

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**CIRCIR BÖCEĞİ ALGORİTMASI: YENİ BİR META-SEZGİSEL
YAKLAŞIM VE UYGULAMALARI**

MURAT CANAYAZ

**DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

AĞUSTOS 2015

Tezin Başlığı: Cırcır Böceği Algoritması: Yeni bir Meta-sezgisel Yaklaşım ve Uygulamaları

Tezi Hazırlayan: Murat CANAYAZ

Sınav Tarihi: 14.08.2015

Yukarıda adı geçen tez jürimizce değerlendirilerek Bilgisayar Mühendisliği Ana Bilim Dalında Doktora Tezi olarak kabul edilmiştir.

Sınav Jüri Üyeleri

Tez Danışmanı: **Prof. Dr. Ali KARCI**
İnönü Üniversitesi

Doç.Dr. Resul DAŞ
Fırat Üniversitesi

Doç. Dr. Bilal ALATAŞ
Fırat Üniversitesi

Yard. Doç. Dr. N. Murat YAĞMURLU
İnönü Üniversitesi

Yard. Doç. Dr. A. Fatih KOCAMAZ
İnönü Üniversitesi

Prof. Dr. Alaattin ESEN
Enstitü Müdürü

ONUR SÖZÜ

“Doktora Tezi olarak sunduđum “Cırcır Böceđi Algoritması: Yeni bir Meta-sezgisel Yaklaşım ve Uygulamaları” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuđunu belirtir, bunu onurumla doğrularım.”

Murat CANAYAZ

ÖZET

Doktora Tezi

CIRCIR BÖCEĞİ ALGORİTMASI: YENİ BİR META-SEZGİSEL YAKLAŞIM VE UYGULAMALARI

Murat CANAYAZ

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

77+viii sayfa

2015

Danışman: Prof. Dr. Ali KARCI

Problemlerin çözümünde en iyi metodun aranması işlemi geçmişten günümüze bilim insanlarının ortak amacı olmuştur. Bu amaç doğrultusunda gerek klasik matematiksel yöntemler gerekse de geliştirilen algoritmalar problemlerin çözümü için vazgeçilmez faktörlerdendir. Meta-sezgisel algoritmalar en iyinin bulunmasında son yıllarda popüler hale gelmiş, halen bir çok optimizasyon problemlerinde kullanılmaktadır. Bu doktora tez çalışmasında genel olarak sosyal tabanlı, fizik tabanlı, biyolojik tabanlı, kimya tabanlı, müzik tabanlı, sürü tabanlı, spor tabanlı, matematik tabanlı ve melez olmak üzere 9 farklı kategoride değerlendirdiğimiz meta-sezgisel algoritmalar incelenmiştir. Bu meta-sezgisel algoritmalar arasına girebilecek tabiatta bulunan cırcır böceği davranışlarından esinlenilerek geliştirdiğimiz yeni bir meta-sezgisel algoritma yaklaşımı önerilmiş ve tanıtılmaya çalışılmıştır.

Cırcır böcekleri doğada ses ile iletişime geçen ve kanat çırpışları ile o andaki havanın sıcaklığını tahmin etme özelliğine sahip mucizevi canlılardır. Bu tez çalışması kapsamında cırcır böceğinin davranışları incelenmiş, bu inceleme sırasında Parçacık sürü optimizasyonu, Ateş böceği ve Yarasa algoritmalarında yer alan canlı türleri ile ortak bazı özelliklere sahip oldukları görülmüştür. Bu aşamadan sonra cırcır böceği davranışlarının matematiksel olarak modellenmesi yapılarak yeni bir meta-sezgisel algoritma yaklaşımı önerilmiştir. Modelleme yapılırken algoritmanın doğaya yakınlığını sağlamak amacıyla sesin doğadaki yayılımı ile alakalı fiziki kanunlar göz önünde bulundurulmuştur. Önerdiğimiz algoritmanın performansını değerlendirmek amacıyla öncelikle test fonksiyonları üzerinde çalıştırılmıştır. Daha sonra sırasıyla bilinen mühendislik optimizasyon problemleri, imge işleme uygulamalarında algoritma çalıştırılarak performans değerlendirilmesi yapılmış ve sonuçlar karşılaştırmalı olarak gösterilmiştir.

ANAHTAR KELİMELER: Optimizasyon, meta-sezgisel algoritma, cırcır böceği algoritması, evrimsel hesaplama

ABSTRACT

Ph.D. Thesis

CRICKET ALGORITHM: A NEW META-HEURISTIC APPROACH AND APPLICATIONS

Murat CANAYAZ

Inönü University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

77+viii pages

2015

Supervisor: Prof. Dr. Ali KARCI

The process of seeking the best for answers of problems has been a common purpose for the scientists from the past to the present. Both classic mathematical methods and algorithms developed in the process of quest of answers of problems are indispensable factors. Meta-heuristic algorithms have become very popular and been still used for optimization problems. In this dissertation, we studied meta-heuristic algorithms in nine different categories – social based, physics based, biological based, chemistry based, music based, swarm based, sport based, mathematics based and finally hybrid based – have been researched. The new approach of meta-heuristic algorithms which is likely to be etched among these meta-heuristic algorithms we developed with inspiration of the attitudes of a cricket in nature has been proposed and endeavored to be introduced.

Crickets getting contact with voice and having the ability to predict the instant air temperature with chirping are miraculous living creatures. The attitudes of crickets have been examined within the scope of this dissertation and have been indicated to possess some common characteristics with some species taking part in the Particle Swarm Optimization, Firefly and Bat algorithms. A new approach of meta-heuristic algorithms has been recommended by mathematically modeling of attitudes of cricket after this phase. The physics laws related to sound propagation in nature on the purpose of verification of closeness to nature have been taken into consideration while modeling. It has been run on test functions primarily with the aim of evaluating the performance of the algorithm we have proposed. Afterwards, common engineering optimization problems and the performance evaluation by actuating the algorithm in the image processing applications have been proceeded respectively and the comparative results have been shown.

KEYWORDS: Optimization, meta-heuristic algorithm, cricket algorithm, evolutionary computation

TEŐEKKÖR

Bu tez alıőmasının baőlangıcından sonuna kadar, bilgisi, uzmanlıđı, sabır ve anlayıőıyla bana canı gönülden rehberlik eden ve sürekli destek veren danıőmanım Sayın Prof. Dr. Ali KARCI hocama en derin teőekkürlerimi ve őükranlarımı sunarım. Ayrıca benim için çok deđerli olan aileme ve dostlarıma doktora alıőmalarım sırasında verdikleri desteklerinden dolayı teőekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ŞEKİLLER LİSTESİ	vi
ÇİZELGELER LİSTESİ	vii
SİMGELER VE KISALTMALAR	viii
1. GİRİŞ	1
1.1. Tez Çalışmasının Amacı ve Kapsamı:	3
1.2. Literatür Taraması ve Değerlendirilmesi:	3
1.3. Tez Çalışmasının Organizasyonu:	4
2. META-SEZGİSEL ALGORİTMALAR	5
2.1. Sosyal Tabanlı Algoritmalar	6
2.2. Fizik Tabanlı Algoritmalar	7
2.3. Biyoloji Tabanlı Algoritmalar	9
2.4. Kimya Tabanlı Algoritmalar	11
2.5. Müzik Tabanlı Algoritmalar	12
2.6. Spor Tabanlı Algoritmalar	12
2.7. Matematik Tabanlı Algoritmalar	12
2.8. Sürü Tabanlı Algoritmalar	13
2.8.1. Parçacık sürü optimizasyonu	15
2.8.2. Yarasa algoritması	18
2.8.3. Ateş böceği algoritması	20
3. ÖNERİLEN YÖNTEM: CIRCIR BÖCEĞİ ALGORİTMASI	22
3.1. Cırcır Böceği Davranışları	22
3.2. Doğadaki Sesin Yayılımı ile Alakalı Temel Bilgiler	23
3.2.1. Sesin hızı	23
3.2.2. Sesin frekansı ve dalga boyu	23
3.2.3. Havanın sesi emme katsayısı	24
3.2.4. Sesin şiddeti	24
3.2.5. Ses gücü seviyesi	25
3.2.6. Ses basıncı seviyesi	25
3.2.7. Atmosferin sesi yutusu	25

3.2.8. Serbest alanda gerçek ses basınç düzeyi	25
3.3. Cırcır Böceği Algoritması	26
4. CBA’NIN TEST FONKSİYONLARINA UYGULANMASI VE BAŞARI ANALİZİ	32
5. CBA’NIN MÜHENDİSLİK PROBLEMLERİNE UYGULANMASI VE BAŞARI ANALİZİ	36
5.1. Beş basamaklı giriş optimizasyon problemi:	36
5.2. Üç çubuk makas optimizasyon problemi:	38
5.3. Himmelblau optimizasyon problemi:.....	40
5.4. Yay dizayn optimizasyon problemi:	43
5.5. Hız düşürücü optimizasyon problemi:	46
6. CBA’NIN İMGE İŞLEME UYGULAMALARINDAKİ BAŞARI ANALİZİ	50
6.1. Entropi temelli imge eşikleme	50
6.2. Entropi temelli imge sıkıştırma	54
6.2.1. Ayrık dalgacık dönüşümü	54
6.2.2. Shannon ve Tsallis entropi metotları.....	56
6.2.3. İmge sıkıştırma yöntemi.....	58
7. TARTIŞMA VE SONUÇ.....	63
8. KAYNAKLAR	65
EKLER.....	72
ÖZGEÇMİŞ.....	76

ŞEKİLLER LİSTESİ

Şekil 1.1. Sezgisel yöntemler	2
Şekil 3.1. CBA akış şeması.....	30
Şekil 5.1. Beş basamaklı giriş	36
Şekil 5.2. Beş basamaklı giriş yakınsama grafiği	38
Şekil 5.3. Üç çubuk makas	39
Şekil 5.4. Üç çubuk makas yakınsama grafiği	40
Şekil 5.5. Himmelblau fonksiyonu birinci sürüm yakınsama grafiği	42
Şekil 5.6. Himmelblau fonksiyonu birinci sürüm yakınsama grafiği	43
Şekil 5.7. Yay dizayn	44
Şekil 5.8. Yay dizayn problemi yakınsama grafiği	45
Şekil 5.9. Hız düşürücü	46
Şekil 5.10. Hız düşürücü problemi yakınsama grafiği.....	48
Şekil 6.1. Test imgeleri ve histogram değerleri	52
Şekil 6.2. Eşiklenmiş test imgeleri	53
Şekil 6.3. Alt bantlar temsili gösterimi	55
Şekil 6.4. Üç seviyeli ayrık dalgacık dönüşümü uygulanmış imge	56
Şekil 6.5. İmge 1 sıkıştırma oranları	60
Şekil 6.6. İmge 2 sıkıştırma oranları	60
Şekil 6.7. İmge 3 sıkıştırma oranları	61
Şekil 6.8. İmge 4 sıkıştırma oranları	61

ÇİZELGELER LİSTESİ

Çizelge 3.1. CBA sözde kodu.....	31
Çizelge 4.1. Test fonksiyonları	32
Çizelge 4.2. Fonksiyon global minimum değerleri.....	34
Çizelge 4.3. CBA'dan elde edilen fonksiyon sonuçları	35
Çizelge 5.1. Beş basamaklı giriş optimizasyon problemi sonuçları	38
Çizelge 5.2. Üç çubuk makas optimizasyon problem sonuçları	40
Çizelge 5.3. Üç çubuk makas problemi için en iyi çözümler.....	40
Çizelge 5.4. Himmelblau fonksiyonu ilk versiyonu için elde edilen değerler	41
Çizelge 5.5. Himmelblau fonksiyonu ilk versiyonu için en iyi çözümler.....	42
Çizelge 5.6. Himmelblau fonksiyonu ikinci versiyonu için elde edilen değerler	42
Çizelge 5.7. Himmelblau fonksiyonu ikinci versiyonu için en iyi çözümler.....	43
Çizelge 5.8. Yay dizayn problemi için CBA'dan elde edilen değerler.....	44
Çizelge 5.9. Yay dizayn problemi için en iyi sonuçlar	45
Çizelge 5.10. Hız düşürücü optimizasyon problem sonuçları	47
Çizelge 5.11. Hız düşürücü optimizasyon problem için en iyi çözümler	48
Çizelge 5.12. Problemlerden elde edilen toplu sonuçlar.....	49
Çizelge 6.1. Entropi temelli eşikleme için CBA'dan elde edilen değerler	51
Çizelge 6.2. Entropi temelli sıkıştırma için CBA'dan elde edilen değerler.....	62

SİMGELER VE KISALTMALAR

β	Rastgele dağılımlı katsayı
λ_{\min}	Minimum dalga boyu
λ_{\max}	Maksimum dalga boyu
ϵ	Rastgele sayı
γ	Çekicilik katsayısı, havanın sesi absorbe etme katsayısı
α	Rastlantısal parametre
ρ	Ortamin yoğunluğu
Q	Yönelme katsayısı
\emptyset	Havanın bağıl nem oranı
ψ	Dönüşüm fonksiyonu
ϕ	Entropik indeks
w	Atalet sabiti
ABA	Ateş Böceği Algoritması
BT	Benzetilmiş Tavlama
CBA	Cırcır Böceği Algoritması
GA	Genetik Algoritmalar
PSO	Parçacık Sürü Optimizasyonu
YA	Yarasa Algoritması

1. GİRİŞ

Optimizasyon, yapılacak olan işlerde belirli şartları gözeterek en iyinin aranması işlemleri olarak bilinmektedir. Diğer bir ifadeyle optimizasyon, hedeflenen amaca ulaştıracak paydaşların aranması olarak da tanımlanabilir. Günlük hayatın her alanında karşılaşılabileceğimiz bir kavram olduğundan dolayı optimizasyonun ne amaçladığını iyi bilmemiz gerekir. Örneğin bir işletmede kârın maksimum düzeyde olması, giderlerin minimuma indirilmesi veya çalışan bir sistemin optimum kararlılıkta çalışması herkes tarafından istenen bir durumdur (Canayaz ve Karcı, 2015a). Dahası üretim aşamasındaki ürünlerde de belirli kuralların olması ve bu kuralları sağlayacak optimum durumların sağlanması gerekmektedir. Optimizasyon burada işin içine dahil olarak belirlenen şartlara göre bu işlemlerin gerçekleşmesini amaçlar. Matematiksel olarak optimizasyon, verilen bir fonksiyonu belirli kısıtlara uygun olarak minimum veya maksimum yapan değerlerin bulunmasını sağlayan işlemler bütünüdür. Basitçe ifade etmek gerekirse optimizasyon alternatif çözümleri bulunan bir problem için en iyi olanı seçme işlemi olarak da tanımlanabilir (Alba, 2005).

Genel bir optimizasyon probleminin matematiksel tanımı:

- Optimize edilecek fonksiyon $f(x)$ olarak kabul edilirse,
- Kısıtlayıcı fonksiyonlar $f(x) = 0$ ve $g(x) \leq 0$
- Tasarım değişkenleri $x_l \leq x \leq x_u$

Optimize edilecek fonksiyon hedef fonksiyon veya uygunluk fonksiyonu olarak tanımlanmaktadır. Bu hedef fonksiyondaki değişkenlere tasarım değişkenleri, fonksiyonun sağlanmasında uyulması gereken kısıtlayıcıları sağlayan fonksiyonlara da kısıtlayıcı fonksiyon adı verilir. Burada amaç, verilen probleme göre hedef fonksiyonun minimum veya maksimum yapılmasıdır (Rao, 1996).

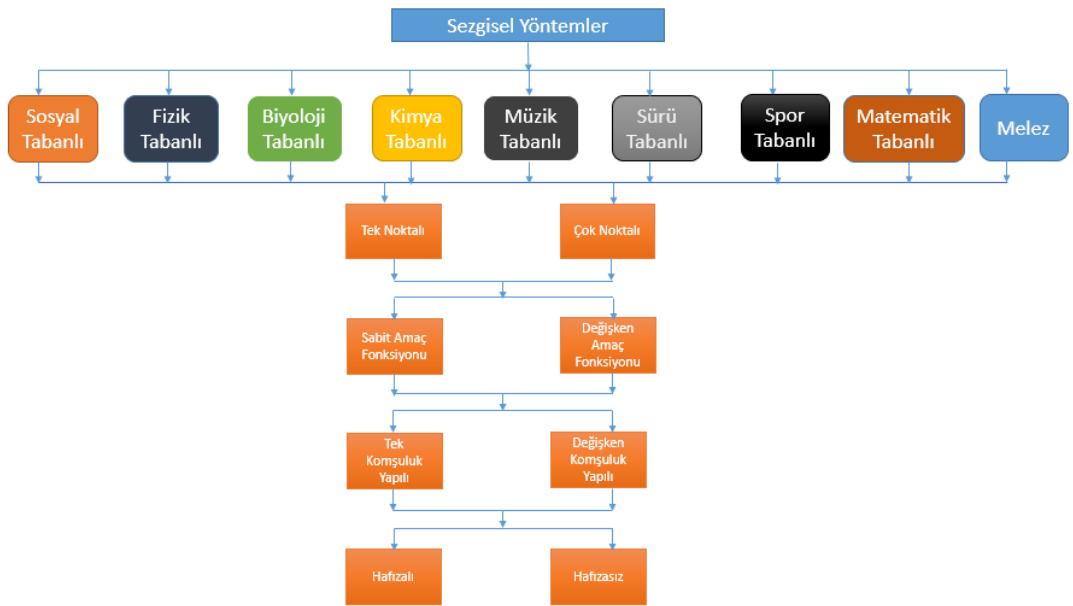
Optimizasyonu, modelleme ve çözümlenme olarak iki alana ayırmak mümkündür (Türkay, 2015). Modelleme belirli durumlar gözetilerek verilen işin veya belirli bir durumun matematiksel olarak modellenmesi işlemidir. Çözümlenme ise modeli sağlayan en iyi çözümün elde edilmesi için yapılacak olan işlemlerin veya kullanılacak elemanların seçilmesi gibi işleri yapma temeline dayanır. Bu çözümlenme işlemlerinde kullanılan birçok algoritma bulunmaktadır. Bu algoritmalar genel olarak

iki grupta incelenmektedir. Bunlar belirleyici (deterministik) ve rastgele (stokastik) algoritmalarıdır.

