5. Melez Evrimsel ve Meta Sezgisel Algoritmalar

Ders çizelgeleme probleminin çözümü için literatürde bulunan çalışmaların çoğu evrimsel veya meta sezgisel yöntemlerin melez haliyle yapılmaktadır. Örneğin memetik algoritma (memetic algorithm), genetik algoritma ve yerel arama yöntemleri birleşimidir. Bu yöntem Alkan ve Özcan [77] tarafından problemin çözümü için ele alınmıştır. Yapılan diğer çalışmalar; ITC-2002 yarışmasında birinci olan Kostuch [64], Chiaradini ve diğer [94] çalışmasında melez yöntemini ITC-2002 ve socha örnek probleminin çözümünde kullanmışlardır. Abdullah ve diğerleri [63, 72] ITC-2007 örnek problemi üçüncü bölümünün için melez yöntemini önermişler.

1.4.6. Çok Ölçütlü Yaklaşım

Bu yaklaşımlarda cezaların değeri (ağırlığı) kısıdına karşı ölçülandırılmaktadır. Kısıtları farklı enstitü ve durumlarda farklı seviyede öneme sahiptirler. Bu yaklaşımlar Great Deluge algoritmasını kullanarak kısıtların ceza ağırlık değerlerinin dinamik değişimiyle genişletilmiştir [95].

1.4.7. Durum-Tabanlı Düşünme Yaklaşımları

Bunlar da diğer zaman çizelgeleme problemlerinin çözümü için kullanılan yöntemlerden, yeni zaman çizelgelemenin çözümü için uygulanan yaklaşımın bölümlerinde yararlanırlar. Ayrıntılı bilgiler [26]'de verilmektedir.

1.4.8. Üst sezgisel ve Uyarlamalı Yaklaşımlar

Üst sezgiselin bir basit tanıtımı “ bir sezgisel seçimi için sezgisel” olabilir; başka bir ifade ile aramanın her adımında sezgisel kullanarak alt sezgiseller kümesinden birisini seçilmesi. Sezgiseller seçilmek için yarışmaya girebilirler. Bu yarışma algoritmanın önceden öğrenilen bilgiler üzerinden yapılabilir. Üniversite ders çizelgeleme probleminin çözümü için [83] bu yöntem kullanılmıştır.

Meta sezgisel yöntemlerden birisi sürü zekâsıdır (swarm intelligence) [96]. Bu yöntemler doğada olan sürüce yaşayan hayvanlar ve böceklerin zeki davranışlarından

gerçek dünyada olan diğer problemlerin çözümü için yararlanılır. Bu tür meta sezgiseller ayrıntılı şekilde alt bölümde verilmektedir.

1.5. Sürü Zekâsı ve Temelleri

Uzun yıllar önce doğada var olan müthiş düzenler insanların ilgisini çekmiştir. Bunlardan birisi sürü halinde yaşayan hayvanlar, böcekler ve kuşların sürü halinde yaptıkları ilginç ve düzenli davranışlardır. Örneğin kuşların sürü halinde uçarken farklı hareketler yapması, karıncaların yem bulması, bilim adamlarının ve araştırmacıların ilgisini çekerek çok sayıda araştırmanın yapılmasını sağlamıştır. Tüm bu davranışlara sürü davranışları denilir.

Son yıllarda, bilgisayar ve biyoloji biliminde yapılan araştırmacılar bu sürülerin birbiriyle nasıl iletişim kurup amaçlarına nasıl ulaştıklarını anlamak ve bu sürülerin matematiksel bir modelini oluşturmak için ortak çalışmalar yapılmaktadır.

Başlangıçta araştırmacılar her bir böceğin veya kuşun kendi sürüsündeki küresel yapıyı bildiğini ve bu bilgiler üzerinden hareket ettiğini düşündüler. Örneğin bir arı kolonisinin yönetiminde kraliçe arının başrolde olduğu sanıldı. Yani bütün koloniye ait bilgilerin kraliçe arı tarafından toplanıp sonra bu bilgiler ışığında işçilerin koloni işlerini yapmak için yönlendirildiği sanıldı. Daha sonra yapılan araştırmalar ise bu varsayımların tersini gösterdi. Bu araştırmalara göre bir böceğin küresel bir bilgiye sahip olmadığı, tüm bilgileri toplayarak onları kontrol edilebilecek bir durumda olmadığı ve böceklerin birbirinden bağımsız olarak iş gördüğü tespit edilmiştir. Yani merkezi bir yönetim olmadan bir kolonide işler yapılır. Her birey sadece bazı basit kuralları kullanarak komşuluk bazındaki bilgilere göre (yerel ve bölgesel olmayan) bir olasılıklı uyarıcı tepki davranışlarını gösterir. İletişim sayesinde koloni seviyesinde bir organizasyonun oluşumu sağlanır. Bireyler arasındaki karşılıklı etkileşimler bilgilerin bütün koloni arasında yayılmasını sağlar ve her bireyin eylemlerini kontrol eder [99].

Küresel bilgilere sahip olmadan sadece bölgesel bilgilere göre kendilerine ait hareket kurallarını kullanarak kendi kendine organize olabilen bireyler arasındaki etkileşim sürü zekâsı (Swarm Intelligence) adı verilen kollektif zekânın ortaya çıkmasını sağlamıştır. Sürü zekâsı, sürünün düzenli ve akıllı davranışlarını Şekil 5'de görüldüğü gibi, kaynaklardan ve çevresinden verimli bir şekilde yararlanmasını sağlar [100].



Şekil 5. Sürü zekâsı

Bonabeau 1999 de [96] sürü zekâsı (Swarm Intelligence) üzerine resmi bir tanıtım yaptı.” Doğadaki hayvan sürülerinin veya böcek kolonilerinin kollektif davranışlarından ilham alınarak problemlerin dağıtılmış çözümü için herhangi algoritma veya cihaz tasarlaması ” tanıtımı bundan sonraki araştırmacılara sürü zekâsı için resmi tanıtım olarak alındı.

Bonabeau sürü sistemleri için iki temel özellik belirtti. Bu özelliklere göre her matematiksel modelin sürü zekâsı davranışını temsil etmek için bu özelliklere sahip olmasını gerekli ve yeterli saydı. Bu özellikler:

1.5.1. Kendi Kendine Organize Olabilme

Bu özellik sayesinde alt-seviyedeki bölgesel etkileşimlerin küresel bilgi kullanmadan üst-seviyedeki küresel cevabı oluşturmaktadır. Yani tamamen yerel bilgiler çerçevesinde olur ve küresel bilgi kullanılmadan gerçekleşir. Böylece hiç bir merkezi yönlendirme olmadan bütün grup eşgüdümle hareket edebilir. Örneğin, bal arılarının kovan yapmalarında, karıncaların yol bulma yöntemlerinde ve termitlerin yuvalarını inşa etmesinde bu özellikten yararlanarak bir ortak karar ifade edilir.

Bonabeau bir sistemin kendi kendine organize özelliğine sahip olması için dört temel özelliğe sahip olmasının gerekli olduğunu söylemiştir. Bunlar;

1.5.1.1. Pozitif Geri Besleme

Uygun ve iyi durumların oluşmasını destekleyen basit ve alt-seviyede (bölgesel) davranışsal kuralların katkısının sonucudur. Örneğin, bal arı kolonisinde arıların dans etmesi, karıncaların yem bulma yöntemlerinde yiyecek kaynağını gösteren yönde kimyasal madde salgılaması ve bu izi güçlendirmek için diğerlerini bu işe yönlendirmesi bir pozitif geri besleme sayılabilir. Diğer bir deyişle, yapılan bir işin tekrar yapılması için gerekli koşulları oluşturmak ve diğerlerine iş seçimi için bir ortam hazırlamaktır.

1.5.1.2. Negatif Geri Besleme

Pozitif geri beslemenin ve ortakça oluşan örüntünün dengelenmesini sağlar. Negatif geri besleme mekanizmasına belli bir iş konusunda yiyeceklerin tüketiminde doyuma gitmeyi önlemek için ihtiyaç duyulur. Örneğin karıncaların yem bulmasında kullanılır.

Kimyasal maddenin buharlaşması negatif geri beslemeye neden olur. Örneğin bal arılarının bulunduğu yiyecek kaynağı eğer tükenme noktasına gelmişse arılar tarafından bırakılması bir negatif geri beslemedir.

1.5.1.3. Dalgalanmalar

Rastgele keşifler veya hatlar üzerindeki bireylerin işini değiştirmek yeni çözümlerin keşfini sağladığından yaratıcılık ve yenilik açısından önem taşımaktadır. Örneğin bal arı kolonisinde kâşif arılar yeni kaynak bulmak için rastgele hareketler yapar, karıncalar yem bulurken bazı kâşif karıncaların kaybolması yeni yiyecek kaynaklarının keşfine veya önceliklere daha iyi yol bulmasına yol açar. Rastgelelik kendi kendine organize olması için kritik bir rol oynamaktadır; yeni çözümlerin bulunmasına olanak sağlar.

1.5.1.4. Çoklu Etkileşimler

Sürüde olan bireylerin başkalarıyla görüşebilmesi ve bilgi paylaşımını sağlar. Böylelikle bilgilerin bütün sürüye dağılması sağlanır. Örneğin bal arı kolonisinde arılar koloni çevresinde olan yiyecek kaynaklarının bilgilerini dans salonunda birbirleriyle paylaşırlar.

1.5.2. İş Bölümü

Bir sürü içerisinde farklı işler olur. Bunlar eşzamanda uzman bireyler tarafından yapılır. Bu tür olaylara iş bölümü denir. Bu özel bireylerin eşzamanda sırayla yaptığı işler, uzman olmayan bireylerin yaptığı işlere göre daha verimlidir.

Sürü içerisinde işlerin bölünmesi, sürünün dış ve iç değişimlere karşı esnekliğini sağlar.

1.6. Doğada Arılar

Doğada, iç ve dış değişimlere kolektif zekâsından yararlanarak tepki verebilen ve dinamik bir iş paylaşımı olan en iyi böcek sürülerden birisi bal arıları kolonisidir. Bu koloni yiyecek bulma, bal üretme ve saklama, iletişim kurma gibi işleri de en iyi şekilde yerine getirebilir.

Kolonideki arılar üç çeşitten oluşurlar:

- Kraliçe Arı (Queen Bee): Her kolonide sadece bir kraliçe arı bulunur. Görevi yumurtalamaktır. Yani kolonide olan bütün arıların annesi sayılır. Kolonide yiyecek azalınca yumurtlamaya başlar ve koloni yeterince kalabalıklaştığı zaman yumurtlamaya ara verir.

- Erkek Arılar (Drones) :Kolonide sadece yaz aylarında olurlar. En temel görevleri kraliçe arının döllenmesidir. Bir başka deyişle kolonide olan arıların babası sayılırlar ve çiftleştikten sonra ölürlar.

- İşçi Arılar (Worker): Koloninin yumurtalama hariç her işinden sorumludurlar. Bal toplanması, toplanan balın saklanması, koloninin havalandırılması, ölü arıların koloniden atılması, erkek arı, larvalar ve kraliçe arının beslemesi ve koloninin güvendiğinden sorumludurlar [100].

1.7. Gerçek Arıların Yiyecek Arama Davranışları

Doğadaki arı kolonileri merkezi bir yönetim olmadan bir iş paylaşımına sahiptirler. Bu işler özelleşmiş arılar tarafından yapılmaktadır. Bu iş dağılımını gerçekleştirdikleri için kendi kendine organize olabilmektedirler. Sürü zekâsının en önemli iki özelliği iş paylaşımının paralel şekilde yapılması ve kendi kendine organize olabilmemesidir. Tereshko, V. ve Loengarov [101], 2000 yılında arıların yiyecek arama davranışlarını gözlemleyerek ve bir minimal hali ele alıp sürü zekâsının ortaya çıkmasını sağlayan bir model geliştirdi. Bu model üç bileşenden oluşur: yiyecek kaynakları, görevli işçi arılar, görevsiz işçi arılar. Derviş Karaboğa [3] bu bileşenleri böyle açıklamıştır:

1.7.1. Yiyecek Kaynakları

Bir kolonideki arıların yiyecek bulma sürecinde, polen, bal ve nektar elde etmek için buldukları kaynaklardır. Bu kaynakların çeşitleri ve değerleri birçok faktöre bağlıdır. Kovandan mesafesi, nektar zenginliği ve çıkarma kolaylığı örnek olarak verilebilir. Fakat basitlik açısından kaynakların zenginliği tek bir değerle alınabilir.

1.7.2. Görevli İşçi Arılar

Adlarından da belli olduğu gibi belli bir işi yapabilmek için görevlendirilmişlerdir. Önceden keşfedilmiş belli kaynaklara ait nektarların kovana taşınmasından sorumludurlar ve aynı zamanda kovanda bekleyen diğer arılarla, geldikleri kaynakların nektarını ve diğer önemli bilgileri paylaşırlar.

1.7.3. Görevsiz İşçi Arılar

Bu arılar kaynaklardan nektar toplamak için değil nektar kaynaklarının aranması ve kaynakların bulunmasından, buldukları kaynak bilgilerinin diğer arılarla paylaşımından sorumludurlar. Kovanda iki tür görevsiz işçi arı vardır; bunlardan biri içsel ve dışsal etmene bağlı olarak rastgele kaynak arayışında olan kâşif arılar, diğeri ise kovanda görevli işçi arıların kaynaklardan dönmesini bekleyen, onları izleyen ve geldikleri kaynaklarla ilgili paylaştıkları bilgilerden yararlanıp, bu bilgiler sayesinde kaynakların içerisinde yeni bir kaynağı seçip, o kaynağa yönelen gözcü arılardır. Kâşif arıların sayısı, ortalama olarak

kolonideki tüm arıların % 5-10 oranı kadardır. Bir sürü zekâsının oluşumunda bilgi paylaşımı önemli bir rol oynamaktadır. Bu bilgi paylaşımı bir kolektif bilginin oluşumuna neden olur. Arı kolonisinde de diğer sürüler gibi bilgi paylaşma özeliğini gerçekleştirmek için her kovanda bir dans alanı bulunmaktadır. Bu dans bölgesinde arılar, dans olarak adlandırılan hareketlerle birbirlerinin antenlerine dokunarak yiyecek kaynakların kalitesi ve yeriyle ilgili bilgileri paylaşırlar. Gördükleri kaynaklara daha fazla arıyı ikna edip onlara yönlendirmek için kovanda olan diğer yerlerde de dansı gerçekleştirirler ve kaynağına geri dönerler. Yaptıkları danslar yiyecek kaynakların özelliklerine ve taşınan bilgiye göre çeşitli olabilir. İyi danslar kaynağın kaliteli olduğunu gösterir. Bu kalite birçok etmene bağlıdır. Örneğin günün hangi saati olduğu, çiçeğin nektarının kıvamı, nektarın çıkarılma kolaylığı, hava koşulları, en önemlisi ise nektarın tatlılık oranı ve çiçeğin kovana olan mesafesidir. Kaynakların kovana olan mesafesine bağlı olarak arılar tarafından yapılan danslar değişmektedir. Örneğin; dairesel dans (round dance), kuyruk dansı (waggle dance) ve titreme dansı (tremble dance).

- Dairesel dans (round dance) Eğer yiyecek kaynağının kovana olan uzaklığı maksimum 50-100 metre arasında ise arılar bu bilgileri dairesel dans yaparak ifade ederler.

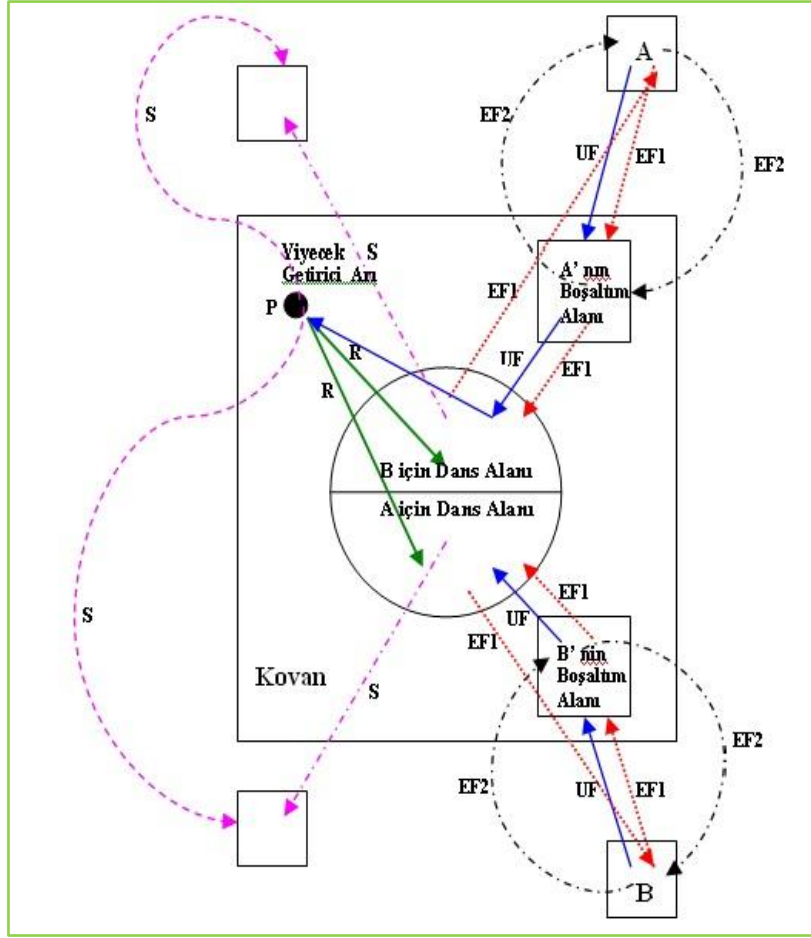
- Titreme dansı (tremble dance): Arı zengin bir nektar kaynağı bulup kovana geldiğinde, kovana getirilen nektar miktarı fazla ve nektar işleyen arıların sayısı getirilen nektarı işlemeye yetmiyorsa bu arı görevini değiştirip nektar işleme işini yapan arı görevine geçer. Bu geçmeyi titreme dansı şeklinde gösterir.

- Kuyruk dansı (waggle dance) : Eğer yiyecek kaynağının kovana olan uzaklığı 100 metreden 10 kilometreye kadar ise bulunan kaynağın bilgisi kuyruk dans kullanarak aktarılmaktadır.

Bu konular için detaylı bilgiler Derviş Karaboğa [3] ve Akay [100] kaynaklarında bulunmaktadır.

Gözcü arılar, dans alanında yiyecek kaynaklarından dönen arıların danslarını iyice izledikten sonra bir kaynağın seçilmesine karar verip o kaynağa yönelirler. Zengin kaynakların bilgisinin aktarılması için fazla dans yapıldığından bu kaynakların gözcü arılar tarafından seçilme olasılığı yükselir.

Derviş Karaboğa [3] ve Akay [100] arıların yiyecek arama davranışlarının daha iyi anlaşılabilmesi için Şekil 6 diyagramını önermişlerdir.



Şekil 6. Yiyecek arama çevrimi [100].

A, B keşfedilmiş iki yiyecek kaynağı olarak varsayılmaktadır. Arama sürecinin başlangıcında potansiyel bir yiyecek arayıcı, görevi belirsiz bir arı olarak kovanın etrafındaki kaynaklar hakkında hiçbir bilgisi olmadan araştırmaya başlamaktadır.

Bu durumdaki bir arı için iki olası seçenek söz konusudur:

- 1) Bu arı kâşif bir arı olabilir ve içsel, dışsal etkenler sayesinde rastgele yiyecek aramaya başlayabilir (şekil 6'da S ile gösterilmektedir).
- 2) Bu arı bir gözcü arı olabilir, dansları izledikten sonra bu dansların anlattığı kaynağa gidebilir (şekil 6'da R ile gösterilmektedir).

Görevi belirsiz arı kaynağı keşfettikten sonra kaynağın yerini hafızasına alır ve hemen nektar toplamaya başlar. Nektarın zenginlik durumunun ölçülmesi ve kaynak bilgisinin kovana aktarılmasından sorumlu olur.

Bir işçi arı kaynaktan nektarı alıp kovana döner. Kovanda yiyecek depolayıcı arılara topladığı nektarı teslim ettikten sonra bu arı için üç seçenek ortaya çıkmaktadır:

- 1) Önceki gittiği kaynağa bağımsız kalarak o kaynağı bırakıp bir başka kaynağın seçimi için bağımsız bir izleyiciye döner.(Şekil 6'da UF ile gösterilmektedir)
- 2) Kovandaki diğer arıları, önceki gittiği kaynağa yönlendirmek için kaynağa dönmeden önce dans eder. (Şekil 6'da EF1 ile gösterilmektedir)
- 3) Sadece kendi işini yapmakla meşgul olup ve diğer arıları yönlendirmeden önceki kaynağına gidebilir. (Şekil 6'da EF2 ile gösterilmektedir)

Daha önce belirtildiği gibi sürü zekâsının oluşmasında en önemli özelliklerden birisi sürünün kendi kendine organize olmasıdır. Bunun oluşmasının nedeni dört tane temel özeliğin bulunmasından kaynaklanır, bal arıların yiyecek aramada kendi kendine organize olması özeliği de aşağıda anlatılan temel özelliklerin bulunmasından kaynaklanır:

- a. Pozitif geri besleme: Bir kaynağın nektar miktarı artınca o kaynağına ait arıların bilgi aktarma için dans sayısı artıyor ve bundan dolayı gözcü arıların bu kaynağı seçme olasılığı da artıyor.
- b. Negatif geri besleme: Eğer bir kaynağın nektarı tükenirse bu kaynağına ait arılar kaynağı terk ederler.
- c. Salınımlar (dalgalanmalar): Kâşif arıların yiyecek kaynağı bulabilmesi için rastgele aramalar yapmasında bu özellik bulunmaktadır.
- d. Çoklu etkileşim: Kovanda olan dans alanında arıların yiyecek kaynakları ile ilgili bilgileri dans ederek paylaşmasında bu özellik bulunmaktadır.

1.8. Yapay Arı Koloni Algoritması

Doğada var olan zeki davranışlar içeren süreçlerin incelemesi araştırmacıları yeni optimizasyon metotları geliştirmeye sevk etmiştir.

Bal arılarının yiyecek arama yöntemleri, zekâlı davranışlardan biri sayılmaktadır ve tam bir sürü zekâsını temsil ediyor. 2005 yılında Tereshko ve Loengarov [101] bu davranış için bir model önerdiler. Derviş Karaboğa 2005 de [97] bu model üzerinde bazı değişiklikler yaptıktan sonra Yapay Arı Kolonisi (Artificial Bee Colony, ABC) adlı bir yeni optimizasyon algoritmasını geliştirmiştir. Bu algoritmanın temel aldığı modelde bazı kabuller yapılmaktadır. Bunlardan birincisi her bir kaynağın nektarının sadece bir görevli arı tarafından alınıyor olmasıdır. Yani görevli arıların sayısı aynı zamanda gözcü arıların sayısına da eşittir. Nektarı tükenmiş kaynağın görevli arısı artık kâşif arı haline dönüşmektedir. Yiyecek kaynaklarının yerleri optimizasyon problemine ait olası

çözümlerin kalitesine karşılık gelmektedir. Dolayısıyla ABC optimizasyon algoritması en fazla nektara sahip kaynağın yerini bulmaya çalışarak uzaydaki çözümlerden problemin minimumunu yada maksimumunu veren noktayı (çözüm) bulmaya çalışmaktadır. Bu modele ait süreç adımları aşağıdaki gibi verilebilir:

1. Yiyecek arama sürecinin başlangıcında, kâşif arılar çevrede rastgele yiyecek aramaya başlarlar.
2. Yiyecek kaynakları bulduktan sonra, kâşif arılar artık görevli arı olurlar ve buldukları kaynaklardan kovana nektarı boşatır ve bu noktadan sonra ya bulduğu kaynağa geri döner yada kaynakla ilgili dans alanında sergilediği dans aracılığıyla kovanda bekleyen gözcü arılara iletir. Eğer faydalandığı kaynak tükenmiş ise görevli arı kâşif arı haline gelir ve yeni kaynak arayışına yönelir.
3. Kovanda bekleyen gözcü arılar zengin kaynakları işaret eden dansları izlerler ve yiyeceğin kalitesi ile orantılı olan dans frekansına bağlı olarak bir kaynağı tercih ederler.

ABC algoritmasının bu süreçleri ve temel adımları şekil 7’de verilmektedir.

Adım 1. Başlangıç yiyecek kaynağı bölgelerinin (üretilemesi *ialization Phase*).

REPEAT

Adım 2. İşçi arıların yiyecek kaynağı bölgelerine gönderilmesi (*Employed Phase*).

Adım 3. Olasılıksal seleksiyonda kullanılacak olasılık değerlerinin görevli arılardan gelen bilgiye göre hesaplanması.

Adım 4. Gözcü arıların olasılık değerlerine göre yiyecek kaynağı bölgesi seçmeleri.

Adım 5. Kaynağı bırakılma kriteri: limit ve kâşif arı üretimi.

UNTIL Çevrim sayısı = Maksimum çevrim sayısı)

Şekil 7. ABC algoritmasının temel adımları [100].

Yiyecek arayan arılarda görülen zeki davranış ile bu davranışı simüle eden ABC algoritmasının temel birimleri aşağıdaki bölümlerde açıklanmaktadır [100].

1.8.1. Başlangıç Yiyecek Kaynağı Bölgelerinin Üretilmesi

Arama uzayını yiyecek kaynaklarını içeren kovan çevresi olarak düşünürsek, algoritma arama uzayındaki çözümlere karşılık gelen rastgele yiyecek kaynağı yerleri üretmek çalışmaya başlamaktadır. Rastgele yer üretme süreci her bir parametrenin alt ve üst sınırları arasında rastgele değer üretmek gerçekleşir (5).

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \quad (5)$$

Burada $i = 1 \dots SN, j = 1 \dots D$ ve SN yiyecek kaynağı sayısı ve D ise optimize edilecek parametre sayısıdır. x_j^{min} , j . parametrenin alt sınırı, x_j^{max} ise j . parametrenin üst sınırıdır. Aynı zamanda başlangıç aşamasında her kaynağın geliştirilememe sayısını ifade eden $failure_i$ (i . Kaynağın geliştirilememe sayısı) sayaçları da sıfırlanmaktadır.

Başlangıç aşamasından sonra yiyecek kaynaklarının görevli arı, gözcü arı ve kaşif arı süreçlerinden geçirilerek, daha iyisi bulunmaya çalışılır. ABC algoritması için durdurma kriteri olarak maksimum çevrim sayısı (MCN) ve kabul edilebilir bir hata değeri (ϵ) veya diğer optimizasyon algoritmaları için kullanılan standart bir durdurma kriteri tanımlanabilir.

1.8.2. Görevli Arıların Yiyecek Kaynağı Bölgelerine Gönderilmesi

Daha önce de belirtildiği gibi her bir kaynağın bir görevli arısı vardır. Dolayısıyla yiyecek kaynaklarının sayısı görevli arıların sayısına eşittir. İşçi arı çalıştığı yiyecek kaynağı komşuluğunda yeni bir yiyecek kaynağı belirler ve bunun kalitesini değerlendirir. Yeni kaynak daha iyi ise bu yeni kaynağı hafızasına alır. Yeni kaynağın mevcut kaynak komşuluğunda belirlenmesinin benzetimi (6) ile tanımlanmaktadır.

$$v_{ij} = x_{ij} + \emptyset_{ij}(x_{ij} - x_{kj}) \quad (6)$$

x_i ile gösterilen her bir kaynak için bu kaynağın yani çözümün tek bir parametresi (rastgele seçilen parametresi, j) değiştirilerek x_i komşuluğunda v_i kaynağı bulunur. (6) de j , $[1, D]$ aralığında rastgele üretilen tamsayıdır. Rastgele seçilen j parametresi

değiştirilirken, yine rastgele seçilen x_k komşu çözümünün ($k \in \{1,2, \dots, SN\}$) j . Parametresi ile mevcut kaynağın j . parametresinin farkları alınıp $[-1,1]$ arasında rastgele değer alan \emptyset_{ij} sayısı ile ağırlıklandırıldıktan sonra mevcut kaynağın j . parametresine eklenmektedir.

(7) den de görüldüğü gibi $x_{i,j}$ ve $x_{k,j}$ arasındaki fark azaldıkça, yani çözümler birbirine benzedikçe, $x_{i,j}$ parametresindeki değişim miktarı da azalacaktır. Böylece bölgesel optimal çözüme yaklaştıkça değişim miktarı da adaptif olarak azalmaktadır.

Bu işlem sonucunda üretilen $v_{i,j}$ 'nin daha önceden belli olan parametre sınırlarını aşması durumunda j . parametreye ait olan alt veya üst sınır değerlerine ötelenmektedir (8).

$$v_{ij} = \begin{cases} x_j^{min} & v_{ij} < x_j^{min} \\ v_{ij} & x_j^{min} \leq x_j^{max} \leq x_j^{max} \\ x_j^{max} & v_{ij} > x_j^{max} \end{cases} \quad (8)$$

Sınırlar dâhilinde üretilen v_i parametre vektörü yeni bir kaynağı temsil eder ve bunun kalitesi hesaplanarak uygun bir değere atanır (9).

$$fitness_i = \begin{cases} \frac{1}{(1 + f_i)}, & f_i \geq 0 \\ 1 + abs(f_i), & f_i < 0 \end{cases} \quad (9)$$

Burada f_i , v_i kaynağının yani çözümün maliyet değeridir. x_i ile v_i arasında nektar miktarlarına yani uygunluk değerlerine göre bir aç gözlü (greedy) seçme işlemi uygulanır. Yani bulunan v_i çözümü daha iyi ise görevli arı hafızasından eski kaynağın yerini silerek v_i kaynağının yerini hafızaya alır. Aksi takdirde görevli arı x_i kaynağına gitmeye devam eder ve x_i çözümü geliştiremediği için x_i kaynağı ile ilgili geliştirememe sayacı ($failure_i$) bir artar, geliştirdiği durumda ise sayaç sıfırlanır.

1.8.3. Görevli Arıların Seleksiyonda Kullanacakları Olasılık Değerlerinin Hesaplanması (Dans Benzetme)

Tüm görevli arılar bir çevrimde araştırmalarını tamamladıktan sonra kovana dönüp buldukları kaynak nektar miktarları ile ilgili gözcü arılara bilgi aktarırlar. Bir gözcü arı dans aracılığıyla paylaşılan bilgiden faydalanarak yiyecek kaynaklarının nektar miktarları ile orantılı bir olasılıkla bir bölge (kaynak) seçer. Bu ABC'nin çoklu etkileşim sergilediğinin bir örneğidir. Olasılıksal seçme işlemi, algoritmada nektar miktarlarına karşılık gelen uygunluk değeri kullanılarak yapılmaktadır.

Uygunluk değerine bağlı seçme işlemi rulet tekerleği, sıralamaya dayalı, stokastik örnekleme, turnuva yöntemi yada diğer seleksiyon şemalarından herhangi biri ile gerçekleştirilebilir. Temel ABC algoritmasında bu seleksiyon işlemi rulet tekerleği kullanılarak yapılmıştır. Tekerlekteki her bir dilimin açısı uygunluk değeri ile orantılıdır. Yani bir kaynağın diğer kaynaklara göre nisbi seçilme olasılığı olduğunu (10) verilmektedir.

$$p_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j} \quad (10)$$

Burada $fitness_i$, i . kaynağın kalitesini, SN görevli arı sayısını göstermektedir. Bu olasılık hesaplama işlemine göre bir kaynağın nektar miktarı arttıkça (uygunluk değeri arttıkça) bu kaynak bölgesini seçecek gözcü arı sayısı da artacaktır. Bu özellik ABC'nin pozitif geribesleme özelliğine karşılık gelmektedir.

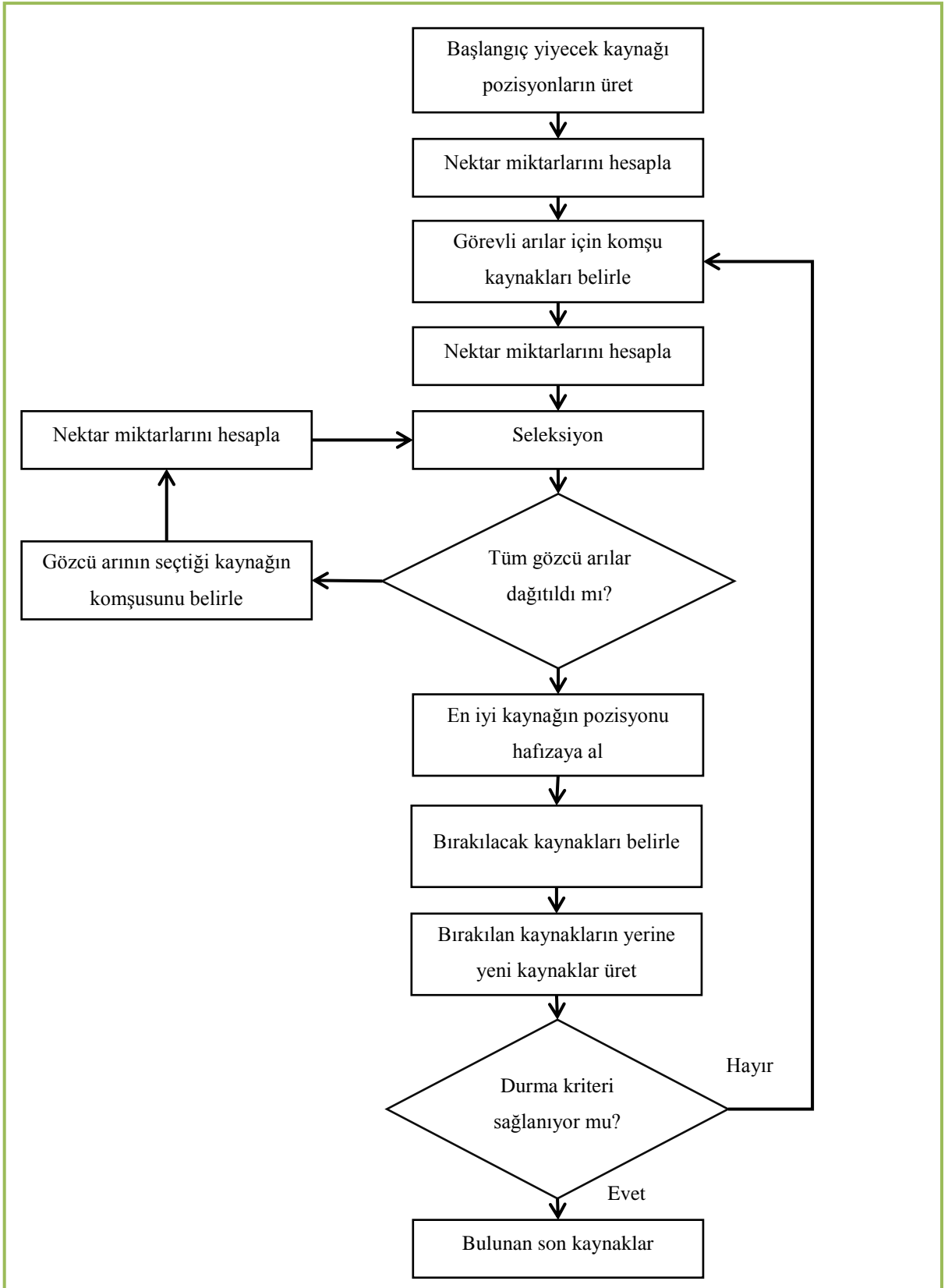
1.8.4. Gözcü Arıların Yiyecek Kaynağı Bölgelerini Seçmeleri

Algoritmada olasılık değerleri hesaplandıktan sonra bu değerler kullanılarak rulet tekerleğine göre seçim işleminde her bir kaynak için $[0,1]$ aralığında rastgele sayı üretilir ve p_i değeri bu üretilen sayıdan büyükse görevli arılar gibi gözcü arı da (6)'yı kullanarak bu kaynak bölgesinde yeni bir çözüm üretir. Yani çözüm daha iyi ise eski çözüm yerine bu çözüm alınır ve çözüm geliştirememeye sayacı ($failure_i$) sıfırlanır. Eski çözümün uygunluğu daha iyi ise bu çözüm muhafaza edilir ve geliştirememeye sayacı ($failure_i$) bir artırılır. Bu süreç tüm gözcü arılar yiyecek kaynağı bölgelerine dağılına kadar devam eder.

1.8.5. Kaynağı Bırakma Kriteri: Limit ve Kaşif Arı Üretimi

Bir çevrim sonunda tüm görevli ve gözcü arılar arama süreçlerini tamamladıktan sonra çözüm geliştiremem sayaçları (*failure_i*) kontrol edilir. Bir arının bir kaynaktan faydalanıp faydalanmadığı, yani gidip gidği kaynağın nektarının tükenip tükenmediği çözüm geliştirememe sayaçları aracılığıyla bilinir. Bir kaynak için çözüm geliştirememe sayacı belli bir eşik değerinin üzerindeyse, artık bu kaynağın görevli arısının tükenmiş olan o çözümü bırakıp kendisi için başka bir çözüm araması gerekmektedir. Bu da biten kaynakla ilişkili olan görevli arının kâşif arı olması anlamına gelmektedir. Kaşif arı haline geldikten sonra, bu arı için rastgele çözüm arama süreci başlar (5). Kaynağın tükendiğinin belirlenmesi için kullanılan eşik değeri, ABC algoritmasının önemli bir kontrol parametresidir ve “*limit*” olarak adlandırılmaktadır. Temel ABC algoritmasında her çevrimde sadece bir kaşif arının çıkmasına izin verilir.

Tüm bu bilimler arasındaki ilişki ve döngü Şekil 8’deki bir akış diyagramı ile şematize edilebilir.



Şekil 8. ABC Algoritmasının akış diyagramı [3, 100].

1.8.6. ABC'nin Temel Özellikleri

Sağlam bir arama sürecinde sömürü (exploitation) ve keşif (exploration) süreçleri birlikte yapılmalıdır. ABC algoritmasında görevli ve gözcü arılar tarafından sömürü ve kaşif arılar tarafından keşif süreci her çevrim içerisinde birlikte yapılması sağlanır [102].

ABC algoritmasında 4 farklı seleksiyon metodunun bir arada kullanılması sayesinde hem küresel ve hem de iyi bir bölgesel araştırmaya dönüşmüştür. Yapılan seleksiyon işlemleri alt da anlatılmaktadır [100]:

1. Tüm kaynaklar içerisinde en iyi kaynağı belirleme yönünde eşitlik 10 kullanıp olasılık değerlerini hesaplamak bir küresel seleksiyon işlemidir.
2. Gözcü ve görevli arılar tarafından eşitlik 6 kullanarak ve kaynak için bir komşu kaynağı üretme bir bölgesel seleksiyon işlemidir.
3. İşçi ve gözcü arılar tarafından her kaynağa bir komşu ürettikten sonra bu kaynak ve komşu arasında iyi olmaya karar veren aç gözlü seleksiyon işlemidir.
4. Kaşif arılar tarafından rastgele yiyecek kaynağı üretimi için kullanılan 5 eşitliği bir rastgele seleksiyon işlemidir.

ABC algoritmasında olan diğer temel özellikler:

- Doğada olan bal arılarının yiyecek arama davranışlarından ilham alarak bu davranışı simule eder.
- Diğer optimizasyon algoritmaların göre az kontrol parametresine sahiptir (maksimum çevrim sayısı, kovanda olan toplam arı sayısı, geliştirememeye için bir limit değeri).
- Oldukça basittir.
- İlkte nümerik problemlerde geliştirilmesine rağmen ayrık problemler için de kullanılabilir.
- Bir sürü zekâsını temsil eden algoritmadır.
- Bölgesel ve küresel araştırmayı dengeli şekilde yürütmektedir.

2. YAPILAN ÇALIŞMALAR

2.1. Giriş

Bölüm 1.2’de açıklandığı gibi zaman çizelgeleme problemi “Kısıtları göz önünde bulundurarak mümkün olduğunca istenileni karşılayabilecek bir şekilde, verilen kaynakların belirli zaman aralıklarında paylaşılmasıdır” şeklinde tanımlanmıştır [6]. Kısıtlar katı ve esnek olmak üzere iki sınıfa ayrılmaktadır. Katı kısıtların sağlanması zorunludur Esnek kısıtların sağlanması zorunlu değildir fakat daha fazla sağlanması çözümün kalitesini artırır. Zaman çizelgeleme probleminin spor, tren, otobüs, uçak ve eğitimsel gibi çok farklı türleri vardır ve gittikçe artan istek ve kısıtlardan dolayı birer NP probleme dönüşmüşlerdir.

Bölüm 1.2.2 de anlatıldığı gibi zaman çizelgeleme problemlerinin bir çeşidi eğitimsel çizelgelemedir. Eğitimsel zaman çizelgeleme araştırmacılar tarafından, okul ders çizelgeleme, üniversite ders çizelgeleme ve sınav çizelgeleme olmak üzere üç dala ayrılmış [12] ve Bölüm 1.2.2 de ayrıntılı bir şekilde açıklanmış ve farkları belirlenmiştir.

Üniversite ders çizelgeleme probleminin bir formal tanımı belirli sayıda faaliyetleri (ders oturumları) sınırlı sayıda derslikler ve zaman dilimleri arasına belirli sayıda kısıtları sağlamak ve sağlanmama durumunda en az ceza alması şartıyla atanmasıdır. Eğer faaliyetler kümesi E , derslikler kümesi R , zaman dilimler T ise toplam atanabilen durumların sayısı veya çizelge sayıları $|E|^{|R||T|}$ dir ve bu problemin NP sayılma nedenidir. Burada önceden belirlenmiş bütün katı kısıtları sağlayan atamalar mümkün çözüm sayılır iken, esnek kısıtların sağlanmaması durumunda ise atamalara ceza verilmektedir [91]. Bu problem çok sayıda teorik olarak yapay zekâ araştırmacıları ve pratik olarak yöneylem araştırmacıları tarafından ele alınmıştır. Son dönemlerde araştırmacıların ilgilerinden dolayı standart problemler tanımlanmıştır ve bu ölçüt problemlerinde örnek veriler hazırlanmıştır. Bu veriler araştırmacılar tarafından zaman çizelgeleme probleminin çözümü için önerilen algoritmaların, karşılaştırılması için uygun bir ortam sağlamışlardır. Eğitimsel zaman çizelgeleme probleminin üzerinde tanımlanan bu ölçüt problemlerinin bazıları ITC-2002, Socha, ITC-2007 tır. Bu ölçüt problemler Bölüm 1.3.2’de ayrıntılı olarak açıklanmıştır.

Bölüm 1.4’de açıklandığı üzere bu problemin çözümü için önerilen yöntemler literatürde Petrovic ve arkadaşları [38] tarafından sekiz sınıfa ayrılmıştır.

- Sıralı Yöntemler
- Kümeleme Yöntemleri
- Kısıt Tabanlı Yöntemler
- Meta Sezgisel Yöntemler
- Melez Evrimsel ve Melez Meta Sezgisel Algoritmalar
- Çok Ölçütlü Yaklaşım
- Durum-Tabanlı Düşünme Yaklaşımları
- Üst sezgisel ve Uyabilen Yaklaşımlar”

Bu yöntemler ayrıntılı bir şekilde Bölüm 1.4’de açıklanmıştır. Problemin NP türünden olmasından dolayı problem uzayı büyüdükçe klasik yöneylem araştırma yöntemleri tarafından kesin çözümün bulunması zorlaşacaktır veya mümkün olmayacaktır. Bu sebepten dolayı bilgisayar aracılığıyla stokastik çözüm bulan yapay zekâ yöntemleri araştırmacılar tarafından ele alınmıştır. Bu yöntemlerin çoğu meta sezgiseller ve meta sezgisellerin melez kullanımına aittir. Lewis [26] araştırmacılar tarafından üniversite ders çizelgeleme problemlerinin çözümü için 2006 yılına kadar yapılan meta sezgisel yaklaşımları üç sınıfa (Tek Aşamalı Optimizasyon Algoritmalar, İki Aşamalı Optimizasyon Algoritmalar, Gevşemeye İzin Veren Algoritmalar) ayırmıştır. Ayırma işlemi katı ve esnek kısıtlarının sağlanmasına bağlı olarak yapılmış ve ayrıntılı açıklama Bölüm 1.4.4.1’de yapılmıştır.

Üniversite ders çizelgeleme problemi için literatürde bulunan başarılı algoritmaların çoğu iki aşamalıdır. İlk aşamada çözümlerin üretimi için esnek kısıtlar göz ardı edilirken katı kısıtların mutlaka sağlanması gerekir. İkinci aşama ise mümkün çözümleri bozmadan esnek kısıtları sağlamaya çalışırken bunların sağlanmaması halinde verilen cezaları minimum yapmaktır. Bu tür yaklaşımlarda birinci ve ikinci aşamalar için farklı yöntemlerden yararlanılabilir.