Deterministik algoritmalar rastgele değer üretilmesinin gerekmediği, parametrelerin sabit değerler aldığı matematiksel fonksiyonların çözümünde kullanılan algoritmalarıdır (Yang, 2013).

Stokastik algoritmalarda ise tanımlanan parametreler rastgele değerler alır ve bu algoritmalar doğrusal olmayan problemlerin çözümünde kullanılır. Bu algoritmalar bazen sezgisel bazen de meta-sezgisel olarak adlandırılır. Bu algoritmalar son yıllarda optimize edilecek modelin çözümü için uygun değerlerin bulunmasında önemli bir yere sahip olmaya başlamıştır.

Sezgisel yöntemler en iyi sonucu bulacaklarını konusunda garanti vermeseler bile en kısa zamanda makul bir sonuç bulacaklarını garanti ederler. Bu yöntemler genel olarak Şekil 1.1'deki gibi sınıflandırılabilir (Akyol ve Alataş, 2012).



Şekil 1.1. Sezgisel yöntemler

Meta-sezgisel algoritmalar ise sezgisel algoritmaların üzerinde yer alan ve sezgisel algoritmaların hangisinin verilen problemin çözümü için daha uygun olacağına karar veren bir üst düzey algoritmalar topluluğudur. Bu algoritmaların en önemli özellikleri arasında yaptığı yerel aramalarda önceki bulunduğu değerleri hafızasında tutabilme yeteneğine sahip olmaları gelmektedir. Bu hafızada tutma işlemini gerçekleştirirken komşuluğunda bulunan elemanların almış olduğu değerleri de göz önünde bulundurdıklarından ve bu sayede arama uzayında daha etkili çözümler sunabildiklerinden dolayı son yıllarda yaygın olarak kullanımları mevcuttur.

Arařtırmacılar bu algoritmaları çeřitli problemlere uygulamıřtır. Son yıllarda bařta bilgisayar bilimleri olmak üzere, elektronik, inřaat, mekanik gibi alanlardaki optimizasyon problemlerinin çözümünde bu algoritmaların kullanımını görmek mümkündür. Bu algoritmaların kullanılma nedenleri arasında klasik matematiksel yöntemlerle oluřabilen hesaplama karmařıklıđını azaltmaları ve bu sayede kısa zamanda etkili sonuçlar verebilmeleri gelmektedir. Birçok optimizasyon problemi birden fazla çözüme sahiptir. Bu çözümler arasından en iyisinin seçilmesi için kullanılacak yöntemin iyi seçilmesi gerekmektedir. Bu nedenle verilen optimizasyon probleminin çözümü için seçilecek algoritmada yerel ve global aramalar arasında uyumun sađlandığı bir yapı olmalıdır. Bu uyum sađlanmazsa arama uzayı büyük olan optimizasyon problemlerinde çözümlerin bulunması zorlařır (Said vd. 2014).

Optimizasyon tekniklerinin çođu bařlangıç çözümünden bařlayarak belirli bir iterasyona göre yaklaşık çözümlerin aranmasını iřlemine gerçekteřiren sayısal tekniklerdir (Rao, 1996).

Optimizasyon problemlerinin çözümünde seçilecek olan optimizasyon tekniđi problemimizin tipine göre farklılık göstermektedir. Örneđin dođrusal olmayan (non-linear) bir problem için global arama yapan bir meta-sezgisel algoritmanın seçilmesi gerekmektedir (Yang, 2013).

1.1. Tez Çalışmasının Amacı ve Kapsamı:

Bu tez çalışmasının amacı, her alandaki optimizasyon problemlerinde kullanılabilen meta-sezgisel algoritmaları incelemek ve tabiattan esinlenilerek oluřturulan algoritmalar arasına girebilecek olan dođada cırcır böceđi olarak bilinen bir böcek türünün davranıřlarının taklidi yoluyla geliřtirmiş olduđumuz yeni bir meta-sezgisel algoritma yaklařımını tanıtmaktır.

1.2. Literatür Taraması ve Deđerlendirilmesi:

Optimizasyon teknikleri konusunda literatürde yüzlerce sezgisel, meta-sezgisel algoritma bulunmaktadır. Bu tez çalışması kapsamında incelenen yayınların geneline bakıldıđında bu algoritmaların çođunluđunun popülasyon tabanlı algoritmalarından oluřtuđunu söylemek mümkündür. Bu tez çalışmasına konu olan Cırcır Böceđi Algoritması (CBA)'da popülasyon temelli olup, meta-sezgisel algoritmalar arasına girebilecek yeni bir yaklařım sunulup sunulamayacađı konusunda yaptığımız arařtırmalar neticesinde cırcır böceklerinin dođadaki davranıřlarının modellenmesi sonucu ortaya çıkan bir algoritmadır. Tezin ilerleyen bölümlerinde CBA ve meta-

sezgisel algoritmalara detaylı olarak yer verilecek ve literatür deęerlendirilmesi yapılacaktır.

1.3. Tez Çalışmasının Organizasyonu:

Tez ařaęıda verilen řekilde organize edilmiřtir;

İkinci bölümde optimizasyon tekniklerinden olan meta-sezgisel algoritmalar üzerinde durularak sosyal tabanlı, fizik tabanlı, biyoloji tabanlı, kimya tabanlı, müzik tabanlı, sürü tabanlı, spor tabanlı ve matematik tabanlı olmak üzere 9 farklı kategoride sınıflandırılan sezgisel yöntemler açıklanacaktır.

Üçüncü bölümde tabiattaki cırcır böceęinden esinlenilerek modellenmesi yapılmıř Cırcır Böceęi Algoritması (CBA) kapsamlı olarak açıklanacak ve algoritmanın oluřturulma safhasında doęadaki sesin yayılması ile alakalı, algoritmada kullanılan temel bilgiler ve fiziki kanunlar basitçe anlatılacaktır.

Dördüncü bölümde algoritma, tüm meta-sezgisel algoritmaların performans testlerinde kullanılan ortak Benchmark fonksiyonları üzerinde çalıştırılarak performans deęerlendirilmesi yapılacak ve sonuçlar gösterilecektir.

Beřinci bölümde mühendislik problemlerinden en bilinen 5 problem üzerinde algoritma çalıştırılarak, literatürde bu problemler üzerinde çalıştırılmıř meta-sezgisel algoritmaların sonuçları ile CBA'dan elde edilen sonuçlar karşılaştırılmalı olarak verilecektir.

Altıncı bölümde CBA, imge işleme uygulamalarında yaygın olarak kullanılan entropi temelli eşikleme ve imge sıkıřtırma yöntemleri üzerinde maksimum entropinin bulunması problemlerinde kullanılarak bu uygulamalardaki performansı gösterilecektir.

Son bölümde ise CBA hakkında genel bir deęerlendirmenin yapılacaęı sonuç ve tartışma kısmı yer alacaktır.

2. META-SEZGİSEL ALGORİTMALAR

Meta-sezgisel algoritmalar sezgisel yöntemlerin üzerinde yer alan ve problemlerin çözümünde hangi yöntemin kullanılacağına karar veren kapsamlı algoritmalar olarak karşımıza çıkmaktadır. Bu algoritmaların kullanılabilir olması için belli başlı ölçütleri sağlaması gerekmektedir. Bu ölçütlerin başında buldukları çözümlerin optimum değere yakınlığı ve bu çözümlerin elde edilmesinde harcadıkları zaman gelmektedir. Algoritmanın herkes tarafından anlaşılabilir şekilde kodlanmış olması ve analiz kolaylığı sağlıyor olması da algoritmaların seçiminde önemli bir faktördür. Bu algoritmaları birçok şekilde gruplara ayırıp sınıflandırmak mümkündür. Burada birkaç örnek vermek gerekirse bu algoritmalar;

- Kullandığı komşuluk yapısı,
- Çözümlerin hafızada tutulması,
- Esinlenen kaynaklar,
- Aramada kullanılan çözüm sayısı,
- Kullanılan amaç fonksiyonu,

gibi özelliklere göre sınıflandırılabilir. Bu özellikler arasından en çok kabul göreni mevcut bilginin arama sırasında kullanılıp kullanılmadığıdır. Meta-sezgisel algoritmaları diğer optimizasyon tekniklerinden ayıran en önemli farkın bu olduğu rahatlıkla söylenebilir. Bu algoritmalar literatürde optimizasyon problemlerinin çözümünde sıklıkla karşılaştığımız teknikler arasında yer almaktadır. Bu algoritmaların önemli bir çoğunluğu popülasyon temelli olan sürü zekasını kullanan algoritmalar olmaktadır. Popülasyon temelli meta-sezgisel algoritmalar ortak olan bazı noktalar bulunmaktadır. Örneğin; bu algoritmalar rastgele popülasyon değerlerinin oluşturulması ile başlar. Bu rastgele işlemi tüm değişkenler için verilen alt ve üst limitler içerisinde değerlerin oluşturulması işlemidir. Algoritmanın başında algoritmanın ihtiyacı olan parametrelerin belirlenmesi gerekir. Daha sonra bu değerler uygunluk fonksiyonuna gönderilerek popülasyondaki her bir birey için uygunluk değeri hesaplanır. Bu değerler hesaplandıktan sonra, her algoritmaya ait olan adımlar izlenerek istenilen minimum veya maksimum değer için algoritma çalıştırılmaktadır. Bu süreç bazı algoritmada belli bir iterasyona kadar, bazılarında ise belli bir tolerans değerine kadar devam etmektedir. Bu değerlere ulaşıldığında ise algoritma sonlanmakta ve sonuçlar kullanıcıya gösterilmektedir.

Sezgisel yöntemler Şekil 1.1’de gösterildiği gibi sosyal tabanlı, fizik tabanlı, biyoloji tabanlı, kimya tabanlı, müzik tabanlı, sürü tabanlı, spor tabanlı, matematik tabanlı ve melez olmak üzere 9 farklı kategoride değerlendirilmektedir. Bunlardan melez kategorisinde bir veya birden fazla kategorideki algoritmalar bir arada kullanılarak çözüm arayışı sağlanmaktadır. Bu kategorilerde literatürde yüzlerce algoritma bulunmaktadır. Bu tez çalışmasında, bilinen en güncel algoritmalarından bazıları anlatılmaktadır.

2.1. Sosyal Tabanlı Algoritmalar

Sosyal tabanlı algoritmalar, hayattaki sosyal olaylardan esinlenme sonucu ortaya çıkmış algoritmalarlardır. Temelinde insan ve insanların sosyal hayattaki davranışlarının olduğu bu algoritmaların en bilinenleri Emperyalist Yarışmacı Algoritması (Gargari ve Lucas, 2007), Öğretme-Öğrenme Tabanlı Optimizasyon Algoritması (Rao ve Patel, 2013), Beyin Fırtınası Optimizasyon Algoritması (Zhan vd. 2012), Parlamenter Optimizasyon Algoritması (Altunbey ve Alataş, 2015), Sosyal Tabanlı Algoritma (Ramezani ve Lotfi, 2013) gibi algoritmalarlardır.

Emperyalist Yarışmacı Algoritmada, ülkeler ve ülkelere bağlı koloniler bulunmaktadır. Algoritmanın başında ülkeler arasından en iyileri emperyalist ülkeler olarak seçilmektedir. Diğer ülkeler ise bu en iyi ülkelere bağlı olan koloniler olarak yer almaktadır. Algoritmanın çalışması süresince ülkeler güçlerini artırmak için bir yarışa girişmekte ve bu yarış sonunda tek bir ülke kalana dek yarışma devam etmektedir. Yarışmanın sonunda zayıf ülkeler elenerek, en güçlü ülkenin kolonisi olmaktadır. En güçlü ülke en iyi olarak kabul edilmekte ve optimum çözüm olarak sunulmaktadır.

Öğretme-öğrenme Tabanlı Optimizasyon Algoritması, okuldaki konuların öğretmen tarafından öğretilme ve öğrenciler tarafından öğrenme yetenekleri baz alınarak geliştirilmiş bir algoritmadır. Bu algoritmada oluşturulan popülasyon içerisindeki en iyi çözüm öğretmen olarak kabul edilmektedir. Öğretme ve öğrenme süreçlerinin sonunda en iyi öğretmene sahip öğrencilerin notları da iyi olmaktadır.

Beyin fırtınası bilinen tabiriyle belirli bir konuda fikir sahibi olanların fikirlerini sunduğu ve bu sunum sonunda var olan problem için en iyi çözümün elde edildiği bir ortamdır. Beyin fırtınası optimizasyon algoritması da bu mantıkla verilen problem için sunulan tüm fikirler arasından en iyi çözümün elde edildiği bir algoritma olarak sunulmaktadır.

Parlamentar Optimizasyon Algoritması, günümüz parlamenter sistemlerinden esinlenilmiş sosyo-politik bir algoritmadır. Parlamenter sistemlerde bir meclis bulunmakta ve bu mecliste bazı partiler yer almaktadır. Bu partiler seçim dönemlerinde birbirleriyle yarışmakta ve yarış sürecinde partilerin belirlemiş olduğu üyeler halk tarafından seçilmektedir. Aday üye belirleme süreci ise grup içi yarışma aracılığıyla yapılmaktadır. Algoritmada grup içi yarışma sonucuna göre her grubun gücü belirlenmektedir. Daha sonra gruplar arasında grup dışı yarışma yapılarak en güçlü gruplar belirlenmekte ve birleştirme işlemi gerçekleştirilmektedir. En güçsüz gruplar ise silinmektedir. Algoritmanın sonunda en iyi aday optimizasyon probleminin çözümü olarak kabul edilmektedir.

Sosyal Tabanlı Algoritma, Evrimsel algoritma ve Emperyalist Yarışmacı algoritmalarının birleştirilmesiyle oluşturulmuş bir algoritmadır. İnsanlar monarşi, cumhuriyet, otokrasi ve çok uluslu topluluklarda yaşadıklarından dolayı her topluluktaki liderlik stili de farklı olmaktadır (Altunbey ve Alataş, 2015). Bu algoritma yaklaşımında en iyi lider vasfına sahip olan bireyin seçilerek diğer ülkelerin bu bireyin liderliğinde birleştirilmesi fikri benimsenmektedir. Bu aşamada rastgele bazı bireyler lider olarak seçilmekte ve emperyalist maliyet fonksiyonuna göre imparatorluklar oluşturulmaktadır. Daha sonra güçlü imparatorluklar tarafından diğer ülkeler asimile edilerek liderleri ile beraber kendi imparatorluklarına taşıma işlemi gerçekleştirilmektedir. Algoritma sonunda en güçlü imparatorluk yarışı kazanmış sayılmaktadır.

Yukarıda kısaca açıklamaya çalışılan algoritmalar, son yıllarda literatürde karşılaşılan algoritmalarlardır. Bunların yanında Sosyal Duygusal Optimizasyon Algoritması (Cui vd. 2010), Grup Liderleri Optimizasyon Algoritması (Pooranian vd. 2013), Tabu Arama Algoritması (Glover, 1989) gibi algoritmalarda sosyal tabanlı algoritmalar arasında yer almaktadır.

2.2. Fizik Tabanlı Algoritmalar

Doğadaki fizik olaylarından esinlenen fizik tabanlı algoritmaların en bilinenleri Elektromanyetizma Algoritması (Birbil ve Fang, 2003), Büyük Patlama-Büyük Büzülme Algoritması (Erol ve Eksin, 2006), Yerçekimsel Arama Algoritması (Rashedi vd. 2009), Yüklü Sistem Arama Algoritması (Kaveh ve Talatahari, 2010), Yapay Fizik Optimizasyon Algoritması (Xie vd. 2009), Akıllı Su Damlaları Algoritması (Shah-Hosseini, 2008), Merkezi Kuvvet Optimizasyonu (Formato, 2007;

Green vd. 2012), Benzetilmiş Tavlama (Kirkpatrick vd. 1983), Su Döngüsü Algoritması (Eskandar vd. 2012) gibi algoritmalarıdır.