Meta sezgisel yöntemlerin bazıları doğadan ilham alarak doğada yapılan işlemlerin benzetimini yapmışlardır. Örneğin; Genetik Algoritma [66], Karınca Kolonisi Algoritması [73], Tavlama Benzetimi [86], Parçacık Sürü Zekâsı (PSO) [103], gibi algoritmalar üniversite ders çizelgeleme probleminin çözümü için ele alınmıştır.

Son yıllarda Derviş Karaboğa tarafından tanımlanan Yapay Arı Kolonisi Algoritması (ABC) [97] doğada bulunan bal arılarının yiyecek aramasından yola çıkarak, sürü zekası sayılan yeni bir meta sezgisel yöntemi üretmiştir. Bu algoritma temel olarak sürekli uzayda nümerik optimizasyon problemleri için tanımlanmış ve denemelerde diğer meta sezgisel

yöntemlere karşı daha iyi sonuçlar elde etmiştir [98]. Son yıllarda ABC algoritması ayrık uzaydaki zaman çizelgeleme probleminin bir türü olan flow shop gibi NP optimizasyon problemlerinin çözümü için de kullanılmış ve iyi sonuçlar elde edilmiştir. [104, 105, 106, 107, 108]. ABC algoritması eğitimsel zaman çizelgeleme probleminin bir türü olan sınav çizelgeleme için de kullanılmıştır [109, 110].

[111, 112]'de üniversite ders çizelgeleme probleminin çözümü için ABC algoritması melez bir şekilde kullanılmış, bu yaklaşımların her birinde belli üniversitenin belli bir bölümünün ders çizelgesi ele alınmıştır. 2012 tarihine kadar Türkiye'de araştırmacılar tarafından üniversite ders çizelgeleme probleminin çözümü için CB-CTT örnek problemi ele alınmamıştır. CB-CTT üniversite ders çizelgeleme probleminin çözümü ABC ile [113] ele alınmış ise maalesef elde edilen sonuçlar iyi değildir. ABC algoritmasıyla 2012 yılına kadar yapılan tüm çalışmalar [114]'de ayrıntılı bir şekilde verilmektedir. ABC açıklanması ayrıntılı bir şekilde Bölüm 1.8'de açıklanmıştır.

2.2. Problem Tanıma

Üniversite ders çizelgeleme problemi son yıllarda çoğu araştırmacı tarafından ele alınmaktadır ve bilgisayardan yararlanan yeni optimizasyon yöntemlerinin gelişmesiyle birlikte, bu problemin çözümü için farklı yaklaşımlar ele alınmıştır. Gerçek dünyada olan üniversitelerin farklı istek ve kısıtlarından dolayı bunların çözümü için ele alınan yaklaşımların karşılaştırılması ve hangisinin iyi sonuçlar elde ettiğinin belirlenmesi mümkün değildir. Bu nedenle üniversite ders çizelgeleme problemi için ortak bir tanım ve belli örnek veriler belirlendi. Bunlardan önemli olanların ve literatürde araştırmacılar tarafından yaygın şekilde kullanılanlardan bazıları ayrıntılı bir şekilde Bölüm 1.3.2'de verilmektedir.

Üniversite ders çizelgeleme üzerine örnek problemlerin en gelişmiş ICT-2007'nin üçüncü bölümünde tanımlanan müfredat tabanlı üniversite ders çizelgeleme (CB-CTT) tır. Bu ölçüt problemin diğerleri ile farkı ve özellikleri detaylı bir şekilde bölüm 1.3.2'de verilmektedir. Buradaki en önemli fark, bu örnek problemin gerçek bir üniversite ders çizelgeleme probleminin biraz daha basitleştirilmiş hali olması ve örnek verilerin İtalyan'ın Udine Üniversitesinde olan dersler ve müfredat üzerinden verilmesidir.

ITC-2007 yarışmasının resmi web sitesinde 21 tane örnek veri verilmektedir. Her bir örnek veri tek bir dosya halinde gelmekte ve dosyalar comp01.ctt, comp02.ctt, comp03.ctt,

... comp21.ctt şeklinde isimlendirilmektedir. Her dosya bir dosya başlığından ve dört farklı bölümden (courses, rooms, curricula ve constraints) oluşmaktadır. Dosya başlığı tüm sayısal değerleri ve her bölümde problemin belirli yönü için bilgileri diziler halinde sağlamaktadır. Dosyaların tam biçimi Tablo 8’de gösterilmiştir.

Tablo 8. Örnek problem için verilen dosya

```

Name: ToyExample
Courses: 4
Rooms: 2
Days: 5
Periods_per_day: 4
Curricula: 2
Constraints: 8

COURSES:
SceCosC Ocra 3 3 30
ArcTec Indaco 3 2 42
TecCos Rosa 5 4 40
Geotec Scarlatti 5 4 18

ROOMS:
A 32
B 50

CURRICULA:
Cur1 3 SceCosC ArcTec TecCos
Cur2 2 TecCos Geotec

UNAVAILABILITY_CONSTRAINTS:
TecCos 2 0
TecCos 2 1
TecCos 3 2
TecCos 3 3
ArcTec 4 0
ArcTec 4 1
ArcTec 4 2
ArcTec 4 3

END.

```

Burada günlerin indeksi ve her gün içerisinde zaman dilimlerin indeksi sıfırdan başlamaktadır

Çeşitli bölümlerin verileri Tablo 9’da verilen yapıya sahiptir.

Tablo 9. Örnek verilerin dosya da bilgilerin biçimi

Courses: <CourseID> <Teacher> <# Lectures> <MinWorkingDays> <# Students>
 Dersler: <Dersin ismi> <Dersi veren öğretim elemanın ismi > <Oturumlar sayısı > <Minimum verme gün sayısı> <Dersi alan öğrenci sayısı>

Rooms: <RoomID> <Capacity>
 Derslikler: <Dersliğin ismi> <Kapasitesi>

Curricula: <CurriculumID> <# Courses> <MemberID> ... <MemberID>
 Müfredat: <Müfredatın ismi > <Bu müfredatta olan dersler sayısı > <Ders isimi>...<Ders isimi>

Unavailability_Constraints: <CourseID> <Day> <Day_Period>
 Kullanıla bilmeyen kısıtları : <Ders ismi> <Gün> <Zaman dilimi>

Bu problemde amaç, verilen 21 tane örnek problemin tümü için belirlenen zaman içerisinde tek işlemci bilgisayar kullanmak şartıyla, en minimum toplam cezaya (11) sahip mümkün çözüm üretmektir.

Her dosyadan problem için verilen bilgiler dört farklı veri yapılarına aktarılır. Alt bölümde bu veri yapılarının biçimi açıklanmaktadır:

- $C = \{c_1, c_2 \dots c_n\}$: Her c_i , i dersi gösterir ve beş farklı bölüm (dersin ismi, dersi anlatan öğretim görevlisinin ismi, dersin haftaya atanan oturum sayısı, oturumlar için en minimum atanan gün sayısı, dersi alan öğrenci sayısı) içerir,
- $CR = \{cr_1, cr_2 \dots cr_s\}$: Her cr_i ; i müfredatı gösterir ve her biri için müfredatın ismini, müfredatta olan derslerin sayısını, müfredatta olan derslerin isim bilgilerini içermektedir,
- $R = \{r_1, r_2 \dots r_m\}$: r_i ; i dersliği gösterir ve iki farklı bilgiyi (dersliğin ismi, dersliğin kapasitesi) içerir,
- $U = \{u_1, u_2 \dots u_a\}$: Her u_i : i ders için (c_i) kullanılmaz periyotların kümesini (gün, zaman dilimi) göstermektedir ($a \leq n$).

Problem için aday çözümü bir X Eğitim gün sayısı \times Zaman dilimler sayısı \times Derslik sayısı matrisiyle gösterilmektedir. Eğer c_i dersin bir oturumu i inci gün, j inci zaman dilimine ve k inci dersliğe atanırsa o zaman bu matrisin $x_{i,j,k}$ hücresinin değeri, c_i dersin indeksi olur. Matrisin hücresine ders atanamazsa atanamayan hücre -1 değerini alır.

CB-CTT problemin ilk matematiksel tanıtımı [83] yapılmıştır ve önerilen tanıtımın temeli yapılan çalışmaya aittir. Bu problemin matematiksel tanıtımı için kullanılan simgeler alt bölümde verilmektedir.

- $D = \{d_1, d_2 \dots d_p\}$: p tane iş günü,
- $T = \{t_1, t_2 \dots t_h\}$: h tane periyod,

- $E(expr)$: $expr$ doğru ise bu fonksiyon 1 değerini yoksa 0 geri döndürür,
- l_z : c_z dersin oturum sayısı,
- $Length\{\dots\}$: Kümenin eleman sayısını geri döndürür.

Problemdeki Katı Kısıtların Matematiksel Tanıtımı:

- Ders Oturumları (Lectures, H1):

$$\forall c_z \in C \quad \sum_{i=1..p, j=1..h, k=1..m} E\{x_{i,j,k} == c_z\} = l_z$$

- Derslik Kullanımı (Room occupancy, H2):

$$Length\{x_{i,j,k} \mid x_{i,j,k} \neq -1, i = 1..p, j = 1..h, k = 1..m\} = \sum_{z=1..n} l_z$$

- Çakışmalar (Conflicts, H3):

$$\sum_{i=1..p, j=1..h} E\{(x_{i,j,k_1} \rightarrow \text{öretim elemanı} == x_{i,j,k_2} \rightarrow \text{öretim elemanı}) \text{ OR } cur(x_{i,j,k_1}) \cap cur(x_{i,j,k_2}) \neq \emptyset, k_1, k_2 \text{ is } k = 1..m\} = 0$$

- Uygunluk (Availabilities, H4):

$$\sum_{i=1..p, j=1..h} E\{x_{i,j,k} == c \in C \ \&\& \ (i, j) \in u_c\} = 0$$

Problemdeki Esnek kısıtların Matematiksel Tanıtımı:

- Derslik Kapasitesi (Room Capacity, S1):

$$\forall x_{i,j,k} = c_z \in X,$$

$$p_1(x_{i,j,k}) = \begin{cases} (c_z \rightarrow std) - (r_k \rightarrow cap) & \text{if } ((c_z \rightarrow std) > (r_k \rightarrow cap)) \\ 0 & \text{otherwise} \end{cases}$$

- Sabit Derslik (Room Stability, S2):

$$\forall c_z \in C, p_2(c_z) = Length\{k \mid x_{i,j,k} == c_z, i = 1..p, j = 1..h, k = 1..m\} - 1$$

- Minimum İş Günü (Minimum Working Days, S3):

$$\forall c_z \in C,$$

$$wd(c_z) = \sum_{i=1..p} E\{x_{i,j,k} == c_z, j = 1..h, k = 1..m\},$$

$$p_3(c_z) = \begin{cases} c_z \rightarrow \text{min verilme gün} - wd(c_z) & \text{if } wd(c_z) < (c_z \rightarrow \text{min verilme gün}) \\ 0 & \text{otherwise} \end{cases}$$

- Müfredat Yoğunluğu (Curriculum Compactness, S4):

$$\forall cr_z \in CR,$$

$$p_4(cr_z) = \sum_{i=1..p, j=1..h-1} E\{cr_z \in cur(\{x_{i,j,k} \mid k = 1..m\}) \text{ XOR } cr_z \in cur(\{x_{i,j+1,k} \mid k = 1..m\})\}$$

Her aday için çözümdeki tüm esnek kısıtların sağlanmama sayısı tek tek hesaplanmaktadır. Daha sonra, çözümün maliyet fonksiyonu denklem (11) yardımıyla hesaplanır.

$$p(X) = \sum_{x_{i,j,k}} p_1(x_{i,j,k}) + \sum_{c_z \in C} p_2(c_z) + \sum_{c_z \in C} 5 * p_3(c_z) + \sum_{cr_z \in CR} 2 * p_4(cr_z) \quad (11)$$

Yarışmanın resmi sitesinde [28] her örnek problemin çözümünün değerlendirilmesi için bir uygulama bulunmaktadır. Bu uygulanmadan yararlanabilmek için her çözüm istenilen biçimde bir metin (.txt) dosyasına kayıt edilmesi gerekir. Bu dosyanın her satırında atanan ders oturumlarının ismi, atanan dersliğin ismi, atanan gün ve atanan zaman diliminin ismi kayıt edilmelidir. Bu biçim aşağıda verilmiştir:

<CourseID> <RoomID> <Day> <Day_Period>

Örneğin; ToyExample.ctt örnek probleminin bir aday çözümü Tablo 10'da gibi bir Output.out dosyasında kayıt edilmektedir.

Tablo 10. ToyExample.ctl problem için bir aday çözüm dosyası (Output.out)

```
SceCosC B 3 0
SceCosC A 3 1
SceCosC A 4 0
ArcTec B 0 1
ArcTec B 1 1
ArcTec B 1 2
TecCos B 0 0
TecCos A 0 1
TecCos B 2 2
TecCos B 4 2
TecCos B 4 3
Geotec A 2 2
Geotec A 2 3
Geotec B 3 0
Geotec A 3 1
Geotec A 4 2
```

Uygulamada istenilen dosya girdileri, komut penceresinde koştuktan sonra bu aday çözümün değerlendirilip Tablo11’de verildiği gibi olan tüm katı kısıtların ve esnek kısıtların sağlanmama yeri ve bu sebepten alınan cezayı hesaplanmakta ve ayrıntılı mesajlarla göstermektedir.

Bu Tablonun ilk satırında gösterilen mesajın anlamı ArcTec ve TecCos derslerin oturumları aday çözümde birinci zaman periyodunda aynı zamana atanmışlar. Bu da H3 katın kısıtın bozulmasına neden olmaktadır çünkü bu dersler Cur1 müfredat içerisindedirler.

Tablo 11. Aday çözümün verilen uygulamayla değerlendirmesi

[H] Courses ArcTec and TecCos have both a lecture at period 1 (day 0, timeslot 1)
 [H] Courses TecCos and Geotec have both a lecture at period 10 (day 2, timeslot 2)
 [H] Courses TecCos and Geotec have both a lecture at period 18 (day 4, timeslot 2)
 [H] 2 lectures in room B the period 12 (day 3, timeslot 0)
 [H] 2 lectures in room A the period 13 (day 3, timeslot 1)
 [S(8)] Room A too small for course TecCos the period 1 (day 0, timeslot 1)
 [S(5)] The course SceCosC has only 2 days of lecture
 [S(5)] The course TecCos has only 3 days of lecture
 [S(5)] The course Geotec has only 3 days of lecture
 [S(2)] Curriculum Cur1 has an isolated lecture at period 10 (day 2, timeslot 2)
 [S(2)] Curriculum Cur1 has an isolated lecture at period 16 (day 4, timeslot 0)
 [S(1)] Course SceCosC uses 2 different rooms
 [S(1)] Course TecCos uses 2 different rooms
 [S(1)] Course Geotec uses 2 different rooms

Violations of Lectures (hard) : 0
 Violations of Conflicts (hard) : 3
 Violations of Availability (hard) : 0
 Violations of RoomOccupation (hard) : 2
 Cost of RoomCapacity (soft) : 8
 Cost of MinWorkingDays (soft) : 15
 Cost of CurriculumCompactness (soft) : 4
 Cost of RoomStability (soft) : 3

Summary: Violations = 5, Total Cost = 30

2.3. Yöntem

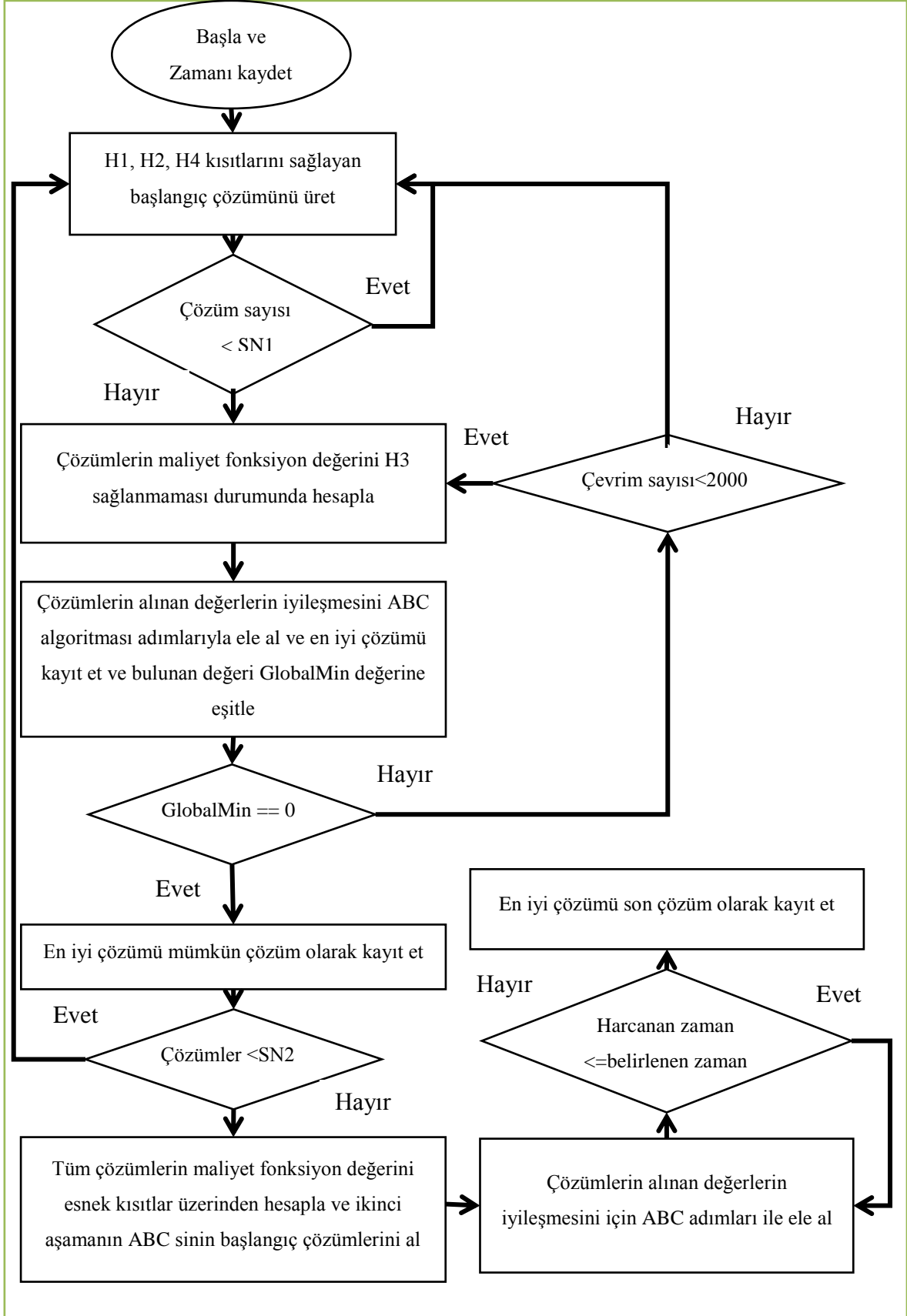
Problemin çözümü için önerilen algoritma, Lewis [26, 58] sınıflandırmasında iki aşamalı meta sezgisel algoritmalar sınıfına girmektedir. Önerilen algoritmanın her aşamasında farklı maliyet fonksiyona sahip popülasyon tabanlı ABC algoritmasından yararlanılmaktadır.

Birinci aşamada ele alınan örnek problem için sadece katı kısıtların sağlanması için çalışılır, esnek kısıtlar göz ardı edilir. Bu aşamanın amacı belirlenen sayı kadar mümkün çözüm üretmektir.

Birinci aşamada ABC algoritmasından yararlanması için problemin bir amaç fonksiyonuna sahip optimizasyon problemine dönüşmesi gerekir. Bu yüzden başlangıç çözümlerinin üretiminde, çözüm matrisinde ders oturumlarının rastgele bir yere atanmasında dört katı kısıttan $H3$ kısıtı hariç diğer tüm kısıtların sağlanması gerekmektedir. Amaç fonksiyonunun değeri $H3$ kısıtını sağlamayan oturumların sayısı kadardır. ABC algoritması, döngüler sırasında $SN1$ tane üretilen başlangıç çözümleri ele alınarak tanımlanan amaç fonksiyon değerini azaltmak için kullanılır. En iyi çözüm değeri sıfıra ulaşıncaya döngü sonlanır ve tüm katı kısıtları sağlayan çözüm (mümkün çözüm) elde edilmiş olur.

Başka bir döngü de $SN2$ tane mümkün çözüm üretimi için ele alınır. Bu veriler ikinci aşamada kullanılacak ABC algoritmasının başlangıç çözümleri sayılır. Bu çözümlerin her birisi için esnek kısıtları sağlanmamasından dolayı alınan cezaların toplamı hesaplanır.

İkinci aşamada, ilk aşamanın çıktısı olan mümkün çözümler girdi olarak ele alınır, buradaki amaç fonksiyonu ise esnek kısıtların sağlanmamasından dolayı alınan cezaların toplamıdır. Çözümlerde katı kısıtların bozulmadan esnek kısıtların sağlanması ele alınır. Bunun amacı, belirlenen zaman içerisinde çözümlerde tanımlanan amaç fonksiyon değerini minimum yapmaktır. Tüm yapılan işlemler Şekil 9'da bir akış diyagramı ile gösterilmektedir.



Şekil 9. Önerilen yöntemin akış diyagramı

Bu aşamalarda kullanılan komşuluk kavramı alt bölümde ayrıntılı bir şekilde açıklanmaktadır ve türleri verilmektedir.

2.3.1. Komşuluk Arama Mekanizmaları

Lü ve diğerleri [115] önceden literatürde bulunan üç farklı komşu arama mekanizmasının (SimpleMove, SimpleSwap, KempeMove) ve kendilerinin önerdiği (KempSwap) mekanizmayı ITC-2007'nin üçüncü bölümü üzerinde tek tek veya birleşmiş durumlarını deneyip sonuçlarını karşılaştırmışlardır. Bunların her birinin açıklaması alt bölümde verilmektedir.

Problemin tanıtımında C (n tane ders), R (m tane derslik), T (h tane zaman periyod) kümeleri ele alınır. Çözümü göstermek için direk gösterimden yararlanılır yani çözüm $X_{h \times m}$ iki boyutlu matris ile temsil edilir. Bu matriste ilk boyut zamanı, ikinci ise derslikleri gösterir. $x_{i,j}$ i . zaman periyodunda ve j . derslikte olan dersin indeksidir. Eğer bu periyod ve dersliğe bir ders atanmazsa o zaman -1 değeri alır.

Derslerin atanması mümkün çözüm ürettiği zaman yani tüm atanmalar dört katı kısıtı ($H1, H2, H3, H4$) sağladığında, bu çözümde esnek kısıtların sağlanmaması için alınan cezaları katı kısıtları bozmadan azaltmak için yerel arama işlemi ele alınır. Bir başka deyişle yerel arama yapılacak arama uzayı mümkün çizelgelerle sınırlandırılmış olur. Oluşturulan tüm mümkün çizelgeler χ matrisler kümesiyle gösterilir.