Elektromanyetizma Algoritması, elektromanyetizma teorisindeki elektrik yüklü parçacıklar arasındaki etkileşimden kaynaklanan itme-çekme olayından esinlenilerek ortaya çıkmış bir algoritmadır. Bu algoritmaya göre, yüklü parçacıklar çözüm uzayındaki bir noktayı temsil ettiği gibi her bir parçacık amaç fonksiyonu ile orantılı bir yüke de sahiptir. Parçacıklar sahip oldukları yüke göre, birbirine karşı itme ya da çekme kuvveti uygulamaktadır. Parçacıkların her birine etkiyen vektörel kuvvetlerin ayrı ayrı hesaplanarak bileşke kuvvetin hesaplanması sonucu tüm parçacıkların bu kuvvet yönünde ve doğrultusunda hareket ettirilmesi sağlanmaktadır. Algoritma boyunca bu hareket sayesinde optimum çözüme ulaşılmaya çalışılmaktadır. Bu algorithma, popülasyondaki her bir birey üzerinde diğer bireylerin tek tek etkisi bulunmaktadır (Birbil ve Fang, 2003; Gürsu, 2014; Özdağ ve Karcı, 2015). Bu algoritmanın ana fikri itme veya çekme kuvveti uygulayarak optimum çözüme yönelik parçacıkları taşımaktır.

Büyük Patlama-Büyük Büzülme Algoritması, evrenin oluşumunda meydana geldiği düşünülen iki sürecin birleşiminin modellenmesi sonucu ortaya çıkmış bir algoritmadır. Algoritmanın büyük patlama sürecinde evrenin oluşumu sırasında yaşanan büyük patlamadan sonra ortaya çıkan enerjinin sönmülmesinin rastgele oluşumu temel alınırken, büyük büzülme aşamasında rastgele dağılan parçacıkların bir araya gelme işlemi ele alınır. Algoritmanın başında çözümlerin rastgele dağıtımı büyük patlamadan sonra parçacıkların dağılımına benzetilmektedir. Daha sonra Büyük Büzülme aşamasına geçilir. Bu aşamada rastgele dağılan parçacıkların bir araya gelmesi sonucu oluşacak olan kütle merkezi hesaplanır. Algoritma sonlanma ölçütüne kadar yeni yeni patlamalar ve bu patlamaların sonucunda kütle merkezinin yeniden hesaplanmasına kadar devam etmektedir. Kütle merkezinin değişmediği durumlarda en son hesaplanan kütle merkezi en iyi çözüm olarak sunulmaktadır. Algoritmanın çalışması bittiğinde tüm parçacıkların (çözümlerin) bir kütle merkezi etrafında toplandığı kabul edilmektedir (Erol ve Eksin, 2006).

Yerçekimsel Arama Algoritması, Newton'un hareket ve yerçekimi kanunu prensiplerine göre (Rashedi vd. 2009) tarafından ortaya çıkarılmış bir algoritmadır. Algorithma her nesnenin belirli bir kütlesi vardır ve nesnelere birbirini yerçekimi kuvveti ile çekerler. Bu çekme işlemi ile nesnelere kütlesi en ağır olan nesnelere doğru hareket ederler. Ağır olan nesnelere yavaş hareket ettiklerinden tüm nesnelere kendisine

çekmektedir. Optimizasyon alanında düşünülduğünde her nesne bir çözümdür ve bu çözümlerin ağırlıkları performanslarını göstermektedir. Çözümlerin ağırlığı en yüksek olana doğru yakınsaması algoritmanın bitimine kadar devam etmektedir. Algoritmanın sonunda ağırlığı en yüksek olan çözüm global çözüm olarak kabul edilmektedir (Rashedi vd. 2009; Can, 2014).

Yüklü Sistem Arama Algoritması, elektrostatik ve Newton'un mekanik yasaları temelli bir optimizasyon algoritmasıdır. Bu algorithmada yüklü parçacıklar Coulomb yasası göz önünde bulundurularak birbirlerine uygunluk fonksiyonu değerleri ve aralarındaki mesafeye göre etki etmektedir. Bilindiği üzere Coulomb yasasına göre yüklü parçacıklar birbirlerini ya iter ya da çekerler. İki parçacığın arasındaki kuvvet aralarındaki mesafenin karesiyle ters orantılıdır (Kaveh ve Talatahari, 2010). Yükler arasındaki bu prensiplere dayanarak her yüklü parçacık problemin çözümü olarak değerlendirmekte ve yükler arasındaki işlemler sonucunda global çözüm elde edilmeye çalışılmaktadır. (Kaveh ve Talatahari, 2010).

2.3. Biyoloji Tabanlı Algoritmalar

Literatürde yaygın olarak kullanımları mevcut olan biyoloji tabanlı algoritmalar genellikle doğadaki canlıların davranışlarının taklidi yoluyla ortaya çıkmış algoritmalar olduğundan dolayı doğadan esinlenen algoritmalar olarak da bilinmektedir. Doğadaki her türlü biyolojik sistemleri esas alan bu algoritmaların en bilinenleri Genetik Algoritmalar (Goldberg, 1989), Bakteriyel Besin Arama Optimizasyon Algoritması (Passino, 2002), Biyo-coğrafya Temelli Optimizasyon (Simon, 2008), Diferansiyel Gelişim Algoritması (Storn ve Price, 1997), Atmosfer Bulut Model (Yan ve Hao, 2013) gibi algoritmalarlardır.

Genetik Algoritmalar (GA), John Holland tarafından geliştirilmiş biyolojik temelli bir algoritmadır. Algoritma parametre olarak kromozomları kullanır. Kromozomlar problem için çözümleri temsil etmektedir. Kromozomların bir araya gelmesi ile popülasyon oluşturulur. Kromozomların aldığı uygunluk değerleri verilen problemin hedef fonksiyonu ile belirlenmektedir. GA üç tip genetik operatör içerir: seçim, çaprazlama ve mutasyon. Seçim operatörü kromozomun aldığı uygunluk değerine göre yüksek değerli kromozomların seçilmesini sağlamaktadır. Seçilen kromozomlar yeni neslin oluşturulmasında kullanılmaktadır. Çaprazlama operatörü seçilen kromozomlar arasından yavru kromozomların elde edilmesi için çaprazlama işlemini gerçekleştirmektedir. Mutasyon operatörü ise belirli bir aşamadan sonra

neslin tekrarlanmaması amacıyla kromozomların bazılarını mutasyona uğratmak için kullanılmaktadır. Bunların yanında bu çaprazlamanın veya mutasyonun ne sıklıkla yapılacağını belirleyen bazı olasılık değerleri bulunmaktadır. Algoritmanın amacı kromozomlar üzerinde yapılan işlemlerden sonra optimum sonuca ulaştıracak yeni neslin oluşturulmasını sağlamaktır. Bu amaçla algoritmanın sonunda en son nesil değerleri optimum değerler olarak sunulmaktadır.

Bakteriyel Besin Arama Optimizasyon Algoritması, bakterilerin besin arama sürecini esas alan bir sezgisel algoritmadır. Optimizasyon döngüsünde kemotaksis, eliminasyon ve dağılma adlı üç olay yer alır. Bunlardan kemotaksis olayında bakterinin mevcut pozisyonunu koruyup korumayacağına diğer bir ifadeyle hareket yönü ve adım uzunluğuna karar verilir. İkinci olay üreme olayıdır. Bu olayda belirli sayıda bakteri alınarak popülasyondan çıkarılmaktadır. Kalan bakteriler değerlendirme ölçütlerine göre sıralanmakta ve sıralamanın son yarısındaki bakteriler yeniden popülasyondan çıkarılmaktadır. Kalan diğer yarıdaki bakteriler ise kopyalanmaktadır. Son olarak eliminasyon ve dağılma olayında her bir bakteri için belirli rassal sayı üretilmekte, bu sayının daha önce belirlenen bir değerden daha küçük olması durumunda bakterilerin (çözümlerin) eliminasyonu sağlanmaktadır. Eliminasyon olayına uğrayan bir bakteri için algoritmada yeni bakteri üretilmektedir. Bu sayede en iyiye ulaşılmaya çalışılmaktadır (Hezer ve Kara, 2013; Passino, 2002).

Biyo-coğrafya Temelli Optimizasyonda yer alan bio-coğrafya canlıların bir yaşama alanından diğerine göç etmesini, çoğalmasını ve soyunun tükenmesini matematiksel olarak tanımlamaktadır. Bu algoritma canlıların en iyi yaşam alanını belirleyebilmesi esasına dayanan bir algoritmadır. Verilen problem uygun yaşam alanını temsil etmektedir. En iyi şartların belirlenmesi için yaşam alanı uygunluk indeksi ve yaşanabilir alanın olup olmadığını belirlemeyi sağlayan uygunluk indeks değişkenleri kullanılmaktadır. Yaşam alanı uygunluk indeksinin büyük olduğu durumlarda daha fazla canlının yaşayabileceği aksi halde soyun tükeneceği sonucu çıkarılmaktadır. Algoritmada ayrıca göç etme ve mutasyon işlemleri de gerçekleştirilmektedir. Algoritmanın sonunda en uygun yaşam alanını sağlayan değişken değerleri global en iyi çözümler olarak sunulmaktadır (Simon, 2008).

2.4. Kimya Tabanlı Algoritmalar

Kimya tabanlı algoritmalar doğadaki kimyasal olaylar göz önünde bulundurularak önerilmiş sezgisel algoritmalarlardır. (Karcı, 2012) tarafından önerilen Yapay Atom Algoritması atomlar arasındaki kovalent ve iyonik bağ ilişkilerinden esinlenilmiştir. (Alataş, 2012) ise kimyasal tepkimelerdeki tepkenler kullanılarak ideal tepkimenin oluşturulması esasına göre Yapay Kimyasal Tepkime Optimizasyon Algoritmasını geliştirmiştir. Kimya tabanlı önemli optimizasyon algoritmalarından olduğundan dolayı burada kısaca açıklanmaya çalışılacaktır.

Yapay Atom Algoritması, (Karcı, 2012) tarafından atomlardaki kovalent ve iyonik bağ işlemlerinin modellenmesi sonucunda önerilen sezgisel bir algoritmadır. Algoritmada problem bir atom olarak düşünülmekte, problemi sağlayan her parametre elektronlar olarak tasarlanmaktadır. Başlangıçta birden fazla atom ile işleme başlanmakta ve bu çözümler kümesine atom kümesi denilmektedir. Önerilen algoritma iki tane işlem içermektedir. Kovalent bağ işleminde hangi elektronların en az iki atom arasında ortaklaşa paylaşılacağı belirlenmekte, iyonik bağ işleminde ise, hangi elektronların değerinin değiştirilmesi gerektiği belirlenmektedir. İyonik bağ işleminin amacı, etkisi kötü olan elektronlar için yeni değer üretme işlemini gerçekleştirmektir. Bu işlem sırasında elektronlar arasında kovalent alanındaki elektronlardan daha iyi bir değere sahip olanı değerlendirilmektedir. Bütün atomlar için her elektronun amaç fonksiyonu üzerindeki etkileri hesaplanmakta ve algoritma sonunda en iyi değere sahip çözüm global en iyi olarak sunulmaktadır (Karcı, 2012).

Yapay Kimyasal Tepkime Optimizasyon Algoritması, kimyevi tepkimelerin oluşumundan ve tiplerinden esinlenilerek oluşturulmuş bir algoritmadır. Algoritmanın ilk safhasında tepkimenin tek moleküllü veya iki moleküllü olup olmayacağına karar verilmektedir. Bu seçime göre tepkime için tepkenler oluşturulmaktadır. Bu algoritmada tepkenler ikili (binary) veya yazı (string) şeklinde kodlanabilmektedir. Oluşturulan tepkenler bir takım kimyevi reaksiyonlardan geçirilir. Daha sonra kimyevi kararlılık testi yapılarak tepkenlerden iyi değer elde edilmediği kontrol edilmektedir. İyi değerleri sağlayan tepken setleri alınmakta, kötü değer sağlayanlar ise dışlanmaktadır. Buradaki tepkenler optimizasyondaki çözümleri, tepkime ise hedef fonksiyonu temsil etmektedir. Sonlanma ölçütü gerçekleştiğinde fonksiyonda en iyi değerleri sağlayan çözümler (tepkenler) global en iyiler olarak sunulmaktadır.

Algoritma daha az parametre içermekte ve kimyasal tabanlı algoritmalara göre daha etkili çözümler sağlamaktadır (Alataş, 2011).

2.5. Müzik Tabanlı Algoritmalar

Müzik tabanlı en bilinen algoritma (Geem, 2001) tarafından ortaya çıkarılan Armoni Arama Algoritmasıdır. Bu algoritma bir orkestradaki elemanların en iyi melodiyi elde edebilmeleri için çaldıkları notalar ile belirli bir armoniyi yakalayabilmeleri prensibine dayanır. Günlük hayattan da bildiğimiz gibi bir orkestranın belirli bir armoniyi elde edebilmesi için tüm orkestra elemanlarının uygun notaları çalmaları gerekmektedir. Ancak bu sayede düzgün melodilerin ortaya çıkması sağlanmaktadır. Algoritmaya baktığımızda her elemana (çözüm) ait belirli sayıda nota bulunmakta ve bu notalar bir araya gelerek en iyi melodinin çalınması amaçlanmaktadır. Bu amaçla armoninin sağlanması için belirli ölçütlerin kontrol edilmesi gerekmektedir. Algoritmada oluşturulan armoninin önceki değerler ile karşılaştırılması için bir armoni hafızası bulunmakta, bu hafızanın dikkate alınıp alınmayacağına karar veren bir oran yer almaktadır. Ayrıca ton ayarlamasına karar veren bir oran ile armoniler her defasında güncellenerek en iyi melodiye ulaşılmaya çalışılmaktadır. Algoritmanın sonunda en iyi melodiyi sağlayan notalar (çözümler) global en iyi olarak sunulmaktadır (Geem, 2001; Kamışoğlu ve Karaboğa, 2012).

2.6. Spor Tabanlı Algoritmalar

Spor tabanlı en bilinen algoritma Lig Şampiyonası Algoritmasıdır (Kashan, 2009). Bu algoritma sanal bir ligdeki takımların birbirleri ile bir şampiyonada karşılaşmasını taklit eden meta-sezgisel bir algoritmadır. Şampiyona sezon sonuna kadar devam etmektedir. Algoritmanın başında lig programı rastgele oluşturulmaktadır. İki takım oyun gücüne göre ya galip gelmekte ya da yenilmektedir. Takımlar sezon sonuna kadar yapacağı transferler ile oyun güçlerini arttırmaya çalışmaktadır (Bingöl ve Alataş, 2015). Algoritmada takımlar problemin çözümü için aday çözümleri temsil etmektedir. Algoritmanın sonunda takımların birbirleriyle yaptığı karşılaşmalar sonunda en iyi takım global çözüm olarak sunulmaktadır.

2.7. Matematik Tabanlı Algoritmalar

Matematik tabanlı algoritmalar meta-sezgisel ve matematik programlama tekniklerinin birleştirilmesi ile oluşturulmuş algoritmalar (Boschetti vd. 2009). Bu

algoritmaların amacı meta-sezgisel algoritmadaki yerel arama işlemlerinin matematiksel teknikler ile daha verimli hale gelmesini sağlamaktır.

2.8. Sürü Tabanlı Algoritmalar

Doğadaki canlıların çoğunluğu sürüler halinde yaşarlar ve hayatlarının tüm evrelerinde sürüyü takip ederler. Sürü tabanlı algoritmalarda sürü halinde yaşayan bazı hayvan topluluklarının hareketlerinin modellenmesi sonucu ortaya çıkan algoritmalar. Aslında biyoloji temelli olarak sınıflanan yukarıda saydığımız birçok algoritma sürü tabanlı algoritmalar sınıfına da girmektedir. Bu algoritmalar en bilineni Parçacık Sürü Optimizasyonudur (PSO) (Kennedy ve Eberhart, 1995). Birçok algoritma çözüm arayışlarında yaptıkları işlemlerde bu algoritmayı temel almışlardır. Bunun haricinde Kedi algoritması (Chu vd. 2006), Karınca Kolonisi Algoritması (Dorigo ve Stützle, 2004), Yapay Arı Kolonisi (Karaboğa ve Baştürk, 2007), Guguk kuşu Arama Algoritması (Yang ve Deb, 2009) gibi birçok sürü tabanlı algoritma bulunmaktadır. Burada önermiş olduğumuz algoritmamız ile ortak özellikler taşıdığından dolayı PSO, Ateş Böceği Algoritması (ABA), Yarasa algoritmaları (YA) biraz daha detaylı incelenecektir.

Kedi Algoritması, kedilerin yiyecek arama işlemlerinde yapmış olduğu hareketlerin taklidi yoluyla geliştirilmiş bir sezgisel algoritma yaklaşımıdır. (Chu vd. 2006) tarafından bu süreç iki aşamada gerçekleştirilmektedir. İlk olarak arama sürecinde kediler avları için tetikte beklemektedir. Bu süreç kedinin hedefi gördükten sonraki pozisyonunu belirlemeye yaramaktadır. Algoritmanın arama sürecinde dört önemli faktör bulunmaktadır. Bunlar; arama hafıza havuzu, seçilen boyutun arama aralığı, değişen boyutların sayısı, kendi pozisyonunu dikkate alma işlemi. Öncelikle arama hafıza havuzunda yer alan tüm kediler için çözümler kopyalanmaktadır. Değişen boyutların sayısına göre seçilen boyutun arama aralığı değerlerinin yüzdesi rastgele artırılmakta veya azaltılmaktadır. Daha sonra aday çözümler için uygunluk fonksiyonu hesaplanmaktadır. Tüm uygunluk fonksiyon değerlerinin eşit olmaması durumunda her aday çözüm için seçilme olasılığı hesaplanıp, kedilerin rastgele bir noktaya yönlendirilmesi işlemi gerçekleştirilmektedir. Algoritmanın ikinci aşamasında ise izleme süreci gerçekleştirilmektedir. Bu süreçte ilk olarak kedilerin sahip olduğu hız değerleri güncellenmektedir. Hız değerleri algoritmada belirtilen maksimum hız değerinin üzerine çıktığı durumlarda bu değerlerin limitlere çekilmesi sağlanmaktadır. Daha sonra bu hız değerleri kullanılarak kedilerin pozisyonları güncellenmektedir.

Algoritma bitirme ölçütüne ulaştığında en iyi pozisyona sahip kedilerin pozisyon değerleri global en iyi olarak sunulmaktadır.

Karınca Kolonisi Optimizasyonu, doğadaki karıncaların yuvalarına yakın olan besin kaynaklarını arama süreçlerini yapay olarak modelleyen bir sezgisel arama yöntemidir. İlk olarak Dorigo doktora tezinde karıncaların yiyecek arama yöntemlerini kapsayan bir çalışma yapmıştır. Daha sonra (Dorigo ve Stützle, 2004) tarafından bu hareketlerin modellenmesi yapılarak literatüre yeni bir meta-sezgisel yöntem kazandırılmıştır. Bu yöntemin esas amacı sürü halinde yaşayan karınca topluluğunun yiyecek arama işlemi sırasında yardımlaşarak yiyeceğe en kısa yoldan ulaşma işlemlerini simüle etmektir. Karıncalar yuvalarından çıktıktan sonra etrafta rastgele dolaşarak yiyecek aramaktadırlar. Yiyeceklerini bulduklarında ise diğer karıncalarında bu yiyeceğe ulaşmasını sağlayacak bir yöntem geliştirmişlerdir. Bu yöntemde karıncalar buldukları yiyeceğin kalitesi ve miktarına göre yollara feromon adlı bir kimyasal madde bırakmaktadır. Bu maddenin miktarına göre diğer karıncalar yiyeceğe ulaştıracak en kısa yolu bulmaktadırlar. Algoritmada öncelikle tüm karıncalar için rastgele bir yol güzergahı belirlenmektedir. Daha sonra yol güzergâhlarındaki feromon konsantrasyonuna göre yiyeceğe en kısa yoldan ulaşılmaya çalışılmaktadır. Feromon belirli bir süre sonunda kaybolan bir madde olduğundan dolayı algoritmada bu ölçütte göz önünde bulundurulmaktadır. Algoritmanın sonunda en kısa mesafeyi sağlayan çözümler en iyi çözüm olarak verilmektedir. En kısa yol esasına dayandığından dolayı algoritmanın gezgin satıcı problemlerinde yoğun olarak kullanımı mevcuttur.

Yapay Arı Kolonisi Algoritması, arıların bal yapımında kullanacakları nektar için kovan dışında yaptıkları arama işlemi ve nektarı bulduktan sonraki süreçte nektar hakkında kovandakilere bilgi verme süreçlerini içeren bir meta-sezgisel algoritma yaklaşımı olarak literatürde yerini almıştır. Algoritma başlangıç yiyecek kaynağı bölgelerinin üretilmesi ile başlamaktadır. Bu bölgeler belirtilen alt ve üst limitlere göre rastgele üretilmektedir. Bu bölgelerin üretilmesinden sonra her bir bölge için aynı sayıda işçi arı görevlendirilmektedir. Bu işçi arıların görevi bölgesinin komşuluğunda yer alan yeni yiyecek kaynaklarının belirlenmesi ve kalitesinin değerlendirilmesini sağlamaktır. Belirlediği kaynağın daha iyi olması durumunda bu yeni kaynağı hafızasına almaktadır. Belirlenen kaynağın kalitesi uygunluk fonksiyonu yardımıyla bulunmaktadır. Bulunan yeni kaynağın daha iyi olması durumunda eski kaynak bilgisi hafızadan silinerek yola yeni kaynak bilgisi ile devam edilmektedir. Algoritmada mevcut kaynak bilgisinin geliştirilemediği durumlarda geliştirememe sayacı (failure)

denen bir sayaç bulunmaktadır. Bu sayaç için maksimum değer algoritmanın başında belirlenmektedir. Çözümün geliştirildiği durumlarda sayaç sıfırlanmakta aksi halde maksimuma değere kadar sayaç artırılmaktadır. Algoritmada bir çevrim gerçekleştikten sonra görevli arılar kovana dönerek kovadaki arıları kaynaklardaki nektar miktarları hakkında bilgilendirmektedir. Bu bilgilendirmeden sonra gözcü arılar tarafından nektarlarla orantılı olarak elde edilen uygunluk değerleri kullanılarak olasılıksal bir seçme işlemi gerçekleştirilmektedir. Bu süreçte olasılık değerlerinin hesaplanmasından sonra $[0,1]$ aralığında rastgele bir sayı üretilmekte ve üretilen bu değer olasılık değerleri ile karşılaştırılmaktadır. Olasılık değerlerinin bu sayıdan büyük olması durumunda gözcü arılar tarafından kaynak bölgesi için yeni bir çözüm üretilmektedir. Elde edilen yeni çözümlerin uygunluk fonksiyonuna göre değerlendirilmesi yapılmakta, daha iyi değerlerin elde edilemediği durumlarda yeniden geliştirememeye sayacı çalıştırılmaktadır. Bu süreç gözcü arıların tümünün bölgelerine dağılmalarına kadar devam etmektedir. Algoritmada kullanılan sayaçlar nektarların tükenip tükenmediği konusunda bilgi sahibi olunmasını sağlamaktadır. Nektarların tükendiği durumlarda görevli arılar kaşif arı olarak değerlendirilmekte ve rastgele çözüm arama süreci gerçekleşmektedir. Ayrıca algoritmada kaynağın terk edildiğini belirleyen “limit” adlı bir eşik değeri parametresi bulunmaktadır. Bu arama süreci algoritmanın durdurma ölçütüne kadar devam etmekte ve sonuçta da en iyi çözümler sunulmaktadır (Akay, 2009).

2.8.1. Parçacık sürü optimizasyonu

PSO, 1995 yılında Kennedy ve Eberhart adlı bilim adamlarının kuş ve balık sürülerinin davranışlarını incelemesi sonucunda ortaya çıkardığı bir meta-sezgisel algoritma yaklaşımıdır (Kennedy ve Eberhart, 1995). Algoritmanın ortaya çıkma safhasında kuş sürülerinin yiyecek arama işlemlerini gerçekleştirdiği sırada yaptığı davranışlar incelenmiştir. Kuşların sürünün başındaki kuşu izlemeleri diğer bir deyişle en baştaki kuşun yiyeceğe daha yakın veya en iyi olabileceği düşüncesi ile hareket ettiği belirlenmiştir. Bu davranışlar modellenerek algoritmanın temeli oluşturulmuştur. Bu algoritmada her parçacık (kuş veya balık) bir çözüm olarak belirlenmiştir. Uygulamada kuşların en iyiye yani yiyeceğe yakın olma olasılıkları verilen bir uygunluk fonksiyonu aracılığıyla hesaplanmaktadır.

Algoritma basit olarak aşağıdaki adımlardan oluşmaktadır (Parsopoulos ve Vrahatis, 2010).

- Sürü temelli olduğundan başlangıçta rastgele popülasyon oluşturulur.
- Bu popülasyondaki her parçacığın ilk değerleri belirlenir.
- Parçacıklar uygunluk fonksiyonuna gönderilerek parçacıkların ilk uygunluk değerleri hesaplanır.
- Bu değerler arasından minimum değere sahip olan parçacık seçilerek parçacığın sahip olduğu pozisyon değeri yeni g_{best} değeri olarak atanır.
- Artık bu yeni g_{best} değerine göre parçacıklar yeni pozisyon ve hız değerini günceller.
- Verilen şartlara göre iterasyona devam edilir.
- İterasyonun sonunda en iyi g_{best} değeri ve bu değer sağladığı minimum uygunluk fonksiyonu bulunmuş olur.

Basit olarak D adet parametreden oluşan n adet parçacık olduğunu varsayarsak bu aşağıdaki matris şeklinde ifade edilebilmektedir.

$$x = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}_{n \times D}$$

Yukarıdaki matrise baktığımızda i . parçacık için $x_i = [x_{i1} + x_{i2} + \dots + x_{id}]$ matrisi kullanılır. Bu matristeki değerler verilen uygunluk fonksiyonundaki değişkenlerin alt ve üst limiti gözetilerek başlangıçta rastgele üretilir. Rastgele üretildiklerinden dolayı algoritmanın çalışma süresinde bu değerlerin limit dışına çıkmasını önlemek için uygun tanımlamalar yapılmalıdır. Örneğin alt limitten küçük bir değere ulaşması durumunda değişkenin değerinin alt limitteki değere eşitlenmesi, üst limitteki değerden büyük bir değer elde edilmesi durumunda ise üst limite eşitlenmesi gerekmektedir. En iyi uygunluk değerini veren i . parçacığın için $p_{best_i} = [p_{i1} + p_{i2} + \dots + p_{id}]$ matrisi kullanılır. g_{best} değeri tüm parçacıkların sahip olduğu p_{best} değerleri arasından seçilir ve tüm parçacıklar için bir tanedir.

$$gbest = [p_1 + p_2 + \dots + p_d]$$

Son olarak i . parçacığın hızı $v_i = [v_{i1} + v_{i2} + \dots + v_{id}]$ olarak ifade edilir. (Kennedy ve Eberhart, 1995) PSO'nun ilk sürümünde aşağıdaki denklemleri kullanmışlardır.

$$v_i^{k+1} = v_i^k + c_1 \cdot rand_1^k (pbest_i^k - x_i^k) + c_2 \cdot rand_2^k (gbest^k - x_i^k) \quad (2.1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2.2)$$

$$i=1,2,\dots,D, j=1,2,\dots,n$$

Formüllerden de anlaşılacağı gibi her parçacığın kendine has bir hız değeri v_i ve bir x_i pozisyonu bulunmaktadır. Algoritmanın temelinde parçacıkların arasında bilgi paylaşımı olması düşüncesi yatmaktadır. Buna göre her parçacık kendisi için en iyi pozisyonu p_{best} ve sürü içindeki en iyi parçacığa ait pozisyonu g_{best} hatırlamak zorundadır. Bu p_{best} değerleri arasından $f(x)$ fonksiyonunu minimum yapan değer g_{best} değeri olarak kabul edilir. Buradaki c_1 ve c_2 değerleri öğrenme faktörleri olarak kullanılmaktadır. Bunlar parçacıkları p_{best} ve g_{best} değerlerine çeken sabit terimler olarak bilinmektedir. Bu değerler küçük alındığında parçacıkların hedefe ulaşmadan önce hedeften uzak yerlerde dolaşılmasına sebep olduğu gibi, yüksek seçilmeleri durumunda parçacıkların hedefe hızlı ulaşmasını sağladığı fakat bazen hedefi es geçebildikleri görülmüştür. Bu yüzden literatürde $c_1=c_2=2$ olarak alındığında optimum değerleri sağladığı görülmektedir. r_1 ve r_2 sayıları $[0,1]$ aralığında rastgele düzgün dağılımlı sayılardır. Bu sayılarda parçacıkların boyutlarını sınırlamak için algoritmaya eklenmiş sabitlerdir. k iterasyon sayısını, i ise parçacığa ait indisi gösterir (Çevik ve Koçer, 2013). Başlangıçta p_{best} değerleri rastgele üretilen pozisyon değerlerine eşittir. Algoritma çalıştırıldıktan sonra bir önceki iterasyonda elde edilmiş olan p_{best} ile karşılaştırılır ve yeni değeri atanır. Eğer t . iterasyondaki değer $(t+1)$ iterasyonundaki değerden küçükse p_{best} değeri değişmezken aksi durumda yeni $p_{best}(t+1)$ iterasyonundaki değer olarak kabul edilerek algoritmaya devam edilir.

$$p_i(t+1) = \begin{cases} x_i(t+1), & \text{eğer } \dots f(x_i(t+1)) \leq f(p_i(t)), \\ p_i(t), & \text{değilse} \end{cases}$$

Mevcut versiyonun basit optimizasyon problemlerinde etkili sonuçlar verdiği fakat zor problemlerde optimum sonuçlara ulaşamadığı görülmüş, bu amaçla PSO'nun kullandığı denklemler aynı kalmak koşulu ile bu denklemlere yeni eklentiler yapılarak

değişik versiyonları tanımlanmıştır. Burada yapılan değişikliklerden sadece bir tanesi gösterilecektir. Değişiklik yapılan sürümünde bazı parametre değerleri için yeni tanımlamalar yapılmış ve birde w atalet sabiti eklenmiştir. Literatürde yaygın olarak kullanılan PSO sürümü (Shi ve Eberhart, 1998) tarafından önerilen atalet ağırlığının kullanıldığı sürümdür. Atalet ağırlığı hızla ilişkilendirilmiş bir ölçekleme faktörüdür (Alataş, 2007). Yeni hız güncelleme denklemi Denklem 2.3'te gösterilmektedir (Parsopoulos ve Vrahatis, 2010).

$$v_{ij} = wv_{ij}(t) + c_1r_1(p_{ij}(t) - x_{ij}(t)) + c_2r_2(p_{gj}(t) - x_{ij}(t)) \quad (2.3)$$

Buradaki w değeri Denklem (2.4)'deki gibi hesaplanmaktadır.

$$w(t) = w_u - (w_u - w_l) \frac{t}{T_{max}} \quad (2.4)$$

Buradaki w_u ve w_l atalet ağırlığının sırasıyla başlangıç ve bitiş değerlerini, t mevcut iterasyon değerini, T_{max} ise maksimum iterasyon değerini göstermektedir.

PSO yapay zeka, imge işleme, mühendislik problemleri, sistem dizayn problemleri gibi birçok optimizasyon gerektiren alanlarda kullanılmış ve kullanılmaya devam etmektedir. Algoritma ortaya atıldığında sürü psikolojisiyle hareket eden birçok hayvan türünün davranışlarının incelenmesi ve modellenmesi konusunda ilham kaynağı olmuştur. Bu sayede literatürde sıklıkla karşılaştığımız Ateş Böceği, Yarasa gibi birçok algoritma bilim adamları tarafından ortaya atılmış ve optimizasyon problemlerinin çözümünde kullanılmaya devam edegelmiştir. Halen PSO geliştirilmekte olan sürümleri ile birçok alanda kullanılmaya devam etmektedir.

2.8.2. Yarasa algoritması

Yarasalar çıkardıkları sesin çevresindeki engellere çarpıp yansıması sonucu yollarını bulan veya bu sayede avlarına ulaşabilen bir hayvan türüdür. Bu davranışları radar sistemlerinin geliştirilmesinde model olarak alınmıştır. X. Yang bu davranışları modelleyerek yeni bir meta-sezgisel algoritma ortaya çıkarmıştır (Yang, 2010).

Yarasalar çevrelerine çok yüksek frekansta ses gönderirler ve etrafındaki nesnelere çarpıp geri dönen sesi dinleyerek yönlerini tayin ederler. Aynı zamanda ses ile avlarının yerini tespit ederek beslenme ihtiyaçlarını karşılarlar. Yarasalar hedefleri ile arasındaki mesafeyi ve avlarının hızlarını tespit edebilen mucizevi canlılardır.

Yarasa algoritmasının temel esasları şunlardır;

1. Bütün yarasalar çıkardıkları ses ile mesafe kontrolü yapabilmektedir. Bu sayede avlarını ve çevresindeki engelleri ayırt edebilmektedirler.
2. Yarasalar v_i hızında, x_i pozisyonunda sabit frekans ve değişen dalga boyunda ve A_0 ses yüksekliğinde rastgele uçarak avlarını ararlar. Hedeflerine yakınlıklarına göre otomatik olarak dalga boylarını ayarlayabilirler.
3. Sesin yüksekliği çeşitli yollarla değişebilmesine rağmen bu algortmada A_0 ile A_{min} arasında değişir.

Bu algortmada $[f_{min}, f_{max}]$ aralığında frekans ve aynı şekilde $[\lambda_{min}, \lambda_{max}]$ aralığında dalga boyu kullanılmaktadır.