Yerel arama işleminde X aday çözümü üzerinde mv hareketi yeni çözümün oluşturur ve bu işlem $X \oplus mv$ ifadeyle gösterilir. Eğer $M(X)$, X çözümü üzerinde katı kısıtları bozmadan uygulanabilen tüm mümkün hareketler olursa, o halde X Komşuluğu (Neighborhood) aşağıdaki şekilde tanımlanır:

$$N(X) = \{ X' \in \chi \mid X' = X \oplus mv, mv \in M(X) \}$$

CB-CTT problemi için dört farklı (Kaydırma (SimpleMove), Değiştirme (SimpleSwap), Kempe kaydırma (KempeMove), Kempe Değiştirme (KempeSwap)) hareket (mv) tanımlanmaktadır ve bu hareketler dört farklı komşuluk $N1, N2, N3, N4$ oluşturmasına neden olur. Bunlardan ilk üçü önceden literatürde bulunmakta olup, dördüncüsü ise [114] yeni olarak önerilmiştir.

- N1 Komşuluk Yapısı: Bu yapıda kullanılan hareket türü kaydırmadır. Kaydırma işleminde, i . zaman periyodunda ve j . derslikte olan bir dersin bir oturumu $x_{i,j}$ ise, boş

bulunan bir yere (i' 'inci zaman periyoduna ve j' 'inci dersliğe), $i \neq i'$ ve $j \neq j'$ taşınması demektir. Bu taşınmada $x_{i',j'} = x_{i,j}$ ve $x_{i,j} = -1$ değerini alır.

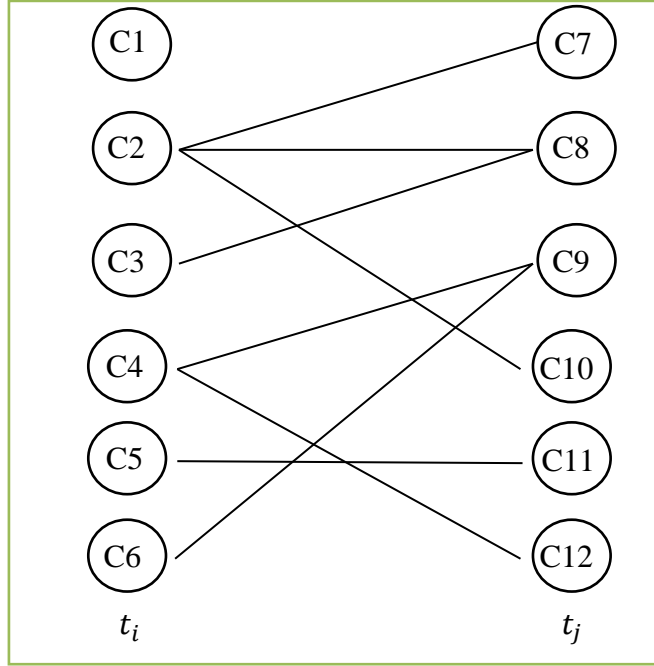
Bu da $x_{i,j} = -1$ t_i zaman periyodunda ve r_j derslikte hiç bir dersin oturumu olmaması anlamına gelmektedir.

- N2 Komşuluk Yapısı: Bu yapıda kullanılan değiştirme hareketi iki farklı dersin oturumlarının yerleştiği zaman periyotlarının ve dersliklerinin birbiriyle değiştirilmesi anlamına gelmektedir. Yani X çözümü üzerinde $x_{i',j'}$, $x_{i,j}$ değerlerinin birbiriyle yer değiştirmesine karşılık gelir.
- N3 Komşuluk Yapısı: Bir KempeMove türünden olan hareket Kempe chain interchange ile gerçekleşmektedir. CB-CTT problemi için aday çözüm olan her X bir G grafına dönüşebilir. Bu grafa düğümler derslerin oturumlarını temsil eder. Eğer bu derslerin öğrencileri veya öğretim elamanları ortak olursa, bu düğümler arasında bir bağlantı (edage) oluşur. Bundan dolayı her dersin oturumları arasında bir bağlantı olmak zorundadır.

Her bir Kempe chain iki farklı periyotta birbiriyle bağlantısı olan ders oturumları kümesidir. Eğer K , t_i, t_j zaman periyodundan oluşan Kampe chain ve $L_i(L_j)$ bu periyotlarda olan ders oturum kümeleri olarak varsayılırsa, o halde bir Kampe chain interchange L_i 'in $(L_i \setminus K) \cup (L_j \cap K)$ ve L_j 'in $(L_j \setminus K) \cup (L_i \cap K)$ yer değişimiyle gerçekleşir. Bu değişimin yapılması için $|K| \geq 3$ olmalıdır.

Örneğin şekil 10'da t_i, t_j periyodunda olan ders oturumları kümesi, her derslikte sadece bir ders olma koşuluyla gösterilmektedir. Bu küçük örnekte var olan dört Kampe chain şu şekildedir: $K1 = \{c_1\}$, $K2 = \{c_5, c_{11}\}$, $K3 = \{c_2, c_3, c_7, c_8, c_{10}\}$, $K4 = \{c_4, c_6, c_9, c_{12}\}$

Oluşan Kempe chainlerin arasından sadece $K4$ mümkün çözüm üreten KempMove yapabilir diğerlerinde ise bu hareket yapılamamaktadır. Bunun nedeni $K1$ ve $K2$ her birisinin içerisinde olan ders oturumlarının sayısı 3'den az olmasıdır. $K3$ 'de ise bu değişim bir mümkün çözüm üretmez. Bunun nedeni $\{c_2, c_3\}$, $\{c_7, c_8, c_{10}\}$ ders oturum kümeleri takas yapılırsa t_i periyodunda olan derslerin sayısı var olan derslik sayısını aşmasıdır. Bu kısıtlamaya derslik dağıtımını bozma ismi (room allocation violation) verilmektedir.



Şekil 10. Kemp chain tanıtımı[114].

Gerçekten de ders oturum sayısının asimetrik özelliği, bu komşuluk için kabul edilen aday çözümlerin sayısını büyük ölçüde kısıtlar ve komşulukta zayıflık oluşturur. Diğer tanımlanan komşuluk yapısı yani $N4$, KempeMove ile karşılaştırıldığında daha esnek ve bu kısıtlamayı önlemektedir.

$N4$ Komşuluk Yapısı: $N3$ 'de iki farklı periyot üzerinde olan ders oturumları içerisinde birbiriyle bağlantısı olan sadece tek bir küme ele alınır. KempeSwap ise burada bulunan iki farklı kampe chain ders oturumlarının yer değişiminden ibarettir. Biçimsel olarak $K1, K2, t_i, t_j$ zaman periyodundan oluşan alt grafın üzerinde iki farklı Kampe chain varsa, bir KempeMove, L_i 'in $(L_i \setminus (K1 \cup K2)) \cup (L_j \cap (K1 \cup K2))$ ve L_j 'in $(L_j \setminus (K1 \cup K2)) \cup (L_i \cap (K1 \cup K2))$ yer değiştirmesine neden olur. KempeSwap, KempeMove'nin genelleşmiş hali sayılabilir.

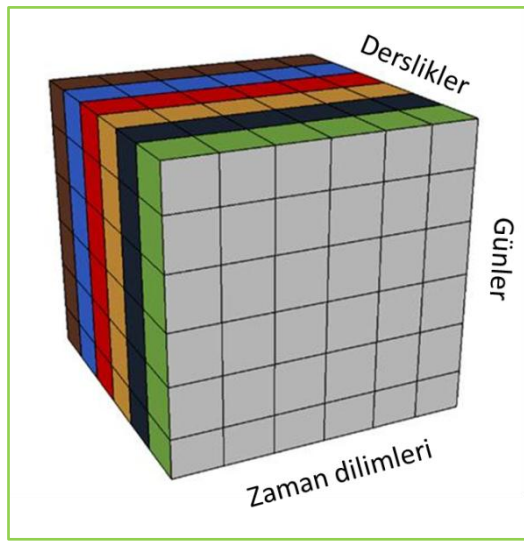
[114] çalışmasında bu komşuluk arama mekanizmalarının tek ve birleşmiş hali kullanımı üzerinde araştırmalar yapılmıştır ve CB-CTT problemi üzerinde denemeler yapılarak ve sonuçları gösterilmiştir. Bu sonuçlar tek birinin kullanımı için $N4$ komşuluk yapısını diğerlerine göre daha güçlü göstermiş ve birleşmiş hallerinin kullanılması durumunda ($N1 \rightarrow N4$) döngü içerisinde kullanımı daha iyi sonuçlar elde etmiştir.

2.3.2. Kullanılan Veri Yapıları

Genellikle literatürde zaman çizelgeleme problemleri basit gösterim için bir tür permütasyon problemi olarak ele alınmaktadır [108]. Faaliyetlerin farklı zaman periyotlarına atanması, farklı çözümler ve sonuç olarak farklı permütasyonları temsil etmektedir. Zaman çizelgelemenin bir türü olan ders çizelgeleme problemi için de bu geçerlidir. Ders çizelgesi iki boyutlu $X_{i \times j}$ matrisiyle ifade edilmekte, burada i zaman periyotlarının indekslerin ve j ise dersliklerin indeksini göstermektedir. Eğer günlerin ve zaman dilimlerinin indeksi birden başlanırsa, k . gün , m . zaman diliminden ve j inci dersliğe atanan dersten i değeri (12) den hesaplanır ve $x_{i,j}$ ise değerini dersin indeksinden almaktadır. Eğer buraya bir ders atanmazsa -1 değerini alır.

$$(k - 1) * \text{her gün içerisinde olan zaman dilimleri sayısı} + m \quad (12)$$

CB-CTT problemi için her aday çözüm önerilen yaklaşımda üç boyutlu $X_{\text{Eğitim gün sayısı} \times \text{Zaman dilimleri sayısı} \times \text{Derslik sayısı}}$ Şekil 11’de verildiği üç boyutlu matrisle gösterilmektedir. $x_{i,j,k}$ i günün, j zaman diliminin ve k derslikte olan ders oturumunun indeksini alır. Eğer bunlardan herhangi biri boş ise -1 değerini almaktadır. Her örnek verilerin dosyası okunduktan sonra, matrisin boyutları belirlenir ve matris inşa edilir.



Şekil 11. Çözümü matrisi

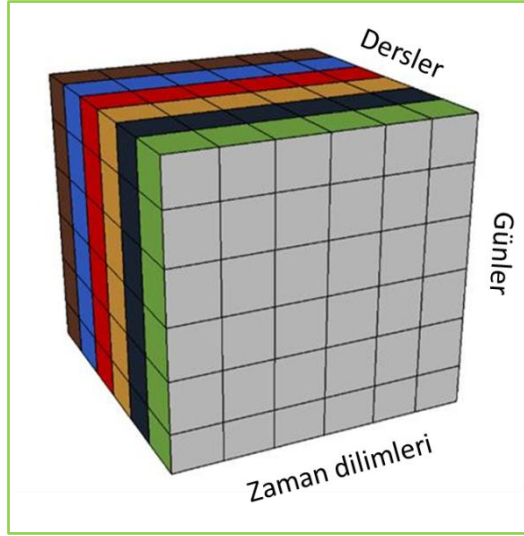
Her örnek problem ele alındığında, probleme ait dosyadan okunan verilerin kayıt edilmesi gerekmektedir. Bu yüzden veri yapılarının tanımlanması gerekir. Her dosyanın biçimi Tablo 9 da verilmektedir. Courses (Dosyanın başlığında dersler sayısı), Rooms (derslikler sayısı), Days (Eğitim gün sayısı), Periods_per_day (Gün içerisinde zaman dilimleri sayısı), Curricula (müfredat sayısı) ve Constraints (öğretim elemanları tarafından belirlenen sınırlamalar sayısı), sırasıyla Corses_Number, Rooms_Number, Days_Number, Period_Per_Day_Number, Curricula_Number ve Constraints_Number değişkenleri ile tanımlanır ve verilerin değerleri bu değişkenlere aktarılır.

Derslerin bilgilerini saklamak için COURSES_{1×Corses_Number} dizisi tanımlanır. Bu dizinin her bir hanesinin içeriğine CourseID (Ders ismi), Teacher (Dersi veren öğretim elemanının ismi), Lectures (Oturumlar sayısı), MinWorkingDays (Minimum iş gün sayısı), Students (Dersi alan öğrenci sayısı) değerleri direk dosyadan aktarılır. Bunlara ek olarak her dersin bilgilerine Group_No_{1×Curricula_Number} dizisi (dersin hangi müfredatta olduğunu belirleme için) ve Prof_ID (Dersi veren öğretim elemanının indeksi, diğer matrislerde dersin öğretim elemanı üzerinden direk ulaşmak için) saklanmaktadır.

Derslik bilgilerini saklamak için ROOMS_{1×Rooms_Number} dizisi tanımlanır. Bu dizinin her bir hanesinin içeriğine RoomID (Derslik ismi), Capacity (Derslik kapasitesi) değerleri direk dosyadan aktarılır.

Müfredat bilgilerini saklamak için CURRICULA_{1×Curricula_Number} dizisi tanımlanır. Bu dizinin her bir hanesinin içeriğine CurriculumID (Müfredat ismi), No_Courses_Cu (Müfredat olan ders sayı) ve Courses_{1×No_Courses_Cu} dizisi (Müfredatta olan derslerin ismi) değerleri direk dosyadan aktarılır. Bu değerler aktarılırken COURSES dizisinde olan ve her dersin müfredatını gösteren Group_No dizi değerleri verilir.

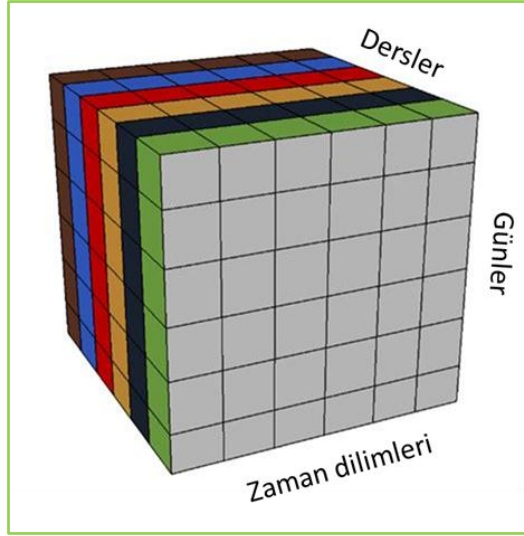
Öğretim elemanları tarafından dersler için belirlenen sınırlamaları saklamak için UNAVAILABILITY_CONSTRAINTS_{Days_Number × Periods_per_day_Number × Corses_Number} şekil 12’de görüldüğü gibi bir matrisi tanımlanır ve içeriği sıfırlanır. Bu bölümde dosyada olan bilgiler CourseID (Ders ismi), Day (Gün), Day_Period (Gün içerisinde zaman dilimi) alınır ve CourseID yardımıyla dersin indeksi belirlenir sonra UNAVAILABILITY_CONSTRAINTS_{Day,Day_Period,Ders_Indeks} matrisinin değerleri 1 alınır ve bu verilen dersin belirlenen zaman periyoduna atanmaz anlamına gelmektedir.



Şekil 12. UNAVAILABILITY_CONSTRAINTS matrisi

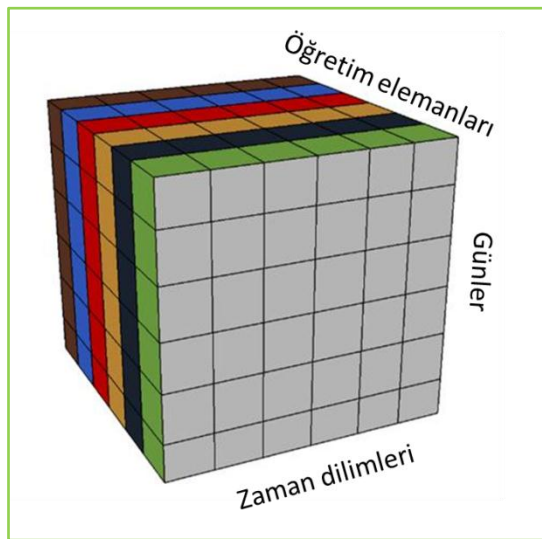
Her çözümün komşuluk üretiminde gerçekleştirilmesinde katı kısıtların sağlandığının incelemesi veya esnek kısıtların sağlanmaması durumunda alınan cezanın hesaplaması direk çözüm üzerinden yapılırsa fazla işlem gerektirmektedir. Bu işlemlerin sayısını azaltmak ve süreci hızlandırmak için üç tane matris tanımlanması gerekir. Her çözüm ele alındığında bu matrislerin değerleri çözüm üzerinden hesaplanır ve çözümde değişiklik yapıldığında bu matrislerin değerleri de güncelleşir.

Bunlardan birincisi $Room_of_course$ $Days_Number \times Periods_per_day_Number \times Courses_Number$ şekil 13’de görüldüğü gibi bir matrisidir. Eğer çözümde i gün, j zaman periyodunda ve k derslikte olan dersin indeksi c_i ise o zaman $Room_of_course$ $_{i,j,c_i}$ hanesinin değeri k olur, eğer bu hane boşise -1 değeri atanır. Bu matris üzerinden her dersin oturumlarının hangi gün, hangi saat ve hangi derslikte olduğu çözümün tümü aranmadan belirlenir.



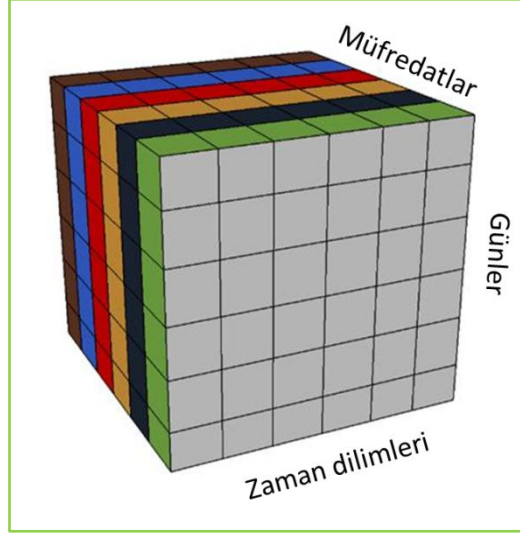
Şekil 13. Room_of_course matrisi

İkinci matris $Prof_time$ $Days_Number \times Periods_per_day_Number \times Prof_Number$ şekil 14'te görüldüğü gibi bir matrisidir. Eğer çözümde i 'inci günde, j zaman periyodunda ve k 'inci derslikte olan dersin indeksi c_i ise ilk olarak c_i dersin öğretim eleman indeksi (Prof_ID) COURSES dizisinden bulunur sonra $Prof_time_{i,j,Prof_ID}$ değeri 1 artırılır. Eğer öğretim elemanı bir periyot zamanında boşa o zaman $Prof_time_{i,j,Prof_ID}$ değeri olarak 0 atanır. Bu matris yardımıyla çözümün tümünü aramadan, direk her zaman periyodunda aranan öğretim elemanın boş ya da dolu olduğu belirlenir.



Şekil 14. Prof_time matrisi

Üçüncü matris *Group_time* $Days_Number \times Periods_per_day_Number \times Curricula_Number$ şekil 15'te görüldüğü gibi bir matrisidir. Bu matrisin başlangıç değeri 0'dır. Çözümde her zaman periyodunda ki derslerin var olan müfredat üzerinden bu matriste o zaman periyodunda olan müfredatı 1 artırılır. Bu matris üzerinden çözümün her zaman periyodunda, hangi müfredattaki ders olduğu çözümün tümü aranmadan direk belirlenir.



Şekil 15. Group_time matrisi

2.3.3. İlk Aşama

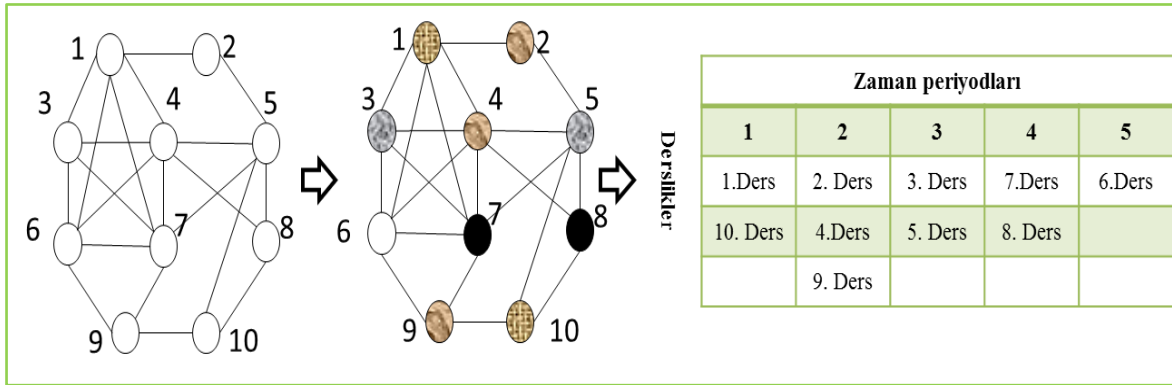
2.3.3.1. İlk Aşama Literatür Analizleri

Bu aşama için literatürde en fazla kullanılan yöntemlerden birisi, verilen zaman çizelgeleme problemini bir grafa dönüştürmek ve sonra bu grafi belirli sayıda renkle renkleme. Üniversite ders çizelgeleme problemini bir grafa dönüştürken, ders oturumlarının garfının düğümleri ve bir zaman periyodunda ki iki faaliyet çakışiyorsa, faaliyetleri temsil eden düğümler arasında bir bağlantı var demektir, yani her dersin oturumunu temsil eden tüm düğümler arasında ve aynı öğretim elemanının verdiği dersler arasında bağlantı var olduğu anlamına gelmektedir.

Zaman periyotlarının her birisi bir renge denk gelmektedir ve her rengin sayısı dersliklerin sayısı kadardır. Bu renklerle oluşan grafın, düğümlerini renkleme için, bu renklerle tüm düğümlere, bağlı olan düğümlere ise farklı renk koşuluyla bir renk verilmesi gerekir. Eğer bu koşulu sağlayacak şekilde verilen renklerle grafın düğümleri

renklendirilebilirse bu durumda çakışma olmayan çizelgeleme oluşturulabilir anlamına gelmektedir.

Örneğin 10 tane ders, 3 tane derslik ve 5 tane zaman periyoduna sahip bir basit çizelgeleme probleminin var olan kısıtlarını göze alarak Şekil 16'da gibi grafa dönüştürüldükten sonra bu grafi var olan renklerle renklendirdikten sonra mümkün bir çizelgesi elde edilmektedir.



Şekil 16. Graf renklendirmekle mümkün ders çizelgeleme oluşturma [26].

Literatürde, graf renklendirmek için çoklu kesin ve sezgisel algoritmalar bulunmaktadır. Bu algoritmalarından bazıları Bölüm 1.4.1 de açıklanmıştır. Graftaki düğüm sayısı arttığında graf renklendirme problemi bir NP-Zor probleme dönüşmektedir. Bu yüzden ders çizelgeleme probleminde dersler, kısıtlar ve derslikler sayısı arttığında bu yöntem çözüm için yetersiz kalmaktadır [26].

Literatürde ilk aşama (mümkün çözümler üretimi) için en başarılı sonuçları sıralı sezgisel yöntemlerden elde edilmiştir. Bu yöntemlerde dersler çizelgeye katı kısıtları sağlamak koşuluyla tek tek atanır. Atama için en zor değerlendirilebilecek derslerden başlanılır. Bu değerlendirmeler her yöntemin kendine özgü sezgisel yapıları kullanılarak yapılmaktadır. Dersler bu değerler üzerinden azalan bir şekilde sıralanırlar ve bu sırada ilk gelenler erken atanmak için öncelik kazanırlar. Bu sezgisellerden bazıları bölüm 1.4.1 de verilmiştir.

Bu yöntemler farklı sezgisellerin birleşiminin kullanmasına rağmen, mümkün çözümü bulmayı garanti etmemektedirler [58].

Kostuch [64] ITC-2002'de verilen 20 tane örnek verilerin mümkün çözümler üretimi için beş adımlı algoritma önermiş. Frausto Solis ve diğerleri [116] isi işlem yönteminden

yararlanarak ITC-2002’de 20 tane olan veriler için mümkün çözüm üreten algoritma önermişler. Mayer ve diğerleri [117] karınca koloni optimizasyon algoritmasından yararlanarak ITC-2007’nin üçüncü bölümü için mümkün çözüm üreten algoritma önermişler. Ejaz ve Younus [118] karınca davranışını ilham alan bir melez yaklaşımını geliştirerek Socha örnek probleminin mümkün çözüm üretimi için önermişler.

Lewis ve Paechter [119] gruplama genetik algoritmadan yararlanarak ITC-2002 probleminin örnek verileri için mümkün çözüm nüfusu üreten algoritma önermişler. Landa-Silva ve D, Obit [120] Socha ve ITC-2002 problemin örnek verileri için melez yaklaşım kullanarak mümkün çözümler üreten algoritma önermişler.

Chiarandini ve diğerleri [94] tarafından önerilen melez sezgisel yaklaşım ITC-2002 örnek problemi üzerinde mümkün çözüm nüfus üretimi için kullanılan üç adımlı bir algoritmadır. Bu işlemler şekil 17d’e verilmiştir. Önerilen algoritma ilk adımda önce en büyük derece (LDF) sezgiselden yararlanarak derslere LDF değeri büyük olanları sırasıyla alır. Rastgele olarak bir zaman periyodu ve derslik seçilir ve katı kısıtları sağlamak koşuluyla buraya atanır. Eğer seçilen dersliğe atanmazsa o zaman her hangi rastgele her hangi bir dersliğe atanır. Bazı durumlarda ise mümkün çözümler sadece bu adımın içerisinde üretilir ve diğer adımlara ihtiyaç kalmamaktadır.

Eğer mümkün çözüm ilk adımda üretilmezse, katı kısıtları sağlamama sayısı kadar ceza alınır ve bu durumda ikinci ve üçüncü adım mümkün çözüm üretilene kadar işleme alınır. İkinci adımda iki türlü komşu arama mekanizması alınan bu cezayı azaltmakta ve mümkün çözümler üretimi için sırasıyla kullanılmaktadır.

```

Set population size, Popsiz
Set Solution_counter←0;
Set Event_counter←0;
do while (Solution_counter < Popsiz)
  do while(Eventcounter < Number of events)
    Phase 1:Apply largest degree heuristic
  end do
  do while (timetable infeasible)
    Phase 2:Apply neighborhood search
    Phase 3:Apply tabu search
  end do
end do

```

Şekil 17. Mümkün çözümler üreten yöntemin adımları [63].

Birincisi, rastgele bir zaman periyodu ve derslik seçimi ve olan dersi rastgele olacak şekilde diğer boş olan bir zaman periyodu ve dersliğe katı kısıtları sağlamak şartıyla atamaktır. (Nbs1). İkincisi rastgele olarak iki ders seçer ve bunların dersliklerinin ve zaman periyodlarının değişimi katı kısıtları sağlanıncaya kadar sürdürülür. (Nbs2). Eğer bu adım 10 kullanımdan sonra ele alınan çözüm için gelişme sağlamazsa o zaman üçüncü adıma geçilir.

Önceki iki adım mümkün çözüm üretimi için başarısız kalınca üçüncü adım tabu arama içerisinde ikinci adımda olan iki komşuluk mekanizmasını kullanır. Eğer 1000 çevrim içerisinde mümkün çözüm bulunmazsa sonlanır ve işlemler bu çözüm için ilk adımdan tekrardan ele alınır. İstenilen sayıda mümkün çözüm üretilene kadar tüm işlemler devam eder.

Bu algoritma literatürde zaman çizelgeleme probleminin mümkün çözüm üretimi için en tercih edilen algoritmalarından birisidir. Örneğin Abdullah ve diğerleri [63] ITC-2007'nin üçüncü bölüm (CB-CTT) için önerdiği iki aşamalı algoritmanın ilk aşamasında bu algoritmadan yararlanmışlardır. Bunun nedeni 21 tane örnek veriler için mümkün çözümlerin en hızlı şekilde üretimidir.

Müller [55] ITC-2007 yarışmanın üçüncü bölümünde birinci gelen kişi mümkün çözüm üretimi için Iterative Forward Search isimli algoritmadan yararlanmıştır. Lü ve Hao [83] aynı bölümde ikinci gelen kişiler ise mümkün çözüm üretimi için fast greedy heuristic (Hızlı Açgözlü Sezgisel). algoritmadan yararlanmışlardır. Bu algoritma her ders oturumun çizelgede yerine atanması için iki sezgiselden faydalanır. Birincisi her adımda ders oturumlarının içerisinde seçmek için, ikincisi ise derslik ve zaman periyot ikiliğinin seçimi için kullanılmaktadır.

Atsuta ve diğerleri [56] aynı bölümde üçüncü gelen kişiler ise kısıt sağlama problem (CSP) yöntemiyle birlikte melez olarak tabu arama ve yerel arama yöntemlerle farklı ağırlıkta olan kısıtların (esnek ve katı) sağlanmasını ele almıştır. Geiger [121] aynı bölümde dördüncü gelen kişi ise mümkün çözümler üretimi için üç aşamalı algoritmadan yararlanmıştır. Bu algoritma da açgözlü sezgiselden faydalanmıştır. Clark ve diğerleri [122] aynı bölümde beşinci gelen kişiler ise tek aşamada her iki kısıtı ele almışlar ve sadece katı kısıtların cezası esneklere göre daha fazla tutmuşlar.

2.3.3.2. İlk Aşama İçin Önerilen Yöntem

Önerilen yaklaşımın ilk aşamasında ABC den yararlanarak, istenilen sayıda mümkün çözümler üretilir. ABC den faydalanmak için CB-CTT problemi üzerinde mümkün çözüm üretimi, bir amaç fonksiyonuna sahip optimizasyon problemi haline dönüşür ve sonra elde edilen problemin amaç fonksiyon değeri ABC algoritmasıyla minimum veya maksimum yapılır. Çözümde ise her dört katı kısıtın (H1, H2, H3, H4) sağlanması gerekmektedir.

Bunların içerisinde diğerlerine göre zor sağlanan çakışma (H3) kısıtıdır. Çözümler bir matris ile ele alındığından dolayı her zaman H2 kısıtı kendi kendine sağlanmaktadır. Çünkü matrisin her bir hanesi birden fazla değer alamaz, bu her zaman periyot içerisinde bir derslikte sadece bir ders atanmaya karşı gelmektedir. Önceden belirlenen $SN1$ (ilk aşamada ABC'nin yiyecek kaynak sayısı) sayısı kadar başlangıç çözümler üretilmelidir.

Başlangıç çözümlerin üretimi için dersler listesinden rastgele birisi seçilir sonra bu dersin oturum sayısı kadar rastgele bir gün, gün içerisinde bir zaman dilimi ve derslik seçilir. Eğer çözümde bu hane boş ise ve dersin oturumunun bu haneye atanması dört katı kısıttan ikisini yani H1, H4 sağlarsa, H3 katı kısıtın sağlanmasına bakmadan dersin indeksi buraya atanır. Eğer bu yer bu ders oturumu için, koşulları sağlamazsa rastgele diğer bir yer seçilir. Bu işlem her dersin oturumu için maksimum 100 kere yapılabilir. Maksimum seçme sayısına ulaşmadan atanabilirse o zaman bu işlem diğer oturma ve dersler listesinden rastgele seçilen diğer ders için de yapılır yoksa tüm çözüm sıfırlanır yani atanmışlar silinir ve -1 değeri çözümün her hanesine atanır ve işleme en baştan tekrar başlanır.

Tüm derslerin tüm oturumları atanana kadar bu işlem devam eder. Tüm dersler atandığı zaman ABC için bir başlangıç çözüm (yiyecek kaynağı) üretilir. $SN1$ sayısı kadar başlangıç çözüm üretildikten sonra her çözüm için amaç fonksiyon değeri çözümde H3 katı kısıtı sağlamayan ders oturumlarının atanmış yer sayısı alınmaktadır.

İlk aşama ABC den yararlanarak, tanımlanmış amaç fonksiyon değeri sıfır olan bir çözüme ulaşmaktır. Böylelikle her dört katı kısıtı sağlayan bir çözüm bulunur. Burada komşu üretme mekanizması temel ABC algoritmasına uyarlayabilmek için rastgele optimizasyon problemin parametresi burada varsayılan derslik indeksi ve rastgele diğer çözümler içerisinde birisi hedef çözüm olarak seçilir. Burada amaç kaynak çözümünde ele alınan derslikte rastgele bir gün ve zaman diliminden başlayarak tüm hane değerlerinin dört katı kısıtı bozmama koşuluyla hedef çözüm gibi yapmaktır. Bunun için kaynak çözüm

içerisinden diğer yerlerde hedef kaynakta olan ders oturumunun aynısını bulup katı kısıtları bozmama koşuluyla ele alınan hanedeki değerle yer değişimi yapılır. Eğer bu dersin kaynak çözüm içerisinde olan tüm oturumları değişim yapmak için test edilip ve değişim yapılmazsa o zaman kaynak çözümde olan değer sabit kalır ve benzetim işlemi sırada olan diğer haneden devam eder. Temel ABC algoritmasından farklı olarak kaşif arılar gelişmeme limitini aşan kaynağın yenisiyle değişim yapmadan sadece kaynak içerisinde maksimum 30% yer değişimi yapılır. Bu değişim maksimum 1000 kere rastgele iki hane seçimi ve bu hanelerin değerleri dört katı kısıtı sağlamak koşuluyla yapılır. Bu işlemler *SN2* (ikinci aşama için ABC'nin yiyecek kaynak sayısı) tane mümkün çözümler üretene kadar devam eder. Eğer her mümkün kaynağın üretiminde ABC döngüsü 2000 kere çevrilirse ve amaç fonksiyonu sıfır olan çözüm bulunmasa o zaman işlem baştan, *SN1* tane başlangıç çözüm üretiminden tekrardan başlatılır.

İlk aşamanın *SN2* tane mümkün (dört katı kısıtı sağlayan) çözümler ürettikten sonra her çözüm için yeni amaç fonksiyon değeri dört tane esnek kısıtların (*S1*, *S2*, *S3*, *S4*) sağlanmaması için alınan ceza değerlerinin toplamı (11) olur.

Başlangıç çözümleri saklamak için,

Foods_initial $SN1 \times Days_Number \times Periods_per_day_Number \times Rooms_Number$ matrisi tanımlanır ve ilk değer olarak tüm hanelerine -1 değeri atanır. Mümkün çözümleri saklamak için ise

Foods $SN2 \times Days_Number \times Periods_per_day_Number \times Rooms_Number$ matrisi tanımlanır ve aynen ilk değer olarak tüm hanelerine -1 değeri atanır.

Her bir başlangıç çözümün üretiminde başlangıçta *Room_of_course*, *Prof_time*, *Group_time* matrislerin sırasıyla -1, 0 ve 0 la değerleri yerleşilir. Dersler rastgele olarak önceden seçilmeme koşuluyla *COURSES* dizisinden seçilir. Bu dizide belirlenen oturum sayısı kadar (Lectures) çözüm matrisinde rastgele bir yer (gün, saat, derslik) seçilir ve bu yer *H1*, *H2*, *H4* sağlamak için sırasıyla:

- *Room_of_course* gün, saat, seçilen ders indeksi değeri -1,
- çözüm matrisinde seçilen yerin değeri -1,
- *UNAVAILABILITY_CONSTRAINTS* gün, saat, seçilen ders indeksi değeri 0 olmalıdır.

Bu koşullar sağlandığı zaman dersin indeks değeri çözümde seçilen yere atanır ve bunun üzerinde;

- *Room_of_course* gün, saat, seçilen ders indeksi dersin indeksin değeri olarak alır,
- *Prof_time* gün, saat, seçilen dersin →Prof_ID ,
- *Group_time* gün, saat, seçilen dersin →var olan müfredat değerleri de bir artar.

Her ders oturumu için maksimum 100 kere H1, H2, H4 kısıtların sağlamsı için rastgele bir yer seçilir, eğer bu çevrim içerisinde bir yer bulunmazsa çözüm, *Room_of_course*, *Prof_time*, *Group_time* matrislerin sırasıyla -1, 0 ve 0 la değerleri yeniden yerleştirilir ve işleme yeniden devam edilir. Tüm derslerin oturumları H1, H2, H4 sağlayarak çözüme atanması bir başlangıç çözüm üretir. Her ders oturumunun maksimum 100 kere seçme sınırı vardır bu sınır atama işleminde zaman tasarruf için belirlenmiştir.

- 1) SN1 (İlk aşama için ABC kaynak sayısı) ve limit1 parametrelerin belirle ve örnek dosyanı oku ve D (derslikler sayısı) optimizasyon parametresini belirle.
- 2) Repeat SN1 times
 - a) Boş bir çözüm
 $X_{\text{Eğitim gün sayısı} \times \text{Zaman dilimler sayısı} \times \text{Derslik sayısı}}$ matrisi üret ve tüm hücrelerine -1 değerini at.
 - b) Repeat for Courses_Number.
 - i. Seçilmemiş derslerin içerisinde birini rastgele seç. C_i
 - ii. Repeat for seçilen dersin oturumlarına (max 100 times).
 1. Rastgele (gün, zaman dilimi, derslik) üçlüğünü üret.
 2. Eğer C_i nin bu hücreye atanması, çakışma katı kısıtı hariç diğerlerin sağlarsa, C_i atansın ve döngeden çık
 3. Eğer bu döngü 100 çevrimi aşarsa 2-a aşamasına dön.
- 3) Her çözüm için maliyet fonksiyonu çakışma kısıtı sağlanmayan oturumların sayısını hesapla ve çözümler için kayıt et.

Şekil 18. İlk aşama için SN1 tane başlangıç çözüm üretimi

Her çözüm için maliyet fonksiyonu değeri çözümde H3 sağlamayan ders oturumları sayısı kadar hesaplanır. Bu işlemler şekil18'de verilmektedir.

ABC'nin yiyecek kaynakları üretilenden sonra amacı her çevrimde işçi ve gözcü arı gönderme adımında çözümlerin amaç fonksiyonunda aldığı değeri azaltmak ve 2000 çevrim içerisinde en iyi çözüm maliyet değerini (GlobalMin) sıfıra ulaştırmak ve böylelikle tüm katı kısıtları sağlayan bir çözüm üretmektir.

İlk aşamada komşu üretme mekanizması üst bölümde anlatıldığı gibi her çözüm için rastgele olarak diğer çözümler içerisinde birisini hedef çözüm olarak seçer ve sonra rastgele optimizasyon parametresi olarak tanımlanan derslikler arasından birisini seçer. Ele alınan çözümde derslik üzerinden rastgele bir zaman periyodundan başlayarak içeren değeri (boş veya ders oturum) katı kısıtları bozmadan aynen hedef çözümde aynı derslikte ve aynı zaman periyotda olan değer gibi yapmaktır.

Bu işlemi gerçekleştirmek için kaydırma ve değişme işlemleri kullanılır. Başlangıçta *Room_of_course*, *Prof_time*, *Group_time* matrisleri baştaki değerleri verilir (sıfırlanır) sonra kaynak çözüm üzerinden değerleri güncelleştirilir. Eğer seçilen derslik üzerinde ele alınan zaman periyodunda kaynak çözüm matrisin hanesinde olan değer x ve hedef çözümde olan y varsayılırsa, benzetim yapılırken dört farklı durumla karşılaşılır:

1. $x == y$ (hanelerin içeriliği aynı)

Birinci durumda hiçbir işlem yapmadan, işlem diğer zaman periyodundan devam edilir.

2. $x == -1$ ve $y \neq -1$ (hedef çözümde bu hane boş hâlbuki kaynak çözümde bu hanede bir ders oturumu var)

İkinci durum için kaynak çözümün tümünde rastgele bir yerden (gün, saat, derslik) başlayarak boş yer aranılır, bir boş yer bulunduğu *Room_of_course*, *Prof_time*, *Group_time* matrisleri üzerinde y 'in oraya taşınması katı kısıtları bozup ve ya sağladığı araştırılır. Eyer katı kısıtlar sağlanmışsa y değeri çözümde bulunan yere ve y 'in eski yerine -1 değeri gelir ve ek olarak bu değişim *Room_of_course*, *Prof_time*, *Group_time* da güncelleşmeye neden olur. Tüm boş yerler denetildiği ve değişim yapı bilmediği zam y eski yerinde kalır.

3. $x \neq -1$ ve $y == -1$ (hedef çözümde bu hanede bir ders oturumu var hâlbuki kaynak çözümde bu hane boştur)

Üçüncü durum için *Room_of_course* matris üzerinden ilk aynı periyot üzerinde kaynak çözümde x dersin bir oturum olması araştırılır. Eğer bir oturum varsa

bulunan derslikte ve y olduğu derslikle katı kısıtlara bakmadan (aynı periyotda olduğundan dolayı önceden sağlanmış) değişim yapılır ve sadece *Room_of_course* metrisi güncelleşir, eğer diğer bir periyotda olursa bulunan yerle y olan yerler taşınması katı kısıtları sadece x için matrisler üzerinden bakıldığında sağlanırsa yer değişimi yapılır ve matrisler güncelleşir. Tüm x oturumlar kaydırmak için denetilir ve hiç biri katı kısıtları sağlamama durumunda y eski haliyle işleme devam edilir.

4. $x \neq -1$ ve $y \neq -1$ (her iki çözümde aynı zaman periyodunda derslikte farklı ders oturumları vardır)

Dördüncü durumda aynen önceki durum gibi *Room_of_course* matris üzerinden ilk aynı periyot üzerinde kaynak çözümde x dersin bir oturum olması araştırılır. Eğer bir oturum varsa bulunan derslikte ve y olduğu derslikle katı kısıtlara bakmadan (aynı periyotda olduğundan dolayı önceden sağlanmış) değişim yapılır ve sadece *Room_of_course* metrisi güncelleşir. Eğer diğer bir periyotda varsa bulunan yerle y olan yerlerin taşınması katı kısıtları matrisler üzerinden bakıldığında sağlarsa ve aynen y 'nin x yerine taşınması katı kısıtları sağlarsa değiştirme işlemi yapılır. Bu koşulları sağlayan x oturumu varsa değişim yapılır ve matrisler güncelleşir. Yoksa y eski yerinde işleme devam eder.

Bu aşamada anlatılan komşuluk yapısı ABC'nin işçi ve gözcü arılar tarafından ele alınır. Her çözüm için bir komşuluk üretildiğinde *Room_of_course*, *Prof_time*, *Group_time* matrisleri değerleri bu çözüm üzerinden güncelleştirirler ve komşuluk üretildikten sonra komşuluğun maliyet fonksiyon değeri (H3 sağlamamak sayısı) hesaplanır. Eğer bu değer çözümün kendi değerinden küçükse komşuluk çözümle yer değişir ve gelişeme sayacı sıfırlanır yoksa gelişeme sayacı bir artar.

Eğer 2000 çevrim içerisinde en iyi maliyet fonksiyon değeri (GlobalMin), sıfıra ulaşırsa döngü biter ve böylelikle katı kısıtları sağlayan çözüm bulunmuş olur. Yoksa bir mümkün çözüm üretimi için işleme yeniden başlanır, yani çözüm başlangıç değeri (-1) alınır ve başlangıç çözümler yeniden üretilir.

Temel ABC den farklı olarak kaşif arılar adımında, gelişeme sayacı limiti aşan yiyecek kaynağının yenisiyle değişim yapılmamaktadır. Sadece kaynak içerisinde

maksimum 30% yer deęişimi yapılır. Bu deęişim maksimum 1000 kere rastgele iki hane seçimi ve bu hanelerin deęerleri dört katı kısıtları sağlamak koşuluyla yapılır. Deęişim yapıldıktan sonra kaynağın gelişmeme sayacı sıfırlanır. Bu bölümde anlatılan işlemler Şekil 19’da verilmektedir.

1. *SN1* tane başlangıç çözüm üret ve maliyet deęerlerini hesapla
Repeat 2000 times or GlobalMin == 0
 - a. Görevli işçi arılar aşaması:
Repeat *SN1* times
 - i. Her çözüm için bir komşu üret
 - ii. Eğer komşuluğun maliyet deęeri ele alınan çözümden küçükse bunların yerini deęiştir ve çözümün geliştirememe sayacını sıfırla. Yoksa çözümün geliştirememe sayacını bir artır.
 - b. Alınan maliyet fonksiyonu deęerlerinin üzerinde her çözümün olasılık deęerini hesapla. $\left(p_i = \frac{fitness_i}{\sum_{i=1}^{SN1} fitness_i} \right)$
 - c. Gözcü arılar aşaması:
Repeat *SN1* times
 - i. Rulet tekerleęi kullanarak çözümlerin aldığı olasılık deęerleri üzerinden birisini seç.
 - ii. Her çözüm için bir komşu üret.
 - iii. Eğer komşuluğun maliyet deęeri ele alınan çözümden küçükse bunların yerini deęiştir ve çözümün geliştirememe sayacını sıfırla. Yoksa çözümün geliştirememe sayacın bir artır.
 - d. Kâşif arı aşaması: Geliştirememe sayacı en fazla limiti aşan kaynağın 30% içeriliğın katı kısıtları sağlayacak koşuluyla deęiştir ve bu sayacı sıfırla.
 - e. Buraya kadar en iyi çözümü hafızaya al ve maliyet deęerini GlobalMin’e aktar.
2. Eğer GlobalMin deęeri sıfırsa ona ait çözümü mümkün çözüm olarak kaydet ve esnek kısıtları sağlaması için alınan toplam cezayı hesapla ve çözüm için kaydet. Yoksa 1’e dön.