Yarasa algortması hareket denklemleri:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (2.5)$$

$$v_i^t = v_i^{t-1} + (x_i - x_*)f_i \quad (2.6)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (2.7)$$

Burada $f_{min}=0$ ve $f_{max}=100$ alınmaktadır. x_* tüm yarasalar içerisinde en iyi lokasyona sahip olan yarasanın koordinatını temsil etmektedir. $\beta [0,1]$ aralığında rastgele dağılım gösteren değerlerden oluşmaktadır. Yarasalar hedeflerine yaklaştıklarında çıkardıkları ses miktarını azaltırlar. Bu da algortmada A parametresiyle gösterilmektedir. Yarasalar rastgele yürüyüş sergilediklerinde yerel (local) arama için Denklem (2.8) kullanılmaktadır. Bu rastgele yürüyüş mevcut en iyi değer etrafında bir yürüyüş olmaktadır.

$$x_{new} = x_{old} + \epsilon A^t \quad (2.8)$$

$\epsilon [-1,1]$ aralığında rastgele sayı iken A^t tüm yarasaların çıkardıkları sesin kullanıldığı zamandaki ortalama değerini temsil etmektedir. $A_{min}=0$ olması yarasaların avlarını buldukları anlamına gelmektedir.

Yang'ın makalesinde kullandığı bir ifade CBA'nın geliştirilmesinde ilham kaynağı olmuştur (Yang, 2010). Yang yarasa algortmasını tanıttığı yayınında algortmanın oluşturulmasında mevcut algortmaların iyi yönlerinin alınıp algortmaya

uyarlanması üzerinde durmaktadır. Bu amaçla kendisi de PSO algoritmasındaki denklemleri kendi algoritmasına uyarlamıştır.

2.8.3. Ateş böceği algoritması

Ateş Böceği Algoritması isminden de anlaşılacağı üzere ateş böceklerinden esinlenilmiş bir algoritmadır. Ateş böcekleri geceleri görülen yaydıkları ışık ile tanınan bir böcek türüdür. Ateş böceklerinde yaydıkları ışık miktarı en yüksek olan ateş böceği diğer ateş böceklerini kendisine çeker. Yaydıkları bu ışık miktarına bağlı olarak diğer ateş böceklerinin yönelmesi işlemi algoritmanın oluşturulmasında temel alınmıştır. Bu algoritmanın da bir takım genel prensipleri vardır (Yang, 2010b). Bunlar;

1. Ateş böcekleri cinsiyetsizdirler. Bu sayede diğer ateş böceklerini cezbederek kendilerine çekebilirler.
2. Bu çekicilik için yaydıkları yüksek ışığı kullanırlar. Yaydıkları ışık oranı çekicilikleri ile doğru orantılıdır. En yüksek ışığa sahip ateş böcekleri diğer ateş böceklerini kendilerine çeker. Yüksek ışığa sahip olunmadığı durumlarda rastgele hareket edilir.
3. Ateş böceklerinin sahip olduğu parlaklık oranı optimizasyon problemlerindeki uygunluk fonksiyonun değeri olarak alınmaktadır.

Bu algoritmada önemli olan iki unsur vardır. Bunlar ateşböceğinin çekiciliğinin ve bu çekiciliği sağlayan ışık yoğunluk değişim miktarının formülize edilmesidir. En basit ifadeyle konumu x ve parlaklığı (I) olan bir ateş böceğinin uygunluk fonksiyonu ile parlaklık ilişkisi $I(x) \propto f(x)$ ile ifade edilebilir. Çekicilik ise göreceli bir kavram olduğundan dolayı herkes tarafından veya algoritmada yer alan ateş böcekleri tarafından farklı algılanabilir. Algoritmada bu kavram için i ve j konumundaki iki ateş böceği arasındaki mesafe (r_{ij}) kullanılarak çekicilik (β) değerlendirilmesi yapılmaktadır. Bu değerlendirme işleminde mesafenin artması nedeniyle ışık yoğunluğunun düşmesi ve ışığın ortam tarafından emilmesi gibi durumlarda göz önünde bulundurulmaktadır. Algoritmada Denklem 2.9'da belirtilen ters kare yasasından yola çıkılarak ateş böceklerinin çekicilikleri formülize edilir.

Ters kare yasasına göre ışık yoğunluğu değişimi

$$I(r) = \frac{I_s}{r^2} \quad (2.9)$$

denklemleri ile ifade edilir. Ters kare yasası, ışığın sabit emilim katsayısı γ ve r mesafesi kullanılarak Denklem 2.10'daki Gauss ifadesi elde edilmektedir.

$$I(r) = I_0 e^{-\gamma r^2} \quad (2.10)$$

Buradaki I_0 $r=0$ durumundaki ilk ışık yoğunluğudur. Çekicilik oranı ışık yoğunluğuna bağlı olduğundan algoritmada çekiciliğin hesaplanmasında Denklem 2.11'deki formül kullanılmaktadır.

$$\beta = \beta_0 e^{(-\gamma r^2)} \quad (2.11)$$

Koordinatları i ve j olan iki ateş böceği arasındaki r mesafesi ise Denklem 2.12'deki gibi Öklid uzaklığı kullanılarak hesaplanmaktadır.

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2.12)$$

Ateş böceklerinin en yüksek ışık yoğunluğuna sahip olan ateş böceğine doğru yönelim hareketleri için Denklem 2.13 kullanılmaktadır.

$$x_i^{t+1} = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \epsilon_i^t \quad (2.13)$$

Bu denklemdaki ikinci ifade yukarıda açıklandığı gibi çekicilik formülünden gelmektedir. Üçüncü ifadede yer alan α değeri rastlantısal parametre olarak kullanılır. ϵ değeri ise $[0,1]$ aralığında rastgele sayı üreticidir (Belen vd. 2014).

3. ÖNERİLEN YÖNTEM: CIRCIR BÖCEĞİ ALGORİTMASI

3.1. Cırcır Böceği Davranışları

Cırcır böcekleri yaz mevsiminde sıklıkla karşılaştığımız çıkardıkları sesler ile bilinen böcek türüdür. Çıkardıkları bu ses birçok manaya gelmektedir. Erkek cırcır böcekleri çıkardıkları bu ses ile dişi cırcır böceklerini çiftleşmek için kendilerine çekmektedirler. Bu sesin diğer bir amaçla kullanımı ise başka cırcır böceklerinin tehlikelere karşı uyarılmasında kullanılmasıdır. Cırcır böceklerinin dişileri en yüksek sese sahip cırcır böceğine doğru yönelmeye başlarlar (Brown, 1999; Stephen ve Hartley, 1995). Cırcır böcekleri bu sesleri vücutlarının ön yüzeyinde bulunan kanatları aracılığıyla çıkarmaktadır. Bu sesler cırlama olarak bilinmektedir. Cırcır böceklerinde bu sesleri alan iki tanede anten bulunmaktadır. Cırcır böceklerindeki yüksek sese doğru bu yönelim hem PSO'daki sürünün yiyeceğe doğru yönelimine hem de ateş böceği algoritmasında ateş böceklerinin en yüksek ışığı yayan ateş böceklerine doğru yönelimlerine benzer. Ayrıca ses ile iletişime geçtiklerinden dolayı bu özellikleri yarasa algoritmasındaki yarasaların hareket özelliklerine benzer. Bu da geliştireceğimiz yaklaşımın bu algoritmalara benzerliğini göstermektedir. Matematiksel manada her dişi cırcır böceği fonksiyonu sağlayan bir aday çözüm olarak ele alındığında fonksiyonu minimum veya maksimum yapan değerlerin bulunabileceği bir optimizasyon modellemesi yapmak mümkün olabileceği düşünülebilir. Bu modellemede cırcır böceklerine has birçok özellik kullanılmaktadır.

Amos Dolbear cırcır böceklerini incelerken bir hipotez ortaya atmıştır (Dolbear, 1897). Hipoteze göre cırcır böceklerinin kanat çırpış sayısı ile havanın sıcaklığı arasında bir ilişki bulunmaktadır. Bu hipotezi Denklem (3.1) ile ifade etmiştir. Denkleme bakıldığında sıcak havalarda kanat çırpış sayısının daha fazla soğuk havalarda daha az olacağı görülmektedir. Diğer bir deyişle cırcır böceğinin çıkardığı ses ile havanın sıcaklığı arasında doğru orantı mevcuttur. Dolbeardan sonra bu hipotezi kanıtlayan birçok araştırma yapılmıştır (Larsen ve Lemone, 2009). Ayrıca cırcır böceklerinin davranışları konusunda literatürde birçok yayın bulmak mümkündür. Bu yayınlardan yola çıkarak tezimizde cırcır böceği davranışları göz önünde bulundurularak modellemeye çalıştığımız yeni yaklaşımın biyolojik temelini sağlam olduğu söylenebilmektedir (Hedwig, 2006; Hedrick ve Mulloney, 2004; Hedrick ve Kortet, 2006). Bu böcekler ses ile iletişime geçtiklerinden dolayı sesin doğadaki yayılımı ile alakalı fiziki kanunları göz önünde bulundurma gerekliliği ortaya

çıkmiştir. Bu sayede geliştireceğimiz yaklaşımın gerçek hayat şartlarına uygunluğu sağlanmış olacaktır. Bu amaçla diğer bölümde sesin doğadaki yayılımı ile alakalı fiziki kanunlar açıklanacak ve yaklaşımda kullandığımız denklemler gösterilecektir.

$$T_F = 50 + \left(\frac{N - 40}{4} \right) \quad (3.1)$$

Burada T_F Fahrenayt derece cinsinden havanın sıcaklığı, N ise 1 dakika içindeki kanat çırpış sayısı

$$T_C = 10 + \left(\frac{N - 40}{7} \right) \quad (3.2)$$

T_C Santigrat derece cinsinden havanın sıcaklığını ifade etmektedir.

3.2. Doğadaki Sesin Yayılımı ile Alakalı Temel Bilgiler

Önceki bölümlerde belirttiğimiz gibi cırcır böcekleri ses ile iletişime geçtiklerinden dolayı algoritmamızın doğaya yakınlığını arttırmak amacıyla kullandığımız sesin doğadaki yayılımı ile alakalı fiziki kanunlar ve ses ile alakalı genel terimler bu bölümde açıklanacaktır (Crocker, 2008; Howard ve Angus, 2009).

3.2.1. Sesin hızı

Tabiattaki her türlü ses bize hava aracılığı ile taşınmaktadır. Taşınan bu seslerin taşındıkları ortamın durumuna göre belirli bir hıza sahip olması gerekir. Algoritmamızda Dolbear yasasına göre elde edilen sıcaklık değeri kullanılarak sesin hızı hesaplanmakta ve bu hız değeri cırcır böceklerinin koordinatlarının güncellenmesinde kullanılmaktadır. Havanın sıcaklık, yoğunluk durumu sesin yayılma hızını etkiler. Soğuk hava ses hızını azaltıcı etki yapar. Ayrıca ses sıcak havadan soğuk havaya geçerken yayılma doğrultusunda değişiklik meydana gelir.

$$V = 20.1 * \sqrt{273 + C} \quad (3.3)$$

C : Santigrat Derece

3.2.2. Sesin frekansı ve dalga boyu

Ses frekansı ses kaynağı tarafından oluşturulan basınç dalgalarının birim zamanda (genellikle 1 sn) uğradıkları değişim veya devir sayısıdır. Başka bir ifadeyle

saniyedeki titreşim sayısıdır. Sesin yüksekliği frekans ile doğru orantılıdır. Frekans ses yüksekliğinin ölçüsü olarak bilinmektedir.

$$f = \frac{V}{\lambda} \quad (3.4)$$

f : frekans, V : sesin hızı, λ : dalga boyu

Dalga boyu bir ses dalgasının oluşması için sesin aldığı yol olarak tanımlanmaktadır. Ses titreşimlerinin dalga boyları yaklaşık 0,02 m ile 20 m arasında değişir. Algoritmamızda dalga boyu için cırcır böceklerinin en iyi koordinata sahip cırcır böceği ile arasındaki mesafe farkı değerleri kullanılacaktır.

3.2.3. Havanın sesi emme katsayısı

Algoritmamızda bu katsayının hesaplanması için ISO 9613-1 (ISO, 1993) uluslararası standardı kullanılmaktadır. Bu standard, herhangi bir kaynaktan yayılan sesin, atmosfer tarafından değişik meteorolojik şartlar altında soğurulması sonucunda sesin azalmasının analitik bir yöntem ile hesaplanması için oluşturulmuş teknikleri içermektedir. Bu standart göz önünde bulundurularak havanın sesi emme katsayısının hesaplanması için yazılan MATLAB kodu Ek 1’de verilmektedir. Bu katsayı cırcır böceklerinin koordinat güncellemelerinde hangi yolun izleneceğine karar verme aşamasında yardımcı olmaktadır.

3.2.4. Sesin şiddeti

Ses kaynakları tarafından üretilen sesler farklı uzaklıklardan değişik şekillerde duyulurlar. Sesin bu özelliği ses şiddeti olarak tanımlanır. Her sesin şiddeti farklı farklı olmaktadır. Ses şiddeti ışık şiddetinde olduğu gibi uzaklığın karesiyle ters orantılı olarak azalır. Bu durum Ters Kare Yasası olarak bilinir.

$$I = \frac{W}{4\pi r^2} = \frac{p^2}{\rho c} \quad (3.5)$$

I : Sesin şiddeti, W :Sesin gücü, r : mesafe, ρ : ortamın yoğunluğu, c : sesin yayılma hızı, p : ses basınç seviyesi, $\rho = 1$ olarak kabul edilecektir.

Buradan sesin gücü

$$W = I * 4\pi r^2 \quad (3.6)$$

hesaplanabilir. Desibel veya dBA ile ifade edilir.

3.2.5. Ses gücü seviyesi

Sesin yayılımında sesin iletilmesi, yansınması, soğurulması sonucunda kalan ses enerji miktarının seviyesidir. Ses kaynaklarının birim zamanda yayımladıkları ses enerjisi, ses gücü (L_w) ile Watt ya da mW cinsinden belirlenir.

3.2.6. Ses basıncı seviyesi

Bilindiği üzere ses kulak zarında havanın basıncının yarattığı değişimler sonunda algılanır. Ses titreşimlerinin yapmış olduğu bu değişimlere akustik basınç denir. Ses basıncı ses kaynağından gelen ses enerjisi ölçüsüdür, desibel veya dBA ile ifade edilir.

Sesin yansıtılacağı alanlarda hiç bir engelin bulunmadığı durumlarda, ses gücü düzeyi L_w olan bir ses kaynağının, kaynaktan belli bir mesafede yaratacağı ses basınç düzeyi L_p Denklem 3.7'deki gibi hesaplanır.

$$L_p = L_w + 10 * \log[Q / (4\pi r^2)] \quad (3.7)$$

L_w : Kaynağın ses gücü düzeyi (dB), Q : Yönelme katsayısı yön faktörü (Q ,engebeli arazide 1, düz arazide 2 alınır), r : Kaynaktan olan uzaklıktır(m.)

3.2.7. Atmosferin sesi yutusu

Sesin havada yayılımı esnasında hava molekülleri tarafından ses bir miktar azalır. Bu duyulan sesin miktarının ses kaynağından çıktığı haliyle değil belirli bir miktar azalmış haliyle duyulması anlamına gelir.

$$A_{atm} = 7.4(f^2 r / \emptyset)10^{-8} \quad (3.8)$$

eşitliğinden bulunabilir. Burada; f = İletilen sesin frekansı (Hz), r = Kaynaktan olan uzaklık (m), \emptyset = Havanın bağıl nemi (%) (%50 olarak alınacaktır)

3.2.8. Serbest alanda gerçek ses basınç düzeyi

Sesin havada yayılımı esnasında atmosfer tarafından bir miktar yutulmasından sonraki ses basınç düzeyi L_p' ,

$$L_p' = L_p - A_{atm} \quad (3.9)$$

şeklinde hesaplanmaktadır (Aydoğdu, 2015). Algoritmamızda cırcır böceklerinin koordinat değerlerinin güncellenmesinde matematiksel ifadeyle yeni çözümlerin elde edilmesinde bu değer kullanılmaktadır.

3.3. Cırcır Böceği Algoritması

Bu yeni yaklaşım (Canayaz ve Karcı, 2013;2015a) sürü temelli olduğundan dolayı verilen problem için uygun değer aralıklarında popülasyon oluşturularak algoritmaya başlanır. Bu değerler hedef fonksiyona gönderilerek cırcır böceklerinin ilk ses şiddeti değerleri (S_i) elde edilir. Elde edilen bu değerler arasından en küçük olanı F_{min} olarak atanır ve bu değeri sağlayan değişken değerleri de en iyi değerler (x_{best}) olarak ele alınır. Bu tür algoritmalarda sonlanma iki şekilde gerçekleşmektedir. Birincisi F_{min} değeri belirli bir tolerans değerinin üstünde olduğu sürece algoritma çalışmaya devam etmekte, ikinci olarak ise belirli bir iterasyona kadar algoritmanın çalışması sürdürülmektedir. Algoritmamızda test fonksiyonlarının optimum değerlerinin bulunmasında belirli bir tolerans değeri sonlandırma kriteri olarak kullanılırken, diğer problemlerde belirli bir iterasyon belirlenmektedir. Cırcır böcekleri kanat çırpışları ile ses ürettiklerinden dolayı algoritmamızda $[0,120]$ değer aralığında rastgele değerler üretilecek (N_i) ve bu değerler kullanılarak Dolbear yasasına uygun şekilde hava sıcaklığı (T_i) tahmini yapılacaktır. Bu sıcaklık değeri Denklem 3.3'te yerine konularak o sıcaklıktaki sesin hızı (V_i) hesaplanır. Bu hız değeri ve dalga boyu λ_i ile Denklem 3.4'teki frekans değeri elde edilir. Elde edilen bu frekans değerleri yaklaşımımızda iki farklı yerde kullanılacaktır. İlk olarak yarasa algoritmasında da kullanılan frekansa bağlı denklem olan Denklem 3.10'da kullanılacaktır.

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (3.10)$$

$$v_i^t = v_i^{t-1} + (x_i - x_*)f_i + \mathbf{V}_i \quad (3.11)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3.12)$$

Denklem (3.10, 3.11, 3.12) YA'daki yarasaların koordinat değerlerinin güncellendiği diğer bir deyişle yeni çözümlerin elde edildiği denklemlerdir (Yang, 2010a). Bu denklemlerden özellikle Denklem 3.11'de yarasa algoritmasındakinden farklı olarak algoritmamızda Dolbear yasası kullanılarak tahmin edilen T_i sıcaklığında sesin yayılma hızını ifade eden yeni bir V_i değeri daha eklenmektedir. Denklem 3.11'deki $(x_i - x_*)$ cırcır böceğinin en yüksek sese sahip cırcır böceği ile arasındaki mesafeyi (dalga boyu), v_i^{t-1} ise cırcır böceğinin bir önceki hızını gösterir. Yaklaşımında f_{min} değeri 0 olarak alınacak, f_{max} değeri ise Denklem 3.4'ten elde edilen

popülasyondaki cırcır böceklerinin her birine ait frekans değeri olarak kabul edilecektir. Elde edilen frekans değeri ikinci olarak havanın sesi soğurma katsayısının hesaplanmasında kullanılacaktır. Bu katsayıyı hesaplayacak olan fonksiyona mevcut frekans değerlerinin ve tahmin edilen sıcaklık değerlerinin ortalaması gönderilerek ISO 9613-1 standardına uygun olarak havanın sesi soğurma katsayısı (γ) hesaplanır. Bu katsayı daha sonra çözümlerin güncellenmesinde kullanılacaktır. Güncelleme sürecinde [0,1] aralığında rastgele sayı üretilerek bu katsayı ile karşılaştırılır. Algoritmanın performansın artırılması konusunda yapmış olduğumuz çalışmalarda bu katsayının kullanılmasında iki farklı yol izlenmiştir. İlkinde üretilen sayının bu katsayıdan büyük olduğu durumlarda en yüksek sese sahip cırcır böceğine doğru yönelim sağlanmıştır. Küçük olduğu durumlarda ise cırcır böcekleri rastgele bir yürüyüş gerçekleştirecek şekilde algoritma çalıştırılmıştır. İkinci kullanımında ise üretilen bu sayı ile soğurma katsayısı karşılaştırılarak cırcır böceklerinin koordinat güncellemelerinde hangi değerlerin kullanılacağına karar verilmektedir. Bu karar verme sürecinde böceklerin koordinatlarının belirlenmesinde ya Denklem 3.12'deki gibi güncelleme yapılmakta ya da Denklem (3.15, 3.16)'daki gibi yönelim gerçekleşmektedir. Yönelimin olmadığı durumlarda ise cırcır böcekleri Denklem 3.13'deki gibi rastgele yürüyüş sergilemektedir.

$$x_i = x_{best} + 0.01 * rand(0,1) \quad (3.13)$$

x_{best} :mevcut en iyi çözüm

Elde edilen çözümler yeniden uygunluk fonksiyonuna gönderilerek her cırcır böceği için S_i değerlerinin yeniden elde edilmesi sağlanır. Daha sonra bu değerler birbirleriyle karşılaştırılır. Böylece en yüksek sese sahip böcek tespit edilerek diğer böceklerin bu böceğe doğru yönelimi gerçekleştirilir. Algoritmamızda sesin doğadaki yayılımı ile alakalı fiziki yasaların uygulanarak gerçek ses şiddetinin hesaplanmak istenmesi yaklaşımımızda esinlenen cırcır böceğinin davranışlarının modellenmesinde gerçek dünyaya uygunluğunun göz önünde bulundurulduğunun kanıtı olmaktadır. Güncellenme aşamasında öncelikle popülasyondaki tüm cırcır böcekleri arasındaki Öklid mesafesi Denklem 3.14'e göre hesaplanır.

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (3.14)$$

Bu mesafe (r) ve sesin şiddeti (S_i) kullanılarak Denklem 3.6'daki gibi sesin gücü hesaplanır. Daha sonra elde edilen bu değer Denklem 3.7'de yerine konularak ses basınç seviyesi (L_p) hesaplanır. Atmosferin havayı tutumunun hesaplanmasında havanın bağıl nem oranı %50 varsayılacak, mevcut frekans ve mesafe de Denklem 3.8'de kullanılarak bu değer elde edilecektir. En sonunda iletilecek olan gerçek ses basınç düzeyi Denklem 3.9 yardımıyla bulunmaktadır. Bu değer iletilen gerçek ses şiddet değeri kabul edilir. Bu şiddet değeri ve havanın sesi absorbe etme katsayısı (γ) kullanılarak ateş böceği algoritmasında (Yang, 2010b) kullanılan Denklem 3.15'teki gibi ters kare yasasına göre cırcır böceğinin çekicilik değeri (K) hesaplanır. Bu değere göre de çözümler Denklem 3.16'daki gibi güncellenir (Yang, 2010b). Güncelleme sonunda elde edilen değerlerin uygunluk fonksiyonuna gönderilmesi sağlanır ve bu sayede yeni uygunluk değerleri elde edilir. Daha sonra elde edilen değerler ile algoritmanın başındaki F_{min} değeri karşılaştırılır. Bu değerden daha küçük bir değer elde edilmesi halinde yeni minimum olarak kabul edilerek optimum sonucu bulana kadar (maksimum iterasyona erişene kadar) döngüye devam edilir.

$$K = K_0 e^{-\gamma r^2} \quad (3.15)$$

$$x_i^{t+1} = x_i + K_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \epsilon_i^t \quad (3.16)$$

γ : havanın sesi absorbe etme katsayısı, $\alpha \epsilon_i$: verilen probleme göre üst ve alt limitlerde ölçekleme için kullanılan katsayılar, x_i : cırcır böceğinin koordinatı (aday çözüm)

Aşağıda algoritmamızda izlediğimiz aşamalar maddeler halinde özetlenmiş ve algoritmamızın akış şeması Şekil 3.1'de, sözde kodu ise Çizelge 3.1'de verilmiştir (Canayaz ve Karcı, 2015a). Ek 2'de ise algoritmaya ait MATLAB kodu verilmektedir.

Adım 1: Verilen problemdeki limitlere uygun ilk popülasyon değerleri rastgele oluşturulur. Bu değerler uygunluk fonksiyonuna gönderilerek böceklerin çıkardıkları sesin şiddeti için ilk değer atamaları yapılır.

Adım 2: Kanat çırpış sayısı rastgele üretilir. Üretilen kanat çırpış sayısı ile havanın sıcaklığı Dolbear yasasına göre elde edilir.

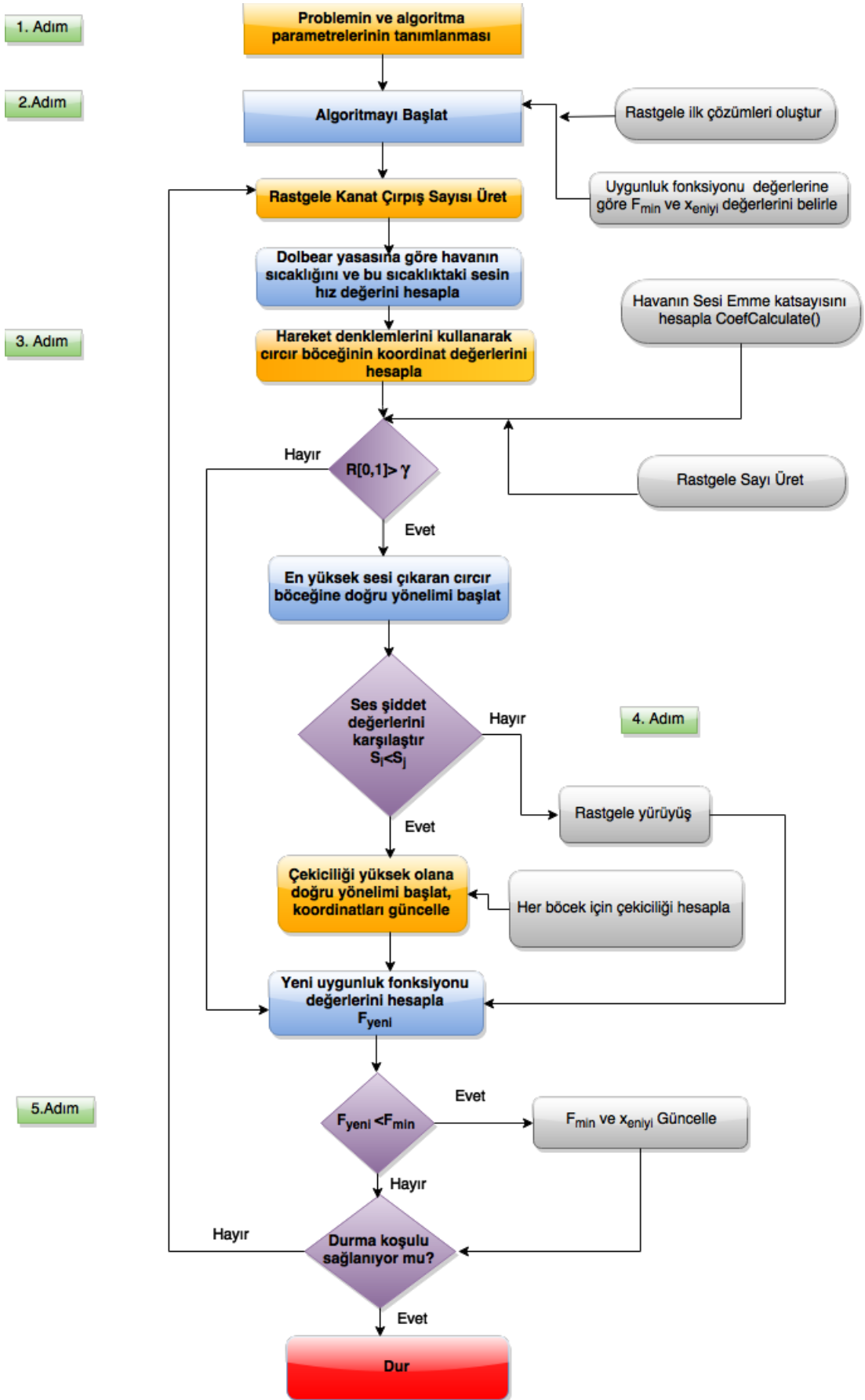
Adım 3: Bu sıcaklıktaki sesin hızı hesaplanır. Bu hızdaki frekans değeri hesaplanır.

Adım 4: Bu frekans ve hız değerlerine göre böceklerin koordinatları güncellenir. Güncellenen bu değerler hedef fonksiyona gönderilir.

Adım 5: Mevcut deęerden daha küçük bir deęer elde edilirse en iyi deęer olarak bu deęer kabul edilir. Aksi takdirde yeni deęerler elde edilerek çözüme devam edilir.

Adım 6: Maksimum iterasyona veya tolerans deęerine ulařana kadar en iyi çözümler aranır.

Adım 7: Algoritmanın sonunda en iyi çözümler ve bu çözümlerden elde edilen uygunluk fonksiyonu deęeri gösterilir.



Şekil 3.1. CBA akış şeması

Çizelge 3.1. CBA sözde kodu

Giris:	
$f(x), x = (x_1, \dots, x_d)^T$	{ Uygunluk Fonksiyonu }
$tol, f_{min}, \alpha, d, n, \beta$	{ Algoritma parametreleri }
Çıkış:	
$x_{i_{min}}, F_{min}$	
begin	
for $i=1$ to n do	
$x_i \leftarrow$ Başlangıç_Çözümlerini_Üret()	
end	
$i_{min} \leftarrow$ argmin $_i$ S(x_i)	
$F_{min} \leftarrow$ argmin S(x_i)	
$x_{i_{min}} \leftarrow$ argmin $_{x_i}$ S(x_i)	
for $i=1$ to t do	
while ($F_{min} > tol$)	
for $i = 1$ to n do	{ Tüm n circır böcekleri }
$N_i \leftarrow$ Rastgele_Kanat_Çırpış_Sayısı_Üret	{ Böcek kanat çırpış sayısı }
$T_i \leftarrow$ Dolbear_Yasası(N_i)	
$C_i \leftarrow (5/9)T_i - 32$	{ Santigrat Derece }
$V_i \leftarrow$ Denklem(3.3), $\lambda \leftarrow x_i - x_{best}$;	
$f_{max} \leftarrow$ Denklem (3.4), $f_i, v_i^t, x_i^t \leftarrow$ Denklem (3.10),(3.11),(3.12)	
$\gamma \leftarrow$ CoefCalculate()	
if rand[0,1] > γ	
for $j=1$ to n do	
if $S_i < S_j$	{ Düşük sese sahip i böceğini yüksek sese sahip j ye yönlendir }
$r_j \leftarrow$ Denklem (3.14)	{ Mesafeyi Hesapla }
$P_s \leftarrow$ Denklem (3.6)	{ Sesi gücü }
$L_p \leftarrow$ Denklem (3.7)	{ Sesi yayılımı }
$A_{atm} \leftarrow$ Denklem (3.8)	{ Atmosferin Sesi Emmesi }
$R_{Lp} \leftarrow$ Denklem (3.9)	{ Boş arazide gerçek ses basınç seviyesi }
$K_o \leftarrow R_{Lp}, K_i \leftarrow$ Denklem (3.15)	{ Çekicilik }
$x_i \leftarrow$ Denklem (3.16)	
end	
end	
else	
$x_i \leftarrow$ Denklem (3.13)	{ rastgele yürüyüş }
end	
$F_{yeni} \leftarrow$ Uygunluk_fonksiyonu(x_i)	
if ($F_{yeni} < F_{min}$)	
$x_{eniyi} = x_i$	
$F_{min} = F_{yeni}$	{ Yeni sonuçları kabul etme }
end	
end	
En iyi çözümü bulma	
end	
end	
end	

4. CBA'NIN TEST FONKSİYONLARINA UYGULANMASI VE BAŞARI ANALİZİ

Algoritmanın performansının ölçülmesi için bir takım problemlere uygulanması gerektiğinden bu bölümde literatürdeki meta-sezgisel algoritmaların performanslarının ölçülmesinde yaygın olarak kullanılan problemler üzerinde uygulama çalıştırılacak ve sonuçlar gösterilecektir. Uygulamanın kodlarına Ek 2'de ulaşılabilmektedir. Uygulama i7 işlemcili 8 GB RAM'a sahip bilgisayar üzerinde çalıştırılmış, uygulama geliştirme platformu olarak MATLAB 2014a kullanılmıştır. Parametre olarak cırcır böceği sayısı=25, tolerans değeri= 10^{-6} , $\alpha=0.5$, $\beta_{min}=0.2$, $f_{min}=0$ değerleri kullanılmıştır.

Algoritma öncelikle Çizelge 4.1'de yer alan standart Benchmark test fonksiyonları üzerinde çalıştırılmıştır (Mishra, 2006; Silagadze, 2004; Sun ve Yuan, 2006). Bu fonksiyonlara ait global minimum değerleri Çizelge 4.2'de verilmektedir. Algoritma sonucu elde edilmiş sonuçlar Çizelge 4.3'te gösterilmektedir. Çizelge 4.3'e bakıldığında önermiş olduğumuz algoritma yaklaşımının bazı fonksiyonlarda global minimum değerlerini sağladığını bazı fonksiyonlarda minimuma yakın değerler sağladığını söylemek mümkündür. Meta-sezgisel algoritmaların performans testlerinde algoritma genelde 30, 50, 100 gibi çalıştırma sayıları ile çalıştırıldıktan sonra elde edilen değerler gösterilmektedir. Çizelge 4.3'te verilen sonuçlara bakıldığında global optimumları sağlayan test fonksiyonları üzerinde CBA tarafından her çalıştırmada aynı değerlerin elde edildiği gözlenmiştir. Diğer test fonksiyonları için verilen sonuçlar 30 kez çalıştırma sonunda elde edilen ortalama değerlerdir.