Şekil 19. İlk aşama için ABC algoritmasının kullanımı

Burada rastgele noktadan benzetim işlemine başlama yöntemi meta sezgisel yöntemlerin temel felsefesi yani rastgeleliğe uyumunu sağlamak içindir. Bu komşuluk

yapısı, üniversite ders çizelgeleme probleminin aday çözümleri üzerinde yeni bir bakış açısı sayılır.

Literatürde bulunan komşuluk yapıları rastgele olarak kaydırma veya değişme işlemini kullanırlar veya N3 v3 N4 gibi iki periyodu ele alırlar. Bu yapı bir dersliği ele alarak kaydırma ve değişme işlemlerini kullanılmaktadır.

2.3.4. İkinci Aşama

2.3.4.1. İkinci Aşama Literatür Analizleri

İkinci aşamada ilk aşamanın çıktısı olan mümkün çözümleri ele alarak, katı kısıtları bozmamak koşuluyla esnek kısıtları sağlamaya çalışılır. Literatürde mümkün çözüm için, katı kısıtları bozmadan değişiklik yapma işlemine komşu arama (Neighborhood Search) yada yerel arama (Local Search) denilir. Kısıt sağlama ve optimizasyon problemlerinin çözümü için kullanılan meta sezgisel algoritmaların son derece etkili olduğu ispatlanmıştır. Genelde komşu arama mekanizmasının tanıtımı yerel aramalarda çok etkileyicidir. Güçlü komşu arama mekanizmaları başlangıç çözümden bağımsız olarak, iyi arama yeteneğine göre sürekli çözümün iyileşme yönünde gelişmesini sağlarlar. Halbuki zayıf komşu arama mekanizmalarının başlangıç çözümüne bağlantısı fazla olur [114].

Literatürde ikinci aşama için bulunan en iyi yaklaşımlardan birisi ITC-2002 problemi üzerinde yapılan çalışmadır [61]. Bunda ikinci aşama için esnek kısıtların sağlanmasında N1, N2 komşuluk yapılarını kullanan tabu arama algoritmasından yararlanmıştır. Benzer bir çalışma [95], ikinci aşama için Great Deluge [90] algoritmasından yararlanmıştır. Her iki çalışmada çözüm üzerinde arama, her zaman mümkün uzayda yapılmaktadır. Socha ve diğerlerinin [21, 74] tanımladığı örnek problem üzerinde esnek kısıtların sağlanması için karınca koloni algoritmasından yararlanmışlar.

Chiarandini ve diğerleri [94] ITC-2002 problemin üzerinde yaptıkları çalışmada ilk aşamada katı kısıtları ele alarak, bunların sağlanmasıyla mümkün çözümler ürettikten sonra ikinci aşamada esnek kısıtların sağlanması için önce değişken komşu arama [84] sonra ise ısı işlem algoritmasından yararlanmışlardır. ITC-2002 yarışmasında birinci olan Kostuch [64] ikinci aşamada esnek kısıtların sağlanması için ısı işleme yönteminden yararlanmıştır. Obit [7] tarafından geliştirilen yeni Great Deluge algoritmasını Socha ve ITC-2002 problemlerinin üzerinde esnek kısıtların sağlanması amacıyla kullanmıştır. Lewis [26]

Socha ve ITC-2002 problemlerinin çözümü için önerdiği yaklaşımda, esnek kısıtların sağlanması için iki farklı yöntemden (evrimsel algoritma, ısı işleme) faydalanmıştır.

Müller [55] CB-CTT probleminin çözümü için önerdiği yaklaşımda mümkün çözümler ürettikten sonra esnek kısıtların sağlanması için bir döngü içerisinde üç adımlı bir yaklaşım kullanmıştır. İlk adımda; tepe tırmanma (Hill Climbing) algoritmasından yararlanmıştır. Bu adım, önceden belirlenen sayıda gelişme göstermediğinde biter ve ikinci adımda Great Deluge algoritmasından faydalanır. Bu adım belli koşulları sağlamayınca sonlanır ve işlemin devamında üçüncü adım ele alınır. Üçüncü adımda; ısı işlem algoritma kullanılır. Belli koşula altında işlem tekrar birinci adıma döner. Bu döngü belli kısıtları sağlayınca biter. Bu çalışmada Müller 6 farklı komşu arama mekanizma tanımlamış ve komşuluk arama için her seferinde rastgele birisi seçilmektedir. Lu ve Hao [83] CB-CTT problemin çözümü için ikinci aşamada esnek kısıtların sağlanması için uyarlamalı tabu arama algoritmasından (ATS) faydalanmışlar.

Geiger [120] CB-CTT problemin çözümü için bu aşamada eşik olan meta sezgisel yöntemden faydalanmıştır. Komşu aramada hareketler rastgele seçilmektedir. Abdullah ve diğerlerinin [63] 2010 yaptığı çalışmada önerdiği yaklaşımları ITC-2002, Socha ve ITC-2007 CB-CTT probleminin çözümü için kullanmışlardır. Bu çalışma iki aşamalı sınıf çalışmalarından, ikinci aşamada esnek kısıtların sağlanması için popülasyon tabanlı yeni geliştirilen meta sezgisel yöntemlerden birisi olan Electromagnetic-like Mechanism (EM) [123] ve Great Deluge [90] algoritmalarından faydalanmışlardır. Her iki algoritmada kullanılan komşu arama yapısı N1 ve N2 dir.

2.3.4.2. İkinci Aşama İçin Önerilen Yöntem

İkinci aşamada esnek kısıtların sağlanması için ABC algoritmasıyla birlikte yerel arama işleme ele alınır. Amaç; ilk aşamadan gelen çözümleri başlangıç çözüm olarak ele alınması ve ilk aşamada harcanan süre dâhil olmak üzere belirlenen zaman içerisinde katı kısıtları bozmadan esnek kısıtların mümkün olduğu kadar sağlanması ve alınan cezayı minimum yapmaktır.

Bu aşamada komşu arama mekanizması, ilk aşamada tanımlayan gibidir. Ek olarak her çözümün ilk ceza değeri kayda alınır. Eğer seçilen dersliğin zaman periyotlarında olan ders oturumlarının benzetiminde hareket N1 ise kaydırma işlemi kayıta alınan değeri

arttırmama koşuluyla yapılabilir. Eğer hareket N2 ise değiştirme işlemi kayıta alınan değeri azaltma koşuluyla yapılmaktadır.

İşçi arı adımında temel ABC algoritmasından farklı olarak her çözümün komşuluğunun üretilmesinden sonra, bunun üzerinde yerel arama yapılır. Yerel arama işleminde bir 50 çevrimli döngü içerisinde çözüm matrisinden rastgele iki hane (gün, gün içerisinde zaman dilimi, derslik) seçilir ve bunların değerleri katı kısıtları sağlamak ve cezaları azaltmak koşuluyla yer değiştirilir ve ilk değişim döngüsü sonlandırılır.

Bu aşamada maliyet fonksiyon değeri ilkinden farklı olarak CB-CTT problemi üzerinden esnek kısıtların her birisinin sağlanmaması durumunda belirlenen cezaların, toplamıdır ve (11) hesaplanması gerekmektedir.

ABC algoritmasının başlangıç çözümleri için ilk aşamanın çıktısı olan *SN2* tane mümkün çözümler kullanılmaktadır. İşçi ve gözcü arı tarafından alınan çözümler için komşuluk üretilir. Burada tanımlanan komşuluk yapısı ilk aşamadan farklıdır.

Komşuluk yapısı aynen ilk aşama gibidir. Farklı olan yeri ise, benzetim işleminde sadece katı kısıtların sağlanması değil belki esnek kısıtların da dikkate alınmasıdır. Her çözüm için *Room_of_course*, *Prof_time*, *Group_time* matrislerinin kaynak çözüm üzerinden değerleri güncelleştirilir ve ilk maliyet değeri de kayıt olur. Benzetim işleminde dört farklı durumda yapılan işlemler şöyledir:

1. $x == y$ (hanelerin içeriliği aynı)

Hiçbir işlem yapmadan, benzetime diğer zaman periyodundan devam edilir.

2. $x == -1$ ve $y \neq -1$ (hedef çözümde bu hane boş. Hâlbuki kaynak çözümde bu hanede bir ders oturumu var)

Kaynak çözümün tümünde rastgele bir yerden (gün, saat, derslik) başlayarak boş yer aranır. Elbet boş yer bulunan derslikte önceden y dersinin bir oturumu olmalıdır, bu koşulu sağlayan bir boş yer bulunduğu *Room_of_course*, *Prof_time*, *Group_time* matrisleri üzerinde y 'in oraya taşınmasında katı kısıtları bozup bozmadığı veya sağlayıp sağlamadığı araştırılır. Eğer katı kısıtların sağlanmasına göre sorun yoksa ve bu kaydırma çözümün maliyet fonksiyon değerinin ilkinden fazla olursa, diğer boş yer aranır, yoksa kaydırma yapılır ve y değeri çözümde bulunan yere ve y 'in eski yerine -1 değeri gelir ve ek olarak bu değişim *Room_of_course*, *Prof_time*, *Group_time* da

güncelleştirilir. Tüm boş yerler denetildiği ve bu koşulları sağlayan değer bulunmadığında, y eski yerinde kalır.

3. $x \neq -1$ ve $y == -1$ (hedef çözümde bu hanede bir ders oturumu var hâlbuki kaynak çözümde bu hane boştur)

Room_of_course matrisi üzerinden ilk aynı periyot üzerinde kaynak çözümde x dersinin bir oturumunun olup olmadığı araştırılır. Eğer x dersinin bir oturumu varsa oturumun bulunduğu derslikte ve y dersinin olduğu derslikle katı kısıtların sağlanmasına bakmadan (aynı periyotda olduğundan dolayı önceden sağlanmış) eğer bu kaydırma çözümün maliyet fonksiyon değerini ilkinden fazla etmezse değişim yapılır ve çözüm geliştirememeye sayacı sıfırlanır. Sadece *Room_of_course* matrisi güncelleştirilir. Eğer x oturumu diğer bir periyotta ise bulunan yerle y oturumu olan yerlerin taşınması için sadece katı kısıtlar x için (üç matris üzerinden) sağlanırsa ve aynen önceki gibi kaydırma çözümün maliyet fonksiyon değerini ilkinden fazla etmese değişim yapılır ve matrisler güncelleştirilir. Tüm x oturumları kaydırmak için denetlenip ve istenilen koşulları sağlamama durumunda y önceden bulunduğu yerde işlem devam eder.

4. $x \neq -1$ ve $y \neq -1$ (her iki çözümde aynı zaman periyodunda aynı derslikte farklı ders oturumları vardır)

Room_of_course matrisi üzerinden ilk olarak aynı periyot üzerinde kaynak çözümde x dersinin bir oturumu olup olmadığı araştırılır. Eğer bir oturum varsa bulunan derslikte ve y oturumunun olduğu derslikle katı kısıtların bozulmasına bakmadan (aynı periyotda olduğundan dolayı önceden sağlanmış bu değişim sadece çözümün maliyet fonksiyon değerini iyileştirecekse yapılır ve *Room_of_course* matrisi güncelleştirilir ve *Room_of_course* matrisi güncelleştirilir. Eğer x oturumu diğer bir periyotta bulunursa x oturumunun olduğu yerden y 'nin olduğu yere taşınması katı kısıtları matrisler üzerinden bakıldığında sağlarsa ve aynen y 'nin x 'in yerine taşınması katı kısıtları sağlarsa ve bu değişim çözümün maliyet fonksiyon değerini iyileştirecekse yapılır. Bu koşulları sağlayan x oturumu varsa değişim yapılır ve matrisler güncelleştirilir, yoksa y eski yerinde işleme devam eder.

Bu aşamada üstte anlatılan komşuluk yapısı ABC'nin işçi ve gözcü arıları tarafından ele alınır. Her çözüm için bir komşuluk üretildikten sonra sadece işçi arılar tarafından bu

komşuluk üzerinde yerel arama yapılır sonra güncelleşen *Room_of_course*, *Prof_time*, *Group_time* matris değerleri üzerinden komşuluğun maliyet fonksiyon değeri (11) hesaplanır, eğer bu değer çözümün kendi değerinden küçükse komşuluk, kaynak çözümle yer değiştirir ve gelişmeme sayacı sıfırlanır yoksa gelişmeme sayacı bir arttırılır.

Kâşif arı adımında temel ABC algoritmasından farklı olarak geliştiremememe sayacı limitini aşan yiyecek kaynağı için yeni rastgele yiyecek kaynağı üretilmemektedir; burada iki farklı durum vardır. Eğer seçilen kaynak buraya kadar en minimum cezaya sahip kaynak ise 1000 çevrimli bir döngü içerisinde maksimum 30% yer değişimi yapılır. Bu yer değişimi için döngüye girmeden ele alınan çözümün ceza değeri kayıta alınır sonra döngü içerisinde rastgele olarak çözüm matrisinden iki hane seçilip bu hane içeriklerinin yer değişimi, katı kısıtları bozmama ayrıca S1 ve S2 cezanı arttırmayacak şekilde eğer toplam ceza kayıta alınan cezadan fazla değilse yapılabilir.

Diğer durumda ele alınan kaynak buraya kadar en iyi kaynak değilse, 1000 çevrimli bir döngü içerisinde maksimum çözüm içerisinde 30% yer değişim yapılır. Bu yer değişimi için döngü içerisinde rastgele olarak çözüm matrisinden iki hane seçilip bu hane içeriklerinin yer değişimi, katı kısıtları bozmama, S1 ve S2 cezasını arttırmayacak şekilde yapılabilir

ABC algoritması döngü içerisinde koşullar sağlanana kadar işçi arı, gözcü arı ve kâşif arı adımlarını çözümler üzerinde uygular ve her döngü sonunda en iyi maliyet değerine sahip çözümü kayıt eder. Anlatılan işlemler şekil 20'de verilmektedir.

1. *SN2* tane başlangıç çözümünü ilk aşamadan al.
 2. Her çözüm için maliyet fonksiyon değerini, esnek kısıtların sağlanması durumunda alınan toplam cezayı hesapla.
- Repeat harcanan zaman \leq belirlene zaman
- a. Görevli arılar aşaması:

Repeat *SN2* times

 - iii. Her çözüm için bir komşu üret.
 - iv. Çözüm üzerinde yerel arama yap.
 - v. Eğer komşuluğun maliyet değeri ele alınan çözümden küçükse bunların yerini değiştir ve çözümün geliştiremeye sayacını sıfırla, yoksa çözümün geliştiremeye sayacını bir artır.
 - f. Alınan maliyet fonksiyon değerlerinin üzerinde her çözümün olasılık değeri hesapla. $\left(p_i = \frac{fitness_i}{\sum_{i=1}^{SN2} fitness_i} \right)$
 - g. Gözcü arılar aşaması:

Repeat *SN2* (gözcü arılar sayısı) times

 - iv. Rulet tekerleği kullanarak çözümlerin aldığı olasılık değerler üzerinden birisini seç.
 - v. Her çözüm için bir komşu üret.
 - vi. Eğer komşuluğun maliyet değeri ele alınan çözümden küçükse bunların yerini değiştir ve çözümün geliştiremeye sayacını sıfırla. Değilse çözümün geliştiremeye sayacını bir artır.
 - h. Kâşif arı aşaması: Geliştiremeye sayacı limiti2 'yi daha fazla geçen kaynağı seç. Eğer seçilen çözüm en iyi maliyet fonksiyon değerine sahip olursa, içeriğin %30'unu katı kısıtları sağlayacak ve maliyet değeri artırmama koşuluyla değiştir, yoksa içeriğin %30'unu katı kısıtları sağlayacak ve S1 ve S2 cezasını artırmama koşuluyla değiştir ve sayacı sıfırla.
 - i. Buraya kadar en kaliteli çözümü hafızaya al ve maliyet değerini GlobalMin'e aktar.

Şekil 20. İkinci aşama için ABC'nin uygulaması

3. BULGULAR VE SONUÇLAR

Bu yarışmada kullanılan bilgisayar ve işletim sisteminin özeliğine bağlı önerilen yaklaşımların örnek problemler üzerinde deneme yapılması için zaman sınırı belirlenmektedir. Yarışmanın resmi sitesin [28]'de benchmarking bölümü bulunmaktadır. Bu bölümde üç farklı uygulama, işletim sistemlerinin türlerine bağlı olarak bulunmaktadır ve bu uygulamalar deneme yapılan bilgisayarda çalıştığında deneme yapılması için geçerli zamanı belirtir.

Önerilen yaklaşımı CB-CTT probleminin örnek veriler üzerinde denemesi için Windows 7 işletim sistemi üzerinde, Intel(R) Core(TM) 2 Duo 1.8 GHz hızla işlemci ve 2GB belleğe sahip bilgisayarda kullanılmaktadır. Bu özeliğe sahip bilgisayarda [28]'den indirilen *benchmark_my_windows_machine.exe* uygulamasını çalıştırdığında zaman sınırı 468 saniye belirlendi.

Yaklaşımında her iki aşamada ABC parametreleri olan *limit* ve *SN* (yiycek kaynaklar sayısı) belirlenmesi için denemeler yapılmış ve tüm örnekler üzerinde 468 saniye içerisinde en iyi sonuçlar *limit* = 15 ve *SN* = 8 elde edilmiştir. Bu parametreler her iki aşamanın kullandığı ABC algoritması için aynıdır.

Bu çalışmada N3 ve N4 komşuluk üretim mekanizmalarından kullanılmama nedeni bu mekanizmaların uygulanması için faza işleme ihtiyaç olması ve bunun dolayı harcanana zamanın fazla olmasıdır. Bu mekanizmalar tek çözüm tabanlı meta sezgisel algoritmalar için örneği Tabu aram iyi sonuçlar gösterse de popülasyon tabanlı meta sezgisel algoritmalarda daha fazala işleme ve harcanan zamana neden olmaktadır.

3.1. Önerilen Yaklaşımının comp15.ctt Problemin Veriler Üzerinde Denemesi

Örnek olarak önerilen yaklaşımı comp15.ctt örnek problem üzerinde uygulandığında ilk aşamada bir mümkün çözüm üretimi için 8'tane başlangıç çözümü üretilir. Üretilen başlangıç çözümlerin en kalitelisi Tablo 12'de verilmektedir.

Tablo 12. Birinci aşamanın en kaliteli başlangıç çözümü (Z.P zaman periyod)

Z.P	Derslikler															
	r36	r37	r38	r31	r27	rB	rD	rE	rF	rG	rA	rL	r50	r52	rDS1	rDS2
0	17	22	29		68	14	38	8		18	33	58		21		7
1	29	12		38	23	43	33			5	46	14	68		45	61
2		17	61	41	55		37								6	
3	24	11	66	6	70			58	47	51			31		56	40
4	70		15		65		71	56	63	32	44		47	46		4
5	35	39	17	7	32			70	23	34		5		67	69	9
6	14	12		28	23	13	61	10			27	5		60	39	42
7										39	21		16	62	53	11
8	62	2	69	34	28		52	15		51		50	44			
9	11	70					39			71	2	40	65			57
10	29	23	51	24	3	20	39	66	30		7			64	52	63
11	38	71	50			35	3		64	0	20	9	40	68	60	59
12	69		41	65		1	35			38	19					44
13	57	71	60	69	15	40	47		64	46	66	48	3			
14		26		53				49	65	0	22		40	61	70	27
15	36	2		1	48	52		64			59	65	7	28	61	
16	23	16	63			33	38	12	55	50	4		19	6	8	
17	16		8			66	27		30	64		45	62		60	38
18	68	4		34	49	13		21	67		39	9	40		43	48
19	40		66	42	54	53	55	36		58			59			
20	37	54	18	19	29		20	14	7	22	13	53				
21		67		60	29		10		56	0	41	14			45	
22			57		49	39		25			43			36		
23		37				69			38	53	71	26	24		54	
24	10	1		18			26	32		42		68				

Bu çözümde ders oturumlarının atanmasında dört katı kısıttan *H3* hariç diğerleri sağlanmaktadır. Çözümün maliyet fonksiyon değeri çözüm üzerinde ders oturumları *H3* sağlanmama sayısı olarak hesaplanmaktadır. Bu aday çözüm [28]'de istenilen biçimde bir comp15.out text dosyasında kayıt edilip ve [28]'de bulunan validator.exe uygulanmadan yararlanarak Şekil 21'de verildiği gibi değerlendirilmiştir.

Violations of Lectures (hard) : 0
 Violations of Conflicts (hard) : 85
 Violations of Availability (hard) : 0
 Violations of RoomOccupation (hard) : 0
 Cost of RoomCapacity (soft) : 5609
 Cost of MinWorkingDays (soft) : 180
 Cost of CurriculumCompactness (soft) : 734
 Cost of RoomStability (soft) : 155

Summary: Violations = 85, Total Cost = 6678

Şekil 21. En kaliteli başlangıç çözümün değerlendirilmesi

Bu değerlendirilmede görüldüğü gibi aday çözümde Lectures (H1), Availability(H4) ve RoomOccupation (H2) katı kısıtların hepsi sağlanmıştır. Sadece conflict (H3) katı kısıtı aday çözüm üzerinde 85 kere ders oturumların atanmasında sağlanmamıştır.

Örneğin 0 ve 9 indeksine sahip ders oturumları aynı müfredatta (q004) olduğundan dolayı 11'inci zaman periyodunda birlikte atanamazlar. Aynen 0 ve 10 indeksine sahip ders oturumları aynı müfredatta (q008) olduğundan dolayı 11'inci zaman periyodunda birlikte atanamazlar ve bu atanmalar H3 katı kısıtını bozmaktadır.

İlk aşamanın ABC algoritması 8 tane başlangıç çözümleri ele alıp aday çözümler üzerinde H3 katı kısıtının sağlanmasına çalışır. Bu katı kısıtın üzerinden tanımlanan maliyet fonksiyon değeri sıfıra ulaşınca bir mümkün aday çözüm üretilmektedir. İlk aşamanın sonunda 8 tane mümkün aday çözümler üretilir. Sonra her mümkün aday çözüm ikinci aşamada tanımlanan maliyet fonksiyon üzerinden (11) değer verilir.

Bu mümkün aday çözümlerin en kalitelisi (maliyet fonksiyon üzerinden aldığı en az değere sahip) Tablo 13'de verilmektedir. Bu aday çözüm [28]'de istenilen biçimde bir text dosyasında kaydedilip [28]'de bulunan validator.exe uygulamasından yararlanarak Şekil 22'de verildiği gibi değerlendirilmiştir.

Tablo 13. İkinci aşamanın başlangıç mümkün çözümlerin en kalitelisi

Z.P	Derslikler															
	r36	r37	r38	r31	r27	rB	rD	rE	rF	rG	rA	rL	r50	r52	rDS1	rDS2
0		46	49				7	65	32	39	33	14		55	29	61
1			38	14	65			5	8	64		58				7
2			41	59					38	3	21		0	15	40	
3	47	6	45		8					54	35	44	64	17	37	
4	36	54	35			26			9		22		47			42
5	21	7		55		24	2		34	17	23	67	69			60
6	70	1	50	39	29	17			56	60	46		68		23	64
7	38	24	47				66	4	61		67	63				51
8	3	13	71				0		69	61	28	19	40	54	32	42
9	49	28			37	26		12		43	58	53		33	64	66
10	50				11	66		16	62	9	69		4	57	40	70
11		23	9		52	38	61	42		64	29	65	71	7		
12			22	16	43	26	12		34			66	37	70		53
13		71	60		13	35	53				51	18		20	5	
14	52	68			18	2			71	27	63	40	39	67	51	
15	23	8			46		29			57	11		62		14	3
16	28	40		66	27	14	7	45	29	23	58	18	59		48	
17	49	12		39		6		1		10		70	63	65	45	27
18	2	48		60	50			24			57	1	68	19	36	
19	44			70	40		11		38		65	20			4	16
20	10	61	69	13				38	22		6	30	55	39	34	
21	5			19	14	53		48			31		59	40	33	
22	0		56					39	41	30	21		15	69		53
23	10	25	20	15		68	60	71	52			62	41		36	
24	56					43			68		32		44	38		39

Violations of Lectures (hard) : 0
 Violations of Conflicts (hard) : 0
 Violations of Availability (hard) : 0
 Violations of RoomOccupation (hard) : 0
 Cost of RoomCapacity (soft) : 4527
 Cost of MinWorkingDays (soft) : 155
 Cost of CurriculumCompactness (soft) : 580
 Cost of RoomStability (soft) : 161

 Summary: Total Cost = 5423

Şekil 22. İkinci aşamanın ABC'sinin başlangıç çözümlerin en kalitelisinin değerlendirilmesi

Şekil 2’de görüldüğü gibi bu aday çözümde katı kısıtların hepsi sağlanmıştır. Esnek kısıtların sağlanmama durumunda alınan cezaların her birisi tek tek hesaplanıp verilmektedir. Bunlar üzerinden alınan toplam ceza 5423 tür.

Ders oturumlarının çözümünde atanan dersliğin kapasitesi dersi alan öğrencilerden az olmasından dolayı maliyet fonksiyon hesaplanmasında 4527 caza almasına neden olmuştur. Bu ceza ders oturumların atanmasında S1 esnek kısıtın sağlanmaması için alınmıştır. Örneğin 0 indeksine sahip dersi alan öğrenciler 150 kişidir. Halbuki 2’inci zaman periyodunda 80 kişi kapasitesine sahip r50 dersliğine atanmıştır ve bu da bu atanmanın bu kısıt üzerinden maliyet fonksiyonunun hesaplanmasında 70 ceza almasına neden olmuştur.

Bu aday çözümde ders oturumlarının atanmasında S2 esnek kısıtın sağlanmaması için alınan toplam ceza 161’dir. Örneğin 0 indeksine sahip dersin üç oturumu çözüm üzerinde üç farklı dersliğe atanması nedeniyle 2 ceza alınmıştır. S2 kısıtı sağlanması için oturumların hafta boyu aynı dersliğe atanması gerekir.

S3 esnek kısıtın sağlanmaması için alınan toplam ceza 155’tir. Örneğin 10 indeksine sahip dersin üç oturumu çözüm üzerinde hafta boyu iki farklı güne atanması nedeniyle 5 ceza alınmıştır. S3 kısıtı sağlanması için oturumların hafta boyu üç farklı güne atanması gerekir.

S4 esnek kısıtın sağlanmaması için alınan toplam ceza 580’dir. Örneğin q000 müfredatın 7’inci zaman periyodunda komşu periyodlarda (6, 8) aynı müfredatta olan ders oturumları bulunmamaktadır ve bu nedenle 2 ceza alınmıştır. S4 kısıtı sağlanması için her zaman periyoda olan müfredatın komşu zaman periyodlarında bu müfredatta olan derslerin en az birisinin bulunması gerekir.

İkinci aşamada uygulanan ABC algoritma adımları sonucunda 100 saniyede elde edilen en kaliteli mümkün aday çözümü Tablo 14’de verilmektedir. Bu aday çözüm [28]’de istenilen biçimde bir text dosyasında kayıt edilip [28]’de bulunan validator.exe uygulamasından yararlanarak Şekil 23’te verildiği gibi değerlendirilmiştir.

Tablo 14. İkinci aşamanın 100. saniyedeki en kaliteli çözümü

Z.P	Derslikler															
	r36	r37	r38	r31	r27	rB	rD	rE	rF	rG	rA	rL	r50	r52	rDS1	rDS2
0	5		28		4				23	7	40		39	43		38
1	6			41	29			22	52	17	40	14	39	21	60	38
2						32	10	22	1	17	40	45	66	21	61	38
3		42			58	26		15	8	64	40	35	51	70	11	49
4				46	27	71			18	64	40	35		56		
5					29		36	68	8	7		14	51			53
6					29	9	13	17	23	7	37	35	44		61	53
7	2		47		4	26	34	0		63	16	65	39	70	61	38
8		55			58	24	67	19	18	50	69	66			11	71
9			12		27	32	67	15			33	66	57			69
10		54		46		9	13	19		50	67	14	51	56	60	71
11	6			41	27		34	0	52	63	57	65		70	60	49
12			47		62	26	10	1	59	20		45	66			
13		42	28		64	32	36	68	59	3	16					53
14	5		43						18	64	37	65				69
15		54	47		29		36		23	7	33	63	66	48		69
16	2	55			58	9		68	23	50	40	19	39	48		38
17			12			24	10		1	64	57	65	44		61	38
18	6			46		71	13		8	30	37	65		56	60	53
19		42	28			31		0		3	40		39	70	11	38
20	2	54			29				59	7	33	14		48		
21	5	55					34	22	23	30	69	14		21	61	71
22			43		62	24	20	15		3			39	70	60	53
23			12	41	4		20	68	62	25	16		39			
24								68	52			45	44			49

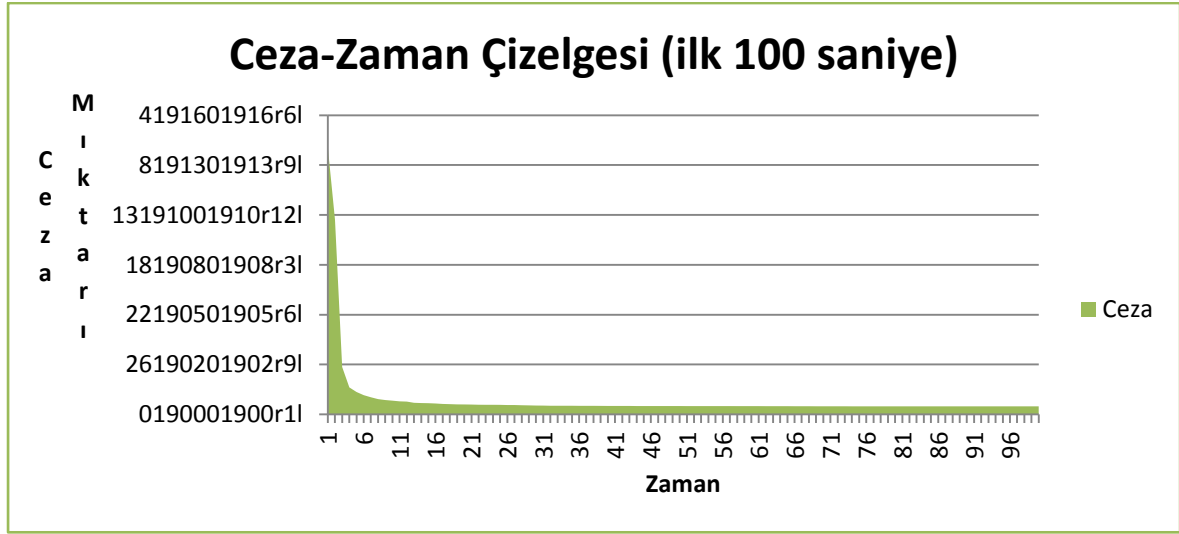
Violations of Lectures (hard) : 0
Violations of Conflicts (hard) : 0
Violations of Availability (hard) : 0
Violations of RoomOccupation (hard) : 0
Cost of RoomCapacity (soft) : 0
Cost of MinWorkingDays (soft) : 40
Cost of CurriculumCompactness (soft) : 108
Cost of RoomStability (soft) : 13

Summary: Total Cost = 161

Şekil 23. İkinci aşamanın ABC'sinin 100 saniyedeki çözümlerinin en kalitelisinin değerlendirilmesi

Şekil 23’de görüldüğü gibi bu aday çözümde katı kısıtların sağlanması bozulmadan esnek kısıtların sağlanmama durumunda alınan cezaların toplamı başlangıçta olan 5423 değerinden 161 değerine düşmüştür.

Zaman içerisinde comp15.ctt örnek problemi üzerinde önerilen yaklaşım uygulandığında elde edilen en kaliteli çözümün değeri Şekil 24’te verilen diyagramda gösterilmektedir. Bu diyagramda görüldüğü gibi ilk 6 saniye içerisinde cezaların değerinde ani düşme görülmektedir.



Şekil 24. Comp15.ctt üzerinde önerilen yaklaşımın sonuçları

İkinci aşamada uygulanan ABC algoritma adımları sonucunda belirlenen zaman içerisinde yani 468 saniyede elde edilen en kaliteli mümkün aday çözümü Tablo 15 verilmektedir. Bu aday çözümün değerlendirmesi şekil 25’de verilmektedir.

Tablo 15. İkinci aşamanın sonunda en kaliteli çözüm

Z.P	Derslikler															
	r36	r37	r38	r31	r27	rB	rD	rE	rF	rG	rA	rL	r50	r52	rDS1	rDS2
0	43				14	7	23		4	64	40	65				
1	46				14	7	23		52	64	40	65			60	38
2		45	12			26	10	1	22	17	37		66		61	
3		42				24		8	22	17	36	35	6	70	11	
4			28	44	27	32		68			33	3	39	71		38
5	41				29	7						35		48		
6	46				29	32	23	0	22	17	21	35	39			
7		42		55		30	34	8	18	64	40	65	39	70	60	38
8	51	45	47	54	69	26	67	15	56	50	40	19	66	71	60	38
9	43	58	12	44		24	63		59	16	40	3				53
10	51	58		55	14	13	67	8					66			38
11	51	57			29	15	23	67	4	50		9	66	71	60	38
12			47	44	69	30	34	1	18	64	36	65	6	70	61	49
13	5		28		27	31		68	59		37	19		48	11	53
14	2					24	63						39			53
15	2				14	7	23		56	50	21	9				
16	43	58	28	54	29	32		0	59	25	33	19	66	48	11	
17	41	57	12		69	26	10	1		64		65	6	70	61	49
18	46		47		27	15	20	68	52	62	40	3		71	60	
19	5	45				13	63			16	40					53
20	2			54	29	7		68	18		33		39			
21	5			55	14		34	68		16	21		39	71		
22		57			69		20	0	4	62			39	70		38
23	41				69	13	20		56	62	37	9			61	53
24		42					10		52		36				61	49

Violations of Lectures (hard) : 0
Violations of Conflicts (hard) : 0
Violations of Availability (hard) : 0
Violations of RoomOccupation (hard) : 0
Cost of RoomCapacity (soft) : 0
Cost of MinWorkingDays (soft) : 15
Cost of CurriculumCompactness (soft) : 84
Cost of RoomStability (soft) : 2

Summary: Total Cost = 101

Şekil 25. İkinci aşamanın ABC'sinin 468 saniyedeki çözümlerinin en kalitelisinin değerlendirilmesi

Şekil 25’de görüldüğü gibi bu aday çözümde katı kısıtların tümü sağlanmaktadır. Esnek kısıtların sağlanmama durumunda alınan cezaların toplamı 101 tir. Değerlendirmede görüldüğü gibi esnek kısıtları içerisinde sağlaması zor olan kısıt, S4 tir ve bundan dolayı en fazla ceza (84) almasına neden olmuştur. Bu değer ITC-2007 yarışmada dördüncü ve beşinci olan kişilerin comp15.ctt örnek veri üzerinde elde ettiklerinden daha iyi olduğunu göstermektedir.

3.2. ITC-2007 (CB-CTT) Örnek Problem Üzerinde Önerilen Yaklaşımın Sonuçları

CB-CTT probleminin örnek verileri üzerinde önerilen algoritmanın uygulanması her iki aşama için diğer yaklaşımlarla karşılaştırılabilen sonuçlar elde etmiştir. İlk aşama için önerilen yaklaşım önceden literatürde bulunmamaktadır. Bu aşama en hızlı şekilde tüm örnekler için mümkün çözümleri üretir. Örnekler üzerinde önerilen algoritmayla mümkün çözümler elde edildiği zaman ITC-2007’de birinci gelen kişinin yaklaşımının sonuçları [33] Tablo 16’da karşılaştırılmaktadır. Denemeler her örnek veri için 50 kere üst bölümde verilen özelliğe sahip bilgisayar üzerinde yapılmasıdır.

Önerilen yaklaşımı 21 tane örnek veri üzerinde uygulanmış ve her biri için program 10 kere 468 saniye koşturulduktan sonra elde edilen sonuçlar Tablo 17’de verilmektedir.

Bu sonuçlara göre önerilen yaklaşım comp01, comp11 örnek için birinci gelen kişinin elde ettiği değere ulaşabilmiştir. comp02, comp03, comp04, comp05, comp06, comp08, comp09, comp10, comp12, comp13, comp14, comp15, comp16, comp17, comp18, comp19, comp20 örneklerde ITC_2007 yarışmasının CB-CTT bölümünde dördüncü ve beşinci kişilerin elde ettiği değerden iyidir.

Comp07 örnek problemde sadece beşinci kişinin elde ettiği değere ulaşılmıştır. Comp21 örneği için elde edilen değer yarışmadaki kişilerin değerlerinin hepsinden iyidir.

Önerilen yaklaşım, Bolaji ve diğerleri [113] tarafından CB-CTT problemi için yararlanılan ABC algoritmasından, tüm örneklerde karşılaştığında iyi sonuçlar elde etmektedir.

CB-CTT örnek problemin çözümü hale araştırmacıların ilgisini çekmektedir ve yeni yaklaşımlar geliştirmesi için araştırmalar devam etmektedir. Önerilen en yeni yaklaşımlardan ve 21 tane örnek için en iyi sonuçlar ve güncel bilgilere bu problemin resmi sitesinden [124] ulaşılabilmektedir.

Tablo 16. İlk aşamanın sonuçları (saniye biriminden)

Örnek Veriler	Önerilen Yaklaşım	Müller'in Yaklaşımı
	Ortalama Zamanı	Ortalama Zamanı
comp01	0.22	1 veya az
comp02	4.04	3
comp03	2.73	1 veya az
comp04	1.05	1 veya az
comp05	10.50	13
comp06	2.76	1 veya az
comp07	3.25	3
comp08	1.36	1 veya az
comp09	1.56	1 veya az
comp10	2.83	1 veya az
comp11	0.10	1 veya az
comp12	2.74	1 veya az
comp13	1.16	1 veya az
comp14	1.40	1 veya az
comp15	2.89	1 veya az
comp16	2.78	1 veya az
comp17	2.19	1 veya az
comp18	0.38	1 veya az
comp19	2.87	1 veya az
comp20	2.95	1 veya az
comp21	4.34	1 veya az

Tablo 17. ITC-2007 örnek verilerin üzerinde önerilen yaklaşımların sonuçları

Yaklaşım	1	2	3	4	5	A	B	C	IABC	Önerilen Yöntem			
										468 s		600 s	
Örnekler	En İyi								Ortalama	En İyi		Ortalama	
comp01	5	5	5	5	10	5	5	13	24	6.1	5	5	6.1
comp02	51	55	50	111	111	39	75	43	299	115.5	86	75	115.6
comp03	84	71	82	128	119	76	93	76	270	116.6	101	87	107.7
comp04	37	43	35	72	72	35	45	38	166	67.3	57	45	58.5
comp05	330	309	312	410	426	315	326	314	456	405.9	377	340	391.3
comp06	48	53	69	100	130	50	62	41	255	94.3	87	87	94.8
comp07	20	28	42	57	110	12	38	19	253	75.3	61	62	67.9
comp08	41	49	40	77	83	37	50	43	173	70	60	49	63.8
comp09	109	105	110	150	139	104	119	102	271	132.4	127	119	128.1
comp10	16	21	27	71	85	10	27	14	239	64.2	51	45	68.5
comp11	0	0	0	0	3	0	0	0	220	0	0	0	0.0
comp12	333	343	351	442	408	337	358	405	751	404.8	397	363	397.8
comp13	66	73	68	622	113	61	77	68	214	97.9	90	88	97.0
comp14	59	57	59	90	84	53	59	54	221	85	77	70	80.7
comp15	84	71	82	128	119	73	87	-	238	106.7	92	92	108.1
comp16	34	39	40	81	84	32	47	-	236	99.4	83	63	78.6
comp17	83	91	102	124	152	72	86	-	280	122.1	110	111	118.2
comp18	83	69	68	116	110	77	71	-	173	121.8	97	101	116.5
comp19	62	65	75	107	111	60	74	-	276	92.6	82	78	91.7
comp20	27	47	61	88	144	22	54	-	241	89.7	77	77	90.7
comp21	103	106	123	174	169	95	117	-	364	90.5	74	77	87.2

Tablo 17’de karşılaştırılan yaklaşımlar;

1: Müler [55], **2:** Lu ve Hau [83], **3:** Atsuta ve diğerleri [56], **4:** Geiger [121] **5:** Clark ve diğerleri [122], **A:** Abdullah ve diğerleri [63], **B:** De Cesco ve diğerleri [125], **C:** Lach, G. ve Lübbecke [48], **IABC:** Bolaji ve diğerleri [113].

4. ÖNERİLER

Önerilen algoritmanın avantajlarından birisi hızlı zamanda CB-CTT probleminin örnekleri üzerinde mümkün çözümlerin üretimi ve az süre içerisinde esnek kısıtların sağlanması durumunda alınan cezaların azaltılması ve dezavantajı ise belli bir süreden sonra iyileştirmenin gelişme hızının düşmesidir. Bundan dolayı diğer tek bir çözüm tabanlı meta sezgisel yöntemlerinden (örneğin Tabu arama) melez bir şekilde kullanılırsa iyi olabilir.

Yani önerilen yaklaşımın elde ettiği en iyi çözüm belli bir süreden sonra daha fazla iyileştirme için başlangıç çözümü olarak Tabu arama mekanizmasına verilerek işleme buradan devam edilirse belki daha iyi sonuçlar elde edilebilecektir.

Bu yaklaşımının en kritik noktası çözümler üzerinde komşu aramaktır. CB-CTT problemin üzerinde iyi komşu arama mekanizmalarının geliştirilmesi sonuçların iyileşmesini sağlayacaktır.

ABC algoritmasının bu problem üzerine uyarlamak için optimizasyon parametresi periyotlar, derslik, günler ve matris haneleri ele alındığında ve denetlemeler yapıldığında en iyi sonuçlar dersliklerle elde edilmiştir. Farklı parametrelerin tanımlanması ile sonuçların iyileştirilmesi sağlanabilir.

5. KAYNAKLAR

1. Chinneck, J. W., Practical Optimization: A Gentle Introduction., Electronic document, <http://www.sce.carleton.ca/faculty/chinneck/po.html>, 10 Mayıs.2012.
2. Weise, T., Global Optimization Algorithms Theory and Application <http://www.it-weise.de/projects/book.pdf>, 10 Mayıs.2012.
3. Karaboğa, D., Yapay Zeka Optimizasyon Algoritmaları, 2.Basım, Nobel Yayın Dağıtım, Ankara, 2010.
4. Papadimitriou, C. and Steiglitz, K., Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, 1982.
5. Schrijver, A., A Course in Combinatorial Optimization, Department of Mathematics, <http://homepages.cwi.nl/~lex/files/dict.pdf>, University of Amsterdam, 10 Mayıs.2012.
6. Wren, G., Scheduling, Timetabling and Rostering A Special Relationship?, International Conference On Practice and Theory of Automated Timetabling (PATAT I), 1996, Bildiriler Kitabı, 46-75.
7. Obit, J. H., Developing Novel Meta-heuristic, Hyper-heuristic and Cooperative Search for Course Timetabling Problems, PhD thesis, University of Nottingham, School of Computer Science, 2010 November.
8. Cooper, T. and Kingston, J., The Complexity of Timetable Construction Problems, Practice and Theory of Automated Timetabling, 1996, Bildiriler Kitabı, 283-295.
9. <http://www.patat2012.com/>, 10 Mayıs.2012.
10. Qu, R., Burke, EK, McCollum, B., Merlot, L.T.G. and Lee, S.Y., A Survey of Search Methodologies and Automated System Development for Examination Timetabling, Journal of Scheduling 12,1 (2009) 55-89.
11. Çoruhlu, A., Sınav ve Personel Çizelgeleme Modeli, Yüksek Lisans Tezi, Endüstri Mühendisliği, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Ekim 2007.
12. Schaerf, A., A Survey of Automated Timetabling, Artificial Intelligence Review, 13,2 (1999) 87-127.
13. Eikelder, H. and Willemen, R., Some Complexity Aspects of Secondary School Timetabling Problems, Third international Conference on Practice and theory of Automated Timetabling III, 2000, Bildiriler Kitabı, 18-27.
14. Kingston, J.H., Resource Assignment in High School Timetabling, Annals of Operations Research, 194,1 (2012) 241-54.