Çizelge 4.1. Test fonksiyonları

Fonksiyon İsimleri	Fonksiyonlar
Rosenbrock(f_1)	$f(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$
Sphere(f_2)	$f(x) = \sum_{i=1}^n x_i^2$
Parallel Hyper Ellipsoid(f_3)	$f(x) = \sum_{i=1}^n ix_i^2$
Schwefel22(f_4)	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $

Çizelge 4.1 (devamı)

Michalewics(f_5)	$f(x) = -\sum_{i=1}^2 \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$
Powell(f_6)	$f(x) = (x_3 + 10x_1)^2 + 5(x_2 - x_4)^2 + (x_1 - 2x_2)^4 + 10(x_3 - x_4)^4$
Beale(f_7)	$f(x) = (x_1x_2 - x_1 + 1.5)^2 + (x_1x_2^2 - x_1 + 2.25)^2 + (x_1x_2^3 - x_1 + 2.625)^2$
Bird(f_8)	$f(x) = (x_1 - x_2)^2 + e^{[1-\sin(x_1)]^2} \cos(x_2) + e^{[1-\cos(x_2)]^2} \sin(x_1)$
Booth(f_9)	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
Bukin2(f_{10})	$f(x) = 100(x_2 - 0.01x_1^2 + 1) + 0.01(x_1 + 10)^2$
Bukin4(f_{11})	$f(x) = 100x_2^2 + 0.01 x_1 + 10 $
Bukin6(f_{12})	$f(x) = 100\sqrt{ x_2 - 0.01x_1^2 } + 0.01 x_1 + 10 $
Chichinadze(f_{13})	$f(x) = x_1^2 - 12x_1 + 8\sin\left(\frac{5}{2}\pi x_1\right) + 10\cos\left(\frac{1}{2}\pi x_1\right) + 11 - 0.2\frac{\sqrt{5}}{e^{\frac{1}{2}(x_2-0.5)^2}}$
Giunta(f_{14})	$f(x) = 0.6 + \sum_{i=1}^n \left[\sin^2\left(1 - \frac{16}{15}x_i\right) - \frac{1}{50}\sin\left(4 - \frac{64}{15}\right) - \sin\left(1 - \frac{16}{15}x_i\right) \right]$
Goldsteinprice(f_{15})	$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$
HimmelBlau(f_{16})	$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$
Leon(f_{17})	$f(x) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$
Matyas(f_{18})	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$
Mccormick(f_{19})	$f(x) = -x_1 + 2x_2 + (x_1 - x_2)^2 + \sin(x_1 + x_2) + 1$
Penholder(f_{20})	$f(x) = -e \left e^{\left \frac{\sqrt{x_1^2 + x_2^2}}{\pi} + 1 \right } \cos(x_1) \cos(x_2) \right ^{-1}$
Testtubeholder(f_{21})	$f(x) = -4 \left e^{\left \cos\left(\frac{1}{200}x_1^2 + \frac{1}{200}x_2^2\right) \right } \sin(x_1) \cos(x_2) \right $
Zeetl(f_{22})	$f(x) = \frac{1}{4}x_1 + (x_1^2 - 2x_1 + x_2^2)^2$

Çizelge 4.2. Fonksiyon global minimum değerleri

Fonksiyonlar	Global Minimumlar ve Sınır Değerleri
(f ₁)	$f(x_i) = 0$ için $x_i = 1$ $i = 1, \dots, n$ $x_i \in [-5, 10]$
(f ₂)	$f(x_i) = 0$ için $x_i = 0$ $i = 1, \dots, n$ $x_i \in [-1, 1]$
(f ₃)	$f(x_i) = 0$ için $x_i = 0$ $i = 1, \dots, n$ $x_i \in [-1, 1]$
(f ₄)	$f(x_i) = 0$ için $x_i = 0$ $i = 1, \dots, n$ $x_i \in [-100, 100]$
(f ₅)	$f(x_i) = -1.8013$ için $x_i = 0$ $i = 1, 2$ $x_i \in [0, \pi]$
(f ₆)	$f(x_i) = 0$ için $x_i = 0$ $i = 1, \dots, 4$ $x_i \in [-4, 5]$
(f ₇)	$f(x_i) = 0$ için $x_i = [3, 0.5]$ $x_i \in [-10, 10]$
(f ₈)	$f(x_i) = -106.764536$ için $x_i = [4.701055, 3.15294]$ veya $x_i = [-1.582142, -3.130246]$
(f ₉)	$f(x_i) = 0$ için $x_i = [1, 3]$ $x_i \in [-10, 10]$
(f ₁₀)	$f(x_i) = 0$ için $x_i = [-10, 0]$ $x_1 \in [-15, -5], x_2 \in [-3, 3]$
(f ₁₁)	$f(x_i) = 0$ için $x_i = [-10, 0]$ $x_1 \in [-15, -5], x_2 \in [-3, 3]$
(f ₁₂)	$f(x_i) = 0$ için $x_i = [-10, 1]$ $x_1 \in [-15, -5], x_2 \in [-3, 3]$
(f ₁₃)	$f(x_i) = -42.94438$ için $x_i = [6.189866, 0.5]$ $x_i \in [-30, 30]$
(f ₁₄)	$f(x_i) = 0.0644704$ için $x_i = [0.4673200277, 0.467320016]$ $x_i \in [-1, 1]$
(f ₁₅)	$f(x_i) = 3$ için $x_i = [0, -1]$ $x_i \in [-2, 2]$
(f ₁₆)	$f(x_i) = 0$ için $x_i = [3, 2]$ $x_i \in [-6, 6]$
(f ₁₇)	$f(x_i) = 0$ için $x_i = 1$ $i = 1, 2$ $x_i \in [-1.2, 1.2]$
(f ₁₈)	$f(x_i) = 0$ for $x_i = 0$ $i = 1, 2$ $x_i \in [-10, 10]$
(f ₁₉)	$f(x_i) = -1.913222$ için $x_i = [-0.547197, -1.547197]$ $x_1 \in [-1.5, 4], x_2 \in [-3, 4]$
(f ₂₀)	$f(x_i) = -0.9635348$ için $x_i = \pm 9.646167$ $i = 1, 2$ $x_i \in [-11, 11]$
(f ₂₁)	$f(x_i) = -10.872299$ için $x_i = [-\pi/2, 0]$ $x_i \in [-10, 10]$
(f ₂₂)	$f(x_i) = -0.0037912$ için $x_i = [-0.029896, 0.0]$ $x_i \in [-1, 5]$

Çizelge 4.3. CBA'dan elde edilen fonksiyon sonuçları

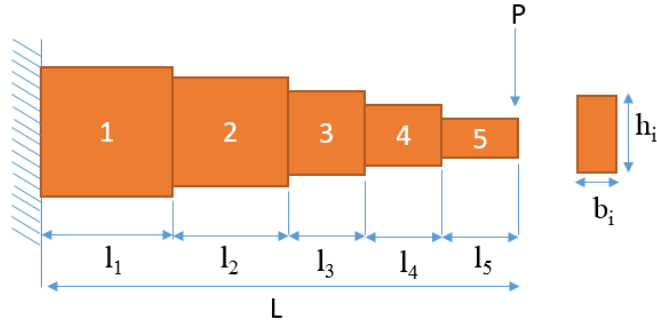
Fonksiyon	x_1	x_2	F_{\min}	Süre(s)
f ₁	0.9993	0.9985	0,0097099	1.6
f ₂	0.00042588	0.00033127	0,0029111	1.22
f ₃	0.00054966	0.00029105	0,0047155	0.225984
f ₄	0.99963	0.99959	0,0065472	0.542095
f ₅	2.2023	1.5707	-1.8016	1.946
f ₆	0.026941	-0.0025964	0.00019521	1.962
f ₇	3.000333	0.500237	0.000001	0.088493
f ₈	4.690590	3.169733	-106.710500	0.007197
f ₉	1.000003	2.999845	0.000000	0.009607
f ₁₀	-10.000156	0.000127	0.000001	0.010159
f ₁₁	-10.000013	0.000077	0.000001	0.009653
f ₁₂	-10.120008	1.024147	0.104632	0.008037
f ₁₃	6.186559	0.516078	-42.941500	0.007895
f ₁₄	0.467268	0.467326	0.064470	0.008650
f ₁₅	0.000896	-1.000103	3.000227	0.008459
f ₁₆	3.000006	2.000164	0.000000	0.008653
f ₁₇	0.999975	1.000046	0.000001	0.011755
f ₁₈	0.000006	0.001959	0.000001	0.008107
f ₁₉	-0.548557	-1.552732	-1.913185	0.006027
f ₂₀	-9.640806	-9.651257	-0.963534	0.009303
f ₂₁	1.580685	-0.027055	-10.867768	0.008487
f ₂₂	-0.030894	-0.007990	-0.003779	0.009591

5. CBA’NIN MÜHENDİSLİK PROBLEMLERİNE UYGULANMASI VE BAŞARI ANALİZİ

Bu bölümde literatürde en bilinen bazı mühendislik problemleri üzerinde algoritma çalıştırılarak elde edilen sonuçlar gösterilecektir.

5.1. Beş basamaklı kiriş optimizasyon problemi:

Bu problem ilk olarak (Thanedar ve Vanderplaats, 1995) tarafından tanıtılmıştır. Şekil 5.1’de gösterilen beş basamaklı konsol kirişin toplam uzunluğu L 500 cm, materyal esneklik katsayısı E 200 GPa’dır. Problemde genişlik için ($x_1 - x_5$) 5, yükseklik için ($x_6 - x_{10}$) 5 olmak üzere toplam 10 adet tasarım değişkeni bulunmaktadır. Ayrıca 11 adet kısıtlayıcı problem mevcuttur. Kısıtlayıcı fonksiyonlardan beşindeki izin verilen gerilim $14,000 \text{ N/cm}^2$ ‘den, izin verilen sapma 2.7 cm ve 5 basamaklı kiriş için en boy oranı ise 20’den az olmalıdır. Bu problemdeki optimizasyonun amacı $P=50000 \text{ N}$ yükü taşıyabilecek kirişin hacmini minimum yapan değerleri bulmaktır (Hsu vd. 2003; Erbatur vd. 2000). Uygunluk fonksiyonu ve kısıtlayıcı fonksiyon değerleri Denklem (5.1-5.12) verilmektedir.



Şekil 5.1. Beş basamaklı kiriş

Uygunluk fonksiyonu:

$$V = l(x_1x_2 + x_3x_4 + x_5x_6 + x_7x_8 + x_9x_{10}) \quad (5.1)$$

Kısıtlayıcı fonksiyonlar:

$$g_1 = \frac{600P}{x_9x_{10}^2} - 14000 \leq 0 \quad (5.2)$$

$$g_2 = \frac{6P(l_5 + l_4)}{x_7x_8^2} - 14000 \leq 0 \quad (5.3)$$

$$g_3 = \frac{6P(l_5 + l_4 + l_3)}{x_5 x_6^2} - 14000 \leq 0 \quad (5.4)$$

$$g_4 = \frac{6P(l_5 + l_4 + l_3 + l_2)}{x_3 x_4^2} - 14000 \leq 0 \quad (5.5)$$

$$g_5 = \frac{6P(l_5 + l_4 + l_3 + l_2 + l_1)}{x_1 x_2^2} - 14000 \leq 0 \quad (5.6)$$

$$g_6 = \frac{Pl^3}{E} \left(\frac{244}{x_1 x_2^3} + \frac{148}{x_3 x_4^3} + \frac{76}{x_5 x_6^3} + \frac{28}{x_7 x_8^3} + \frac{4}{x_9 x_{10}^3} \right) - 2.7 \leq 0 \quad (5.7)$$

$$g_7 = \frac{x_2}{x_1} - 20 \leq 0 \quad (5.8)$$

$$g_8 = \frac{x_4}{x_3} - 20 \leq 0 \quad (5.9)$$

$$g_9 = \frac{x_6}{x_5} - 20 \leq 0 \quad (5.10)$$

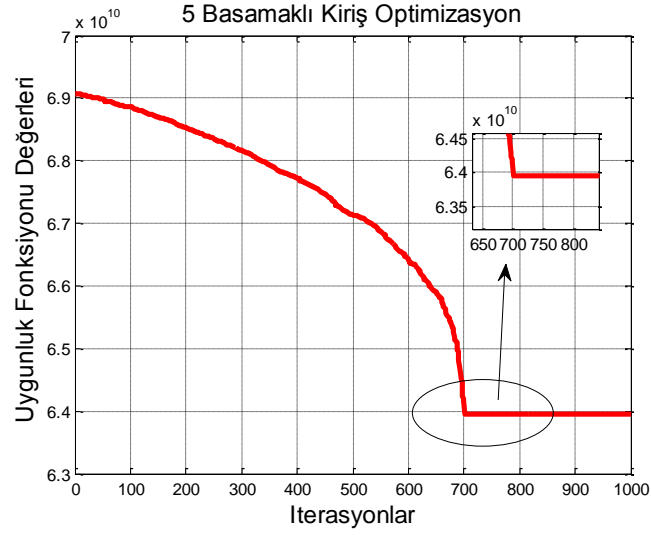
$$g_{10} = \frac{x_8}{x_7} - 20 \leq 0 \quad (5.11)$$

$$g_{11} = \frac{x_{10}}{x_9} - 20 \leq 0 \quad (5.12)$$

Tasarım deęişkenleri alt ve üst limit aralığı aşığıda verilmektedir.

$$1 \leq x_i \leq 5 \quad (i=1,2,\dots,5) \quad 30 \leq x_j \leq 65 \quad (j=6,7,\dots,10) \quad l_i=100\text{cm} \quad (i=1,2,\dots,5)$$

Bu probleme ait elde edilen sonuçlar Çizelge 5.1'de verilmektedir. Ayrıca Çizelge 5.1'de Ateş Böceęi (ABA) ve Benzetilmiş Tavlama (BT) algoritmalarından elde edilen sonuçlarda verilmektedir. Cırcır böceęi algoritmasında 25000 evrimsel gelişim sayısı, 25 cırcır böceęi kullanılmıştır. ABA için ise 20 ateş böceęi ve 10000 gelişim sayısı parametre olarak kullanılmıştır. Bu probleme ait yakınsama grafięi Şekil 5.2'de gösterilmektedir.



Şekil 5.2. Beş basamaklı kiriş yakınsama grafiği

Çizelge 5.1. Beş basamaklı kiriş optimizasyon problemi sonuçları

Tasarım Değişkenleri	ABA	BT	CBA
x ₁	3.03	3.00	3.06
x ₂	2.88	2.97	2.82
x ₃	2.74	2.62	2.61
x ₄	2.21	2.21	2.23
x ₅	1.75	1.75	1.88
x ₆	60.70	59.99	61.27
x ₇	55.33	54.93	56.59
x ₈	50.23	52.49	49.54
x ₉	44.21	44.12	43.84
x ₁₀	35.02	35.03	33.68
g ₁	-0.050	-0.088	-7.377
g ₂	-0.125	-0.099	-52.575
g ₃	-1.01	-1.564	-6.861
g ₄	-0.415	-0.647	-758.132
g ₅	-0.587	-0.121	-959.436
g ₆	-0.005	-0.099	-0.0073
g ₇	-0.002	-0.006	-0.0019
g ₈	-0.002	-0.004	-0.0039
g ₉	-0.004	-0.002	-2.8433
g ₁₀	-0.003	-0.002	-0.9047
g ₁₁	-0.002	-0.001	-4.0997
V	64.105	64.097	63.942

5.2. Üç çubuk makas optimizasyon problemi:

Bu optimizasyon problemi ilk olarak (Ray ve Saini, 2001) tarafından tanıtılmıştır. Şekil 5.3'ten (Gandomi vd. 2013) de görüleceği üzere üç çubuk yerleşiminde belirli kısıtlayıcılara tabi olarak ağırlığın minimize edilmesi

istenmektedir. Bu problemde 2 adet tasarım değişkeni (x_1, x_2), 3 adet kısıtlayıcı fonksiyon bulunmaktadır. Bu probleme ait matematiksel ifadeler aşağıda verilmiştir (Yang ve Gandomi, 2012).

Uygunluk fonksiyonu;

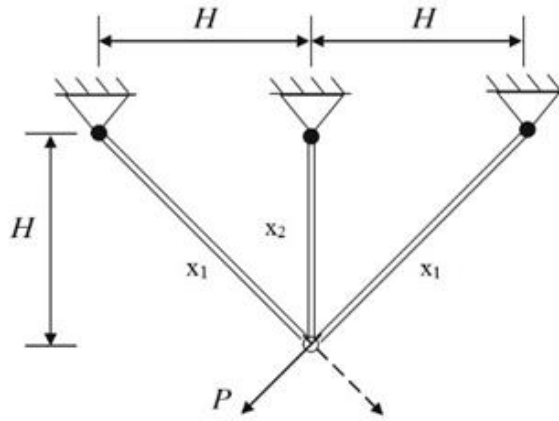
$$f(x) = (2\sqrt{2}x_1 + x_2) \times l \quad (5.13)$$

Kısıtlayıcı fonksiyonlar;

$$g_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0 \quad (5.14)$$

$$g_2 = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0 \quad (5.15)$$

$$g_3 = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0 \quad (5.16)$$



Şekil 5.3. Üç çubuk makas

Tasarım değişkenleri $0 < x_1, x_2 < 1$ aralığında ve diğer sabitler ise $l=100$ cm. $P = 2$ KN/cm² $\sigma = 2$ KN/cm² dir.