15. Botsali, A.R., A Timetabling Problem: Constraint and Mathematical Approaches, Master Dissertation, The Institute of Engineering and Sciences of Bilkent University, Ankara, Şubat 2000.
16. Carter, M. ve Laporte, G., Recent Developments in Practical Examination Timetabling, Practice and Theory of Automated Timetabling (PATAT) I, 1996, Bildiriler Kitabı, 3-21.
17. Carter, M., A Survey of Practical Applications of Examination Timetabling Algorithms, Operations Research, 34 (1986) 193-202.
18. Qu, R., Burke, E. K., McCollum, B., Merlot, L. T. G. ve Lee S. Y., A Survey of Search Methodologies and Automated System Development for Examination Timetabling, Springer J Sched, 12 (2009) 55–89.
19. Bardadym, V., Computer-aided School and University Timetabling: The New Wave, Practice and Theory of Automated Timetabling, 1996, Bildiriler Kitabı, 22–45.
20. Carter, M. ve Laporte, G., Recent Developments in Practical Course Timetabling, international Conference on Practice and Theory of Automated Timetabling II, 1997, Bildiriler Kitabı, 3-19.
21. Socha, K., Knowles, J. ve Samples, M., A Max-Min Ant System for the University Course Timetabling Problem, Proceedings of the Third International Workshop on Ant Algorithms (ANTS'02), 2002, Bildiriler Kitabı, 2463, 1-13.
22. Werra, D., An Introduction to Timetabling, European Journal of Operational Research, 19 (1985) 151-162.
23. Burke, E. ve Petrovic, M., Recent Research Directions in Automated Timetabling, European Journal of Operational Research, 140 (2002) 266-280.
24. Burke, E., Elliman, D. ve Weare, R., The Automation of the Timetabling Process in Higher Education, Journal of Education Technology Systems, 23 (1995) 257-266.
25. Burke, E., Kingston, J., Jackson, K. ve Weare, R., Automated University Timetabling: The State of the Art, The Computer Journal, 40,9 (1997) 565-571.
26. Lewis, L., Metaheuristics for University Course Timetabling, PhD thesis, Napier University, 2006.
27. <http://www.idsia.ch/Files/ttcomp2002/>, Website of the International Timetabling Competition-2002, 12 Mayıs 2012.
28. <http://www.cs.qub.ac.uk/itc2007/>, Website of the International Timetabling Competition-2007, 12 Mayıs 2012.
29. Rossi-Doria, O., Sampels, M., Chiarandini, M., Knowles, J., Manfrin, M., Mastrolilli, M., Paquete, L. ve Paechter, B., A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem, Practice and Theory of Automated Timetabling IV, 2003, Bildiriler Kitabı, 329–351.

30. <http://www.metaheuristics.net/>, 12 Mayıs 2012.
31. <http://www.soc.napier.ac.uk/~benp/>, 12 Mayıs 2012.
32. McCollum, B., A Perspective on Bridging the Gap Between Theory and Practice in University Timetabling, 2007, *Bildiriler Kitabı*, 3–23.
33. McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A. J. ve et al, Setting the Research Agenda in Automated Timetabling, The second international timetabling competition, *INFORMS Journal on Computing*, 22,1 (2010) 120-30.
34. Lewis, R, Paechter, R. ve McCollum, B., Post Enrolment based Course Timetabling: A Description of the Problem Model used for Track Two of the Second International Timetabling Competition, *Cardiff Working Papers in Accounting and Finance. Prifysgol Caerdydd / Cardiff University, Wales. ISSN: 1750-6658, v 1.0 A2007-3.*
35. Di Gaspero, L., McCollum, B. ve Schaerf, A., The second International Timetabling Competition (ITC2007): Curriculum-based Course Timetabling (track 3). Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1.0/1, School of Electronics, Electrical Engineering and Computer Science, Queen’s University, Bel-fast (UK), August 2007.
36. Abdullah, S., Burke, E. K. and McCollum, B., Using a Randomised Iterative Improvement Algorithm with Composite Neighbourhood Structures for the University Course Timetabling Problem, *Metaheuristics*, 39 (2007) 153-169.
37. Gotlieb, C. C., The Construction of Class-Teacher Timetables, IFIP congress 1963, North-Holland, *Bildiriler Kitabı*, 73-77.
38. Petrovic, S. ve Burke, E. K., University Timetabling, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, CRC Press, Boca Raton, 2004.
39. Burke, E. K., Elliman, D. G. ve Weare, R. F., A University Timetabling System Based on Graph Colouring and Constraint Manipulation, *Journal of Research on Computing in Education*, 27 (1994) 1-18.
40. Burke, E. K., McCollum, B., Meisels, A., Petrovic, S. ve Qu, R., A Graph-based Hyper-heuristic for Educational Timetabling Problems, *European Journal of Operational Research*, 176 (2007) 177–192.
41. Kiaser, L. ve Yellen, J., Weighted Graphs and University Course Timetable, *Computers & Operations Research*, 19 (1992) 59-67.
42. Burke, E. K., Mareček, J., Parkes, A. J. and Rudová, H., On a Clique-based Integer Programming Formulation of Vertex Colouring with Applications in Course Timetabling, <http://arxiv.org/abs/0710.3603>, 10 Mayıs.2012.
43. Burke, E., MacCarthy, B., Petrovic, S. ve Qu, R., Case-Based Reasoning in Course Timetabling: An Attribute Graph Approach, Case-Based Reasoning Research and Development, 4th International Conference on Case-Based Reasoning, 2001, Vancouver Canada, *Bildiriler Kitabı*, 90-104.

44. Lawrie, N. L., An Integer Linear Programming Model of a School Timetabling Problem, Comput J, 12 (1969) 307–16.
45. Burke, E. K., Mareček, J., Parkes, A. J. ve Rudová, H., Penalising Patterns in Timetables: Strengthened Integer Programming Formulations, Operations research proceedings, (2007) 409–414.
46. Daskalaki, S. ve Birbas, T., Efficient Solutions for a University Timetabling Problem Through Integer Programming. European Journal of Operational Research, 127,1 (2005) 106–120.
47. Daskalaki, S., Birbas, T. ve Housos, E., An Integer Programming Formulation for a Case Study in University Timetabling. European Journal of Operational Research, 153 (2004) 117–135.
48. Lach, G. ve Lübbecke, M. E., Curriculum Based Course Timetabling: New Solutions to Udine Benchmark Instances, Annals of Operations Research, (2010) 1-18.
49. Hao, J. K. ve Benlic, U., Lower Bounds for the ITC-2007 Curriculum-Based Course Timetabling Problem, European Journal of Operational Research, 212 (2011) 464–472.
50. Burke, E. K., Mareček, J., Parkes, A. J. ve Rudová, H., Uses and Abuses of MIP in Course Timetabling, Poster at the workshop on mixed integer programming MIP2007, Montréal, Available online at <http://cs.nott.ac.uk/jxm/timetabling/mip2007-poster.pdf>, 10 Mayıs.2012.
51. Burke, E. K., Marecek, J., Parkes, A.J. ve Rudova, H., Decomposition, Reformulation, and Diving in University Course Timetabling, Computers & Operations Research, 37,3 (2010) 582–597.
52. White, G. M. ve Chan, P. W., Towards the Construction of Optimal Examination Timetables, INFOR, 17 (1979) 219-229.
53. Ferland, J. A. ve Roy, S., Timetabling problem for university as assignment of activity to resources. Computer and Operational Research, 12,2 (1985) 207-218.
54. Zervoudakis, K. ve Stamatopolous, P., A Generic Object-Oriented Constraint-Based Model for University Course Timetabling, The Third International Conference Practice and Theory of Auto-mated Timetabling, 2001, , Bildiriler Kitabı, 28-47.
55. Müller, T., ITC2007 solver description: a hybrid approach. Ann Oper Res, 172 (2009) 429–446.
56. Atsuta, M., Nonobe, K.ve Ibaraki, T., ITC-2007 Track2: An Approach using General CSP Solver.(2007).http://www.cs.qub.ac.uk/itc2007/winner/bestcoursesolutions/Atsuta_et_al.pdf, 10 Mayıs.2012.

57. Burke, EK., MacCarthy, BL., Petrovic, S. ve Qu, R., Knowledge Discovery in a Hyper-Heuristic for Course Timetabling Using Case-Based Reasoning, Practice and Theory of Automated Timetabling IV, 2003, Bildiriler Kitabı, 276-287.
58. Lewis, R., A Survey of Metaheuristic-based Techniques for University Timetabling Problems, Operation Research Journal, 30 (2008) 167-190.
59. Colomi, A., Dorigo, M. ve Maniezzo, V., Metaheuristics for High-School Timetabling, Comput Optim App, 19,3 (1997) 277-298.
60. Di Gaspero, L. ve Schaerf, A., Multi-Neighbourhood Local Search with Application to Course Timetabling, Practice and theory of automated timetabling (PATAT) IV, 2002, Bildiriler Kitabı, 263-287.
61. Arntzen, H. ve Løkketangen, A., A Tabu Search Heuristic for a University Timetabling Problem, Metaheuristics: progress as real problem solvers, 32 (2005) 65-86.
62. Lewis, R. ve Paechter, B., New Crossover Operators for Timetabling with Evolutionary Algorithms, The fifth international conference on recent advances in soft computing RASC2004. Nottingham, 2004, England, Bildiriler Kitabı, 189-19,
63. Abdullah, S., Turabieh, T., McCollum, B. ve McMullan, P., A Hybrid Metaheuristic Approach to the University Course Timetabling Problem, J Heuristics (2010) 1-23.
64. Kostuch, P., The University Course Timetabling Problem with a 3-Phase Approach, Practice and theory of automated timetabling (PATAT), 2005, Bildiriler Kitabı, 109-125.
65. Nebiyev, V. V., Yapay Zeka, 3 baskı, Seçkin yayıncılık, Ankara, 2010.
66. Holland, J. H., Adaptation in Natural and Artificial Systems, University of Michigan Press, Anna Arbor, 1975.
67. Erben, W. ve Keppler, J., A Genetic Algorithm Solving a Weekly Course-timetabling Problem, 1st International Conference on the Practice and Theory of Automated Timetabling (PATAT I), 1996, Bildiriler Kitabı, 198-211.
68. Ueda, H., Ouchi, D., Takahashi, K. ve Miyahara, T., A Coevolving Timeslot/Room Assignment Genetic Algorithm Technique for University Timetabling, 3rd International Conference on the Practice and Theory of Automated Timetabling (PATAT III), 2001, Bildiriler Kitabı, 48-63.
69. G. Konstantinow, G. ve Coakley, C., Use of Genetic Algorithms in Reactive Scheduling for Course Timetable, In proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling, 2004, Bildiriler Kitabı, 3-24.
70. Abdullah, S. ve Turabieh, H., Generating University Course Timetable Using Genetic Algorithm and Local Search, in Proc. 3rd Int. Conf. Hybrid Inform. Tech., 2008, Bildiriler Kitabı, 254-260.

71. Yang, S. ve S. N. Jat., Genetic Algorithms with Guided and Local Search Strategies for University Course Timetabling, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, 41 (2011) 93-106.
72. Abdullah, S., et al., An Investigation of a Genetic Algorithm and Sequential Local Search Approach for Curriculum-based Course Timetabling Problems, in In Proceedings of the 4th Multidisciplinary International Scheduling Conference: Theory and Applications (MISTA 2009) 2010, Bildiriler Kitabı, 727-731.
73. Dorigo, M., Maniezzo, V. and Colorni, A., The Ant System: An Autocatalytic Optimizing Process, Technical report, Technical Report TR91-016, Politecnico di Milano, 1991.
74. Socha, K., Sampels, M. ve Manfrin, M., Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art, In Applications of Evolutionary Computing: Proceedings of the Evo Workshops, 2003, Bildiriler Kitabı, 334-345.
75. Moscato, P., On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Report 826, Caltech Concurrent Computation Program, California Institute of Technology, Pasadena CA, USA, 1989.
76. Burke, E.ve Landa-Silva, J., The Design of Memetic Algorithms for Scheduling and Timetabling Problems, Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing, 2004, Bildiriler Kitabı, 289-312.
77. Alkan, A. ve Ozcan, E., Memetic Algorithms for Timetabling, In proceeding IEEE Congress on Evolutionary Computation, December 2003, Bildiriler Kitabı, 1796-1802
78. Glover, F., Tabu Search: A Tutorial, Interfaces, 20 (1990) 74-94.
79. Elloumi, K. H., Abdelkarim and Ferland, J., A Tabu Search Procedure for Course Timetabling Problem at a Tunisian. In proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling, 2008.
80. Costa D., A Tabu Search Algorithm for Computing an Operational Timetable, European Journal of Operational Research, 76 (1994) 9-110.
81. Alvarez-Valdes, R., Crespo, E. ve Tamarit, J., Assigning Students Sections Using Tabu Search. Annals of Operations Research, 96 (2000) 1-16.
82. Arntzen, H. and Lokketangen, A., A Tabu Search Heuristic for a University Timetabling Problem. In Metaheuristics: Progress as Real Problem Solvers, Computer Science Interfaces Series, 32 (2005) 65-86.
83. Lü, Z ve Hao, J.K., Adaptive Tabu Search for Course Timetabling, European Journal of Operational Research, 200 (2010) 235-244.
84. Mladenovic, N. ve Hansen, P., Variable Neighbourhood Search. Computers and perations Research, 24,11 (1997) 1097-1100.

85. Abdullah, S., E. K. Burke, and B. McCollum. An Investigation of Variable Neighbourhood Search for University Course Timetabling. In *The 2nd Multidisciplinary Conference on Scheduling: Theory and Applications*, 2005, *Bildiriler Kitabı*, 413-427.
86. Kirkpatrick, S., Gelatt, C.D. ve Vecchi, M.P., *Optimization by Simulated Annealing*, Science, 220,4598 (1983) 671–680.
87. Davis, L. and Ritter, L., *Schedule Optimization with Probabilistic Search*. In *proceedings of the 3rd IEEE Conference on Artificial Intelligence Applications*, 1987, Orlando, Florida, USA, *Bildiriler Kitabı*, 231-236,.
88. Elmohamed, M., Coddington, P. and Fox, G., *A Comparison of Annealing Techniques for Academic Course Scheduling*, *The Practice and Theory of Automated Timetabling II: Selected Papers from 2nd International Conference on the Practice and Theory of Automated Timetabling (PATAT II)*, Toronto Canada, 1998, *Bildiriler Kitabı*, 92-112.
89. Tuga, M., Berretta, R. and A. Mendes, *A Hybrid Simulated Annealing with Kempe Chain Neighborhood for the University Timetabling Problem*, in *Proc. 6th IEEE/ACIS Int. Conf. Comput*, 2007, *Bildiriler Kitabı*, 400–405.
90. Dueck, G., *New Optimization Heuristic: The Great Deluge Algorithm and the Record-to-Record Travel*, Journal of Computational Physics, 104 (1993) 86-92.
91. Petrovic, S. and Burke, E. K., *University Timetabling*, Chapman & Hall/CRC, 2004.
92. Obit, J. H. ve Landa-Silva, D., *Computational Study of Non-Linear Great Deluge for University Course Timetabling*. *Intelligent Systems: From Theory to Practice*, Studies in Computational Intelligence, 299 (2010) 309-328.
93. Landa-Silva, D. and Obit J. H, *Evolutionary Non-Linear Great Deluge for University Course Timetabling*, *Proceedings of the 2009 International Conference on Hybrid Artificial Intelligence Systems (HAIS 2009)*, *Hybrid Artificial Intelligent Systems*, 2009, *Bildiriler Kitabı*, 269-276.
94. Chiarandini, M., Birattari, Socha, K. ve Rossi-Doria, O, *An Effective Hybrid Algorithm for University Course Timetabling*, Journal of Scheduling, 9 (2006) 403-432
95. Burke, E., Bykov, Y., Newall J.P. ve Petrovic, S., *A Time-Defined Approach to Course Timetabling*, Yugoslav Journal of Operations Research (YUJOR), 13 (2003) 139-151.
96. Bonabeau, E., Dorigo, M. and Theraulaz, G., *Swarm Intelligence: From Natural to Artificial Systems*, , NY: Oxford University Press, New York, 1999.
97. Karaboga D.,*An Idea Based on Honey Bee Swarm for Numerical Optimization*, *Technical Report TR06*, Computer Engineering Department, Erciyes University, Turkey, 2005.

98. Karaboga, D. and Akay, B., A Comparative Study of Artificial Bee Colony Algorithm, Applied Mathematics and Computation, 214 (2009) 108-132.
99. Aydın, D., Sürü Zekası Yaklaşımlarının Görüntü Kesimlemeye Uyarlanması ve Tanıma Sistemleri Üzerinde Gerçekleştirimi, Doktora Tezi, EGE Üniversitesi Fen Bilimleri Enstitüsü, İzmir Bornova, 2011.
100. Akay, B., Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi, Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri, 2009.
101. Tereshko, V. and Loengarov, A., Collective Decision-Making in Honey Bee Foraging Dynamics, Computing and Information Systems, 2005, University of the West of Scotland UK, Bildiriler Kitabı, 1-7.
102. http://www.scholarpedia.org/article/Artificial_bee_colony_algorithm, 10 Mayıs 2012.
103. Kennedy, J. and Eberhart, R. C., Particle Swarm Optimization, IEEE International Conference on Neural Networks, 1995, Bildiriler Kitabı, 1942–1948.
104. Singh, A. and Sundar, S., An Artificial Bee Colony Algorithm for the Minimum Routing Cost Spanning Tree Problem, Soft Computing-A Fusion of Foundations, Methodologies and Applications, 15 (2011) 1-11.
105. Sonmez, M., Discrete Optimum Design of Truss Structures Using Artificial Bee Colony Algorithm, Structural and Multidisciplinary Optimization, 43,1 (2011) 85-97.
106. Sonmez, M., Artificial Bee Colony Algorithm for Optimization of Truss Structures, Applied Soft Computing, 11,2 (2011) 2406-18.
107. Karaboga, D. and Gorkemli, B. A., Combinatorial Artificial Bee Colony Algorithm for Traveling Salesman Problem, In: 2011 international symposium on innovations in intelligent systems and applications (INISTA), 2011, Bildiriler Kitabı, 50–53.
108. Pan, Q.K., Tasgetiren, M.F., Suganthan, P.N. ve Chua, T.J., A Discrete Artificial Bee Colony Algorithm for the Lot-Streaming Flow Shop Scheduling Problem, Information Sciences, 181 (2011) 2455-2468.
109. Alzaqebah, M. and Abdullah, S., Artificial Bee Colony Search Algorithm for Examination Timetabling Problems, Int J Phys Sci, 6,17 (2011) 4264–4272.
110. Alzaqebah, M. and Abdullah, S., Comparison on the Selection Strategies in the Artificial Bee Colony Algorithm for Examination Timetabling Problems, Int J Soft Comput Eng, 1,5 (2011) 158–163.
111. Oner, A., Ozcan, S. and Dengi, D., Optimization of University Course Scheduling Problem with a Hybrid Artificial Bee Colony Algorithm, In: 2011 IEEE congress on evolutionary computation (CEC) 2011, Bildiriler Kitabı, 339–346.

112. Tokmak, M., Yapay Arı Kolonisi Algoritması İle Ders Çizelgeleme Probleminin Çözümü, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi Fen Bilimler Enstitüsü, Elektronik Bilgisayar Eğitim Anabilim, Sparta, 2011.
113. Bolaji, A. L., Khader, A.T., Al-betar, M. A. and Awadallah, M., An Improved Artificial Bee Colony for Course Timetabling, in Sixth International Conference on Bio-Inspired Computing, Theories and Applications, (2011) 9-14.
114. Karaboga, D., Gorkemli, B., Ozturk, C. and Karaboga, N., A Comprehensive Survey: Artificial Bee Colony (ABC) Algorithm and Applications, Artificial Intelligence Review, (2012) 1-37.
115. Lu, Z., Hao, J. and Glover, F., Neighborhood Analysis: A Case Study on Curriculum-Based Course Timetabling, Journal of Heuristics,17,2 (2011) 97-118.
116. Frausto-Sols, J., Alonso-Pecina, F. ve Mora-Vargas, J., An Efficient Simulated Annealing Algorithm for Feasible Solutions of Course Timetabling, Advances in Artificial Intelligence, 5317 (2008) 675-685.
117. A. Mayer, C. Nothegger, A. Chwatal, and G. Raidl. Solving the Post Enrolment Course Timetabling Problem by Ant Colony Optimization, In proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling, 2008.
118. Ejaz, N. and. Younus, J. M. A Hybrid Approach for Course Scheduling Inspired by Die-hard Co-operative Ant Behavior. In proceedings of the IEEE International Conference on Automation and Logistics. Jinan, China, 2007.
119. Lewis, R. ve Paechter, B, Finding Feasible Timetable Using Group-Based Operators. IEEE Transactions on Evolutionary Computation, 11,3 (2007) 397-413.
120. Landa-Silva, D. ve Obit, J.H. Great Deluge with non-Linear Decay Rate for Solving Course Timetabling Problem, The Fourth International IEEE Conference on Intelligent Systems, 2008, Bildiriler Kitabı, 11–18.
121. Geiger M., Applying the Threshold Accepting Meta-Heuristic to Curriculum Based Course Timetabling, Annals of Operations Research, (2010) 1-14.
122. Clark, M., Henz, M. and Love, B., QuikFix A Repair-based Timetable Solver., <http://www.comp.nus.edu.sg/~henz/publications/ps/PATAT2008.pdf>, 10 Mayıs.2012.
123. Birbil, S. I. and Fang, S. C., An Electromagnetism-like Mechanism for Global Optimization, J. Glob. Optim, 25 (2003) 263–282.
124. <http://tabu.diegm.uniud.it/ctt/index.php>, 10 Mayıs.2012.
125. De Cesco, F., Di Gaspero, L. and Schaerf, A., Benchmarking Curriculum-based Course Timetabling: Formulations,Data Formats, Instances, Validation, and Results. Proceedings of the Seventh PATAT Conference 2008, <http://tabu.diegm.uniud.it/ctt/DDS2008.pdf>, 10 Mayıs.2012.

ÖZGEÇMİŞ

Saeid AGAHIAN 1983 yılında İran'nin Bonab (Binab, Binev Azericede) Şehrinde doğdu. İlköğrenimini 1990 yılında Bonab'da tamamladıktan sonra 2002 yılında Bonab'ın Şeykh behayi lisesinden mezun oldu. 2002 yılında Bonab Peyam Noor Üniversitesi Bilgisayar Mühendisliği Bölümü'ne girmeye hak kazanmış ve öğrenimini burada 2006 yılında tamamlanmış ve 2010 yılında Karadeniz Teknik Üniversitesi Fen Bilimleri Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans öğrenimine başlamıştır. Bekar olan Saeid AGAHIAN iyi derecede Azerice, Türkçe, Farsça ve orta derecede İngilizce bilmektedir.