Çizelge 5.2’de geliştirmiş olduğumuz yaklaşımın bu probleme uygulanması sonucu elde edilen sonuçlar gösterilmektedir. Bu problemin çözümünde CBA’da 25 circır böceği popülasyonu kullanılmakta ve sonuçta 25000 gelişim sayısı elde edilmektedir. CBA yaklaşımı ile elde edilen en iyi sonuçlar $(x_1, x_2) = (0.788633, 0.408368)$ ve en iyi değer ise 263.895896’dır. Çizelge 5.3’te ise literatürdeki bu probleme uygulanmış diğer yaklaşımlar ile bizim yaklaşımımızın sonuçları

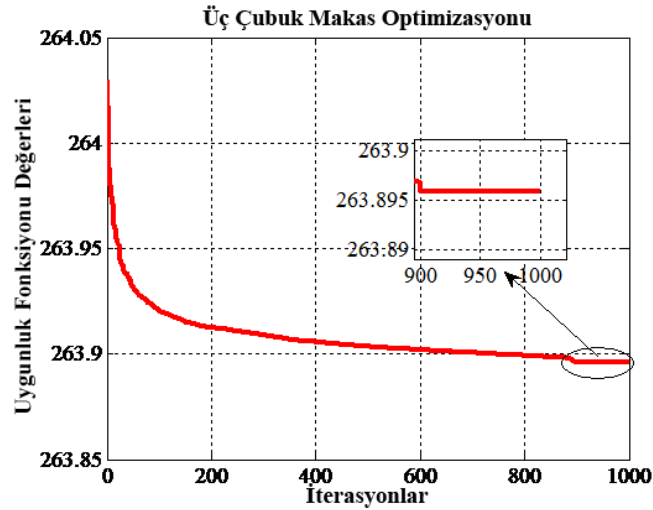
karşılaştırılmaktadır. Çizelge 5.3'e bakıldığında bizim yaklaşımımızın karşılaştırılan bazı yaklaşımlardan daha iyi sonuçlar verdiği bazılarında ise benzer sonuçlar verdiği görülmektedir. Ayrıca yaklaşımımızda en iyi sonucun elde edilmesinde kısıtlayıcı fonksiyonların tümünde şartları sağladığı görülmektedir. Probleme ait yakınsama grafiği Şekil 5.4'te verilmektedir.

Çizelge 5.2. Üç çubuk makas optimizasyon problem sonuçları

$x [x_1, x_2]$	F_{min}	Popülasyon Sayısı	Evrimsel Gelişim Sayısı	Ortalama Süre (s)
[0.788633, 0.408368]	263.895896	25	25000	5.8

Çizelge 5.3. Üç çubuk makas problemi için en iyi çözümler

Tasarım Değişkenleri	En İyi Çözümler					
	(Ray ve Saini, 2001)	(Gandomi vd. 2013)	(Yang ve Gandomi, 2012)	(Mirjalili, 2015)	(Sadollah vd. 2013)	CBA
x_1	0.79500	0.78867	0.78863	0.78866	0.7885650	0.788633
x_2	0.39500	0.40902	0.40838	0.40828	0.4082482	0.408368
$g_1(x)$	-0.00169	-0.00029	0.0000	---	-5.29E-11	-0.0000
$g_2(x)$	-0.26124	-0.26853	-0.26802	---	-1.4637475	-1.4640
$g_3(x)$	-0.74045	-0.73176	-0.73198	---	-0.5362524	-0.5360
$f(x)$	264.3000	263.9716	263.8962	263.8958	263.8958	263.8958



Şekil 5.4. Üç çubuk makas yakınsama grafiği

5.3. Himmelblau optimizasyon problemi:

Doğrusal olmayan (nonlinear) bu optimizasyon problemi ilk olarak (Himmelblau, 1972) tarafından tanıtılmıştır. Bu probleme ait 5 tasarım değişkeni (x_1, \dots, x_5), 6 kısıtlayıcı fonksiyon, 10 tane sınır değeri mevcuttur.

Minimize edilecek uygunluk fonksiyonu:

$$f(x) = 5,3578547x_3^2 + 0,8356891x_1x_5 + 37,293239x_1 - 40792,141 \quad (5.17)$$

Kısıtlayıcı fonksiyonlar;

$$0 \leq g_1 \leq 92, 90 \leq g_2 \leq 110, \text{ ve } 20 \leq g_3 \leq 25$$

$$g_1 = 85.334407 + 0.0056858x_1x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \quad (5.18)$$

$$g_2 = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 - 0.0021813x_3^2 \quad (5.19)$$

$$g_3 = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 - 0.0019085x_3x_4 \quad (5.20)$$

Tasarım değişkenleri sınır değerleri;

$$78 \leq x_1 \leq 102, 33 \leq x_2 \leq 45 \text{ ve } 27 \leq x_3, x_4, x_5 \leq 45$$

Bu problem daha sonradan g_1 kısıtlayıcı fonksiyonu Denklem 5.21'deki gibi olacak şekilde değiştirilmiştir. Bu nedenle problemin literatürde iki sürümü bulunmaktadır.

$$g_1 = 85.334407 + 0.0056858x_1x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5 \quad (5.21)$$

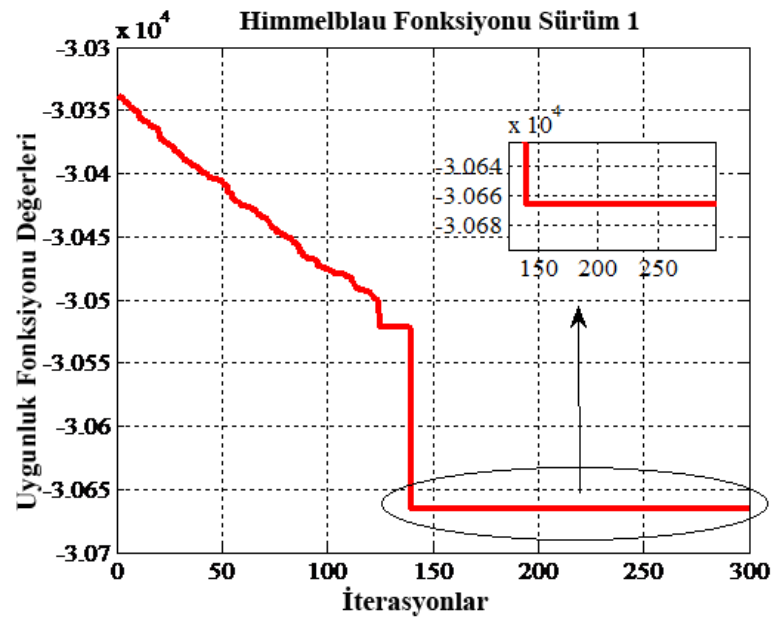
Bu problemin ilk sürümüne ait sonuçlar Çizelge 5.4 ve 5.5'de ve ikinci sürümüne ait sonuçlar ise Çizelge 5.6 ve 5.7'de verilmektedir. Sonuçlara ait yakınsama grafikleri Şekil 5.5 ve 5.6'da verilmektedir.

Çizelge 5.4. Himmelblau fonksiyonu ilk versiyonu için elde edilen değerler

x [x₁, x₂, x₃, x₄, x₅]	F_{min}	Popülasyon Sayısı	Evrimsel Gelişim Sayısı	Ortalama Süre (s)
[78, 33, 29.99, 45, 36.77]	-30665.481	20	20000	3.78

Çizelge 5.5. Himmelblau fonksiyonu ilk versiyonu için en iyi çözümler

Tasarım Değişkenleri	En İyi Çözümler				
	(Gandomi vd. 2013)	(Yang ve Gandomi, 2012)	(Sadollah vd. 2013)	(Garg, 2014)	CBA
x_1	78	78	78	78	78.000000
x_2	33	33	33	33	33.000000
x_3	29.99	29.99	29.99	29.99	29.995570
x_4	45	45	45	45	45.000000
x_5	36.77	36.77	36.77	36.77	36.775149
$g_1(x)$	91.99	91.99	92	91.99	91.99
$g_2(x)$	98.84	98.840	108.89	98.84	98.84
$g_3(x)$	20.0003	20.00	20.0	20.0	20.0
$f(x)$	-30665.232	-30665.4	-30665.5	-30665.5	-30665.481



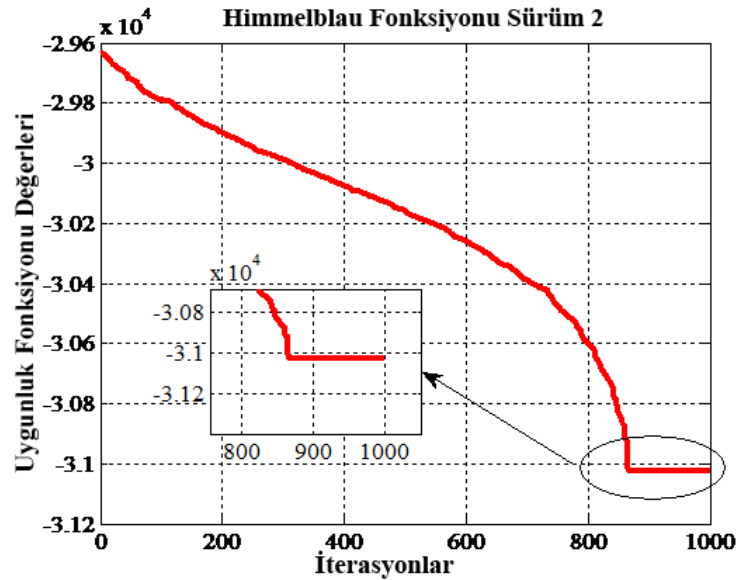
Şekil 5.5. Himmelblau fonksiyonu birinci sürüm yakınsama grafiği

Çizelge 5.6. Himmelblau fonksiyonu ikinci versiyonu için elde edilen değerler

$x [x_1, x_2, x_3, x_4, x_5]$	F_{\min}	Popülasyon Sayısı	Evrimsel Gelişim Sayısı	Ortalama Süre (s)
[78, 33, 27.07, 45, 44.96]	-31025.55	20	20000	3.78

Çizelge 5.7. Himmelblau fonksiyonu ikinci versiyonu için en iyi çözümler

Tasarım Değişkenleri	En İyi Çözümler		
	(Garg, 2014)	(Fesanghary vd. 2008)	CBA
x_1	78	78	78
x_2	33	33	33
x_3	27.07	27.08	27.07
x_4	45	45	45
x_5	44.96	44.925	44.96
$g_1(x)$	91.99	---	92
$g_2(x)$	100.40	---	100.4088
$g_3(x)$	20.000002	---	20.00
$f(x)$	-31025.57	-31024.3	-31025.55



Şekil 5.6. Himmelblau fonksiyonu birinci sürüm yakınsama grafiği

Elde edilen sonuçları incelediğimizde geliştirmiş olduğumuz yaklaşımın bu problemin literatürdeki sonuçları bildirilen diğer yaklaşımlar ile karşılaştırılmasında bazılarına benzer bazılarında ise daha iyi sonuçlar verdiği görülmektedir.

5.4. Yay dizayn optimizasyon problemi:

Yaylar çeşitli alanlarda sıklıkla kullanılan araçlardır. Yaygın kullanımlarına bağlı olarak optimum dizayn edilmeleri gerekmektedir. Bu optimizasyon probleminde üç dizayn değişkeni bulunmaktadır. Bunlar; x_1 telin çapı, x_2 esas yayın çapı, x_3 aktif yay sayısı veya yayın uzunluğu. Bu problemdeki amaç yayın ağırlığının minimize edilmesini sağlamaktır. Minimize edilme aşamasında belirli kısıtlayıcılar bulunmaktadır. Bunlar minimum bükülme, kayma gerilmesi, dalgalanma frekansı,

geometrik limitlerdir. Probleme ait genel bir yapı Şekil 5.7’de verilmektedir (Jaberipour ve Khorram, 2010).

Uygunluk fonksiyonu:

$$f(x) = (x_3 + 2)x_2x_1^2 \quad (5.22)$$

Kısıtlayıcı fonksiyonlar:

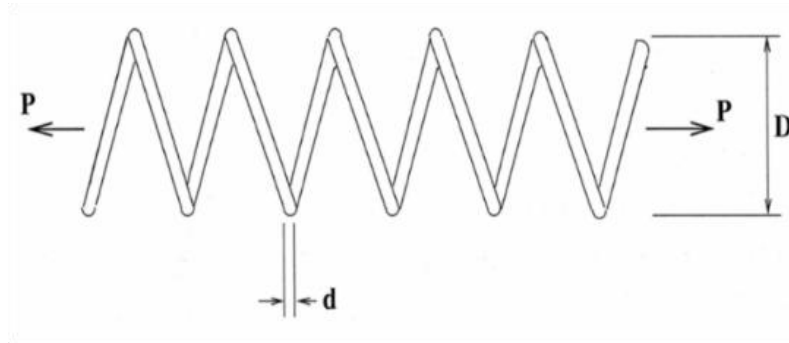
$$g_1 = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \quad (5.23)$$

$$g_2 = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3) - x_1^4} + \frac{1}{5108x_1^2} - 1 \leq 0 \quad (5.24)$$

$$g_3 = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \quad (5.25)$$

$$g_4 = \frac{x_2 + x_1}{1.5} - 1 \leq 0 \quad (5.26)$$

Tasarım değişkenleri sınır değerleri $0.05 \leq x_1 \leq 2.0$, $0.25 \leq x_2 \leq 1.3$ ve $2.0 \leq x_3 \leq 15.0$



Şekil 5.7. Yay dizayn

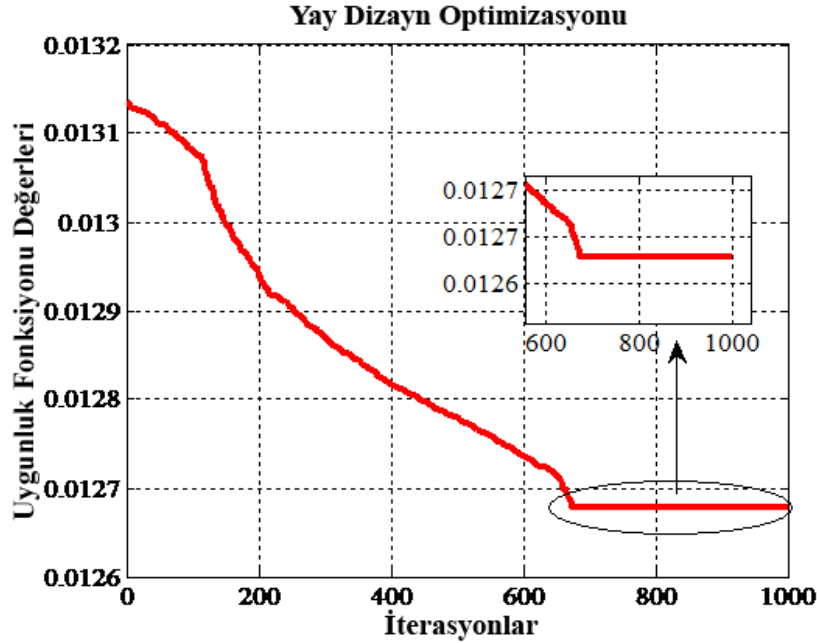
Çizelge 5.8. Yay dizayn problemi için CBA'dan elde edilen değerler

$x [x_1, x_2, x_3]$	F_{min}	Popülasyon Sayısı	Evrimsel Gelişim Sayısı	Ortalama Süre (s)
[0.052036, 0.365113, 10.815329]	0.012670	20	20000	3.4

Çizelge 5.9. Yay dizayn problemi için en iyi sonuçlar

Tasarım Değişkenleri	En İyi Çözümler				
	(Sadollah vd. 2013)	(Jaberipour, ve Khorrarn, 2010)	(Gong vd. 2014)	(Haipeng vd. 2015)	CBA
x_1	0.051656	0.0500	---	0.0516890609	0.052036
x_2	0.355940	0.317352	---	0.3567177390	0.365113
x_3	11.344665	14.037531	---	11.2889658089	10.815329
$g_1(x)$	0	-4.765397	---	---	-1.664901
$g_2(x)$	0	-1.812437	---	---	-1.924598
$g_3(x)$	-4.052248	-3.967268	---	---	-4.096105
$g_4(x)$	-0.728268	-0.755098	---	---	-1.082851
$f(x)$	0.012665	0.0127238	0.012665	0.012665	0.012670

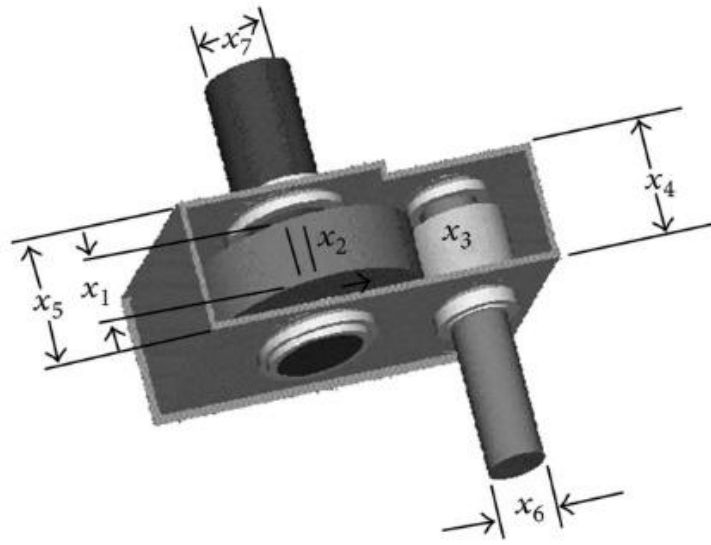
Çizelge 5.8’de bu probleme ait yaklaşımımızdan elde edilen sonuçlar gösterilmektedir. Bu optimizasyon problemi ile alakalı literatürde birçok makale bulmak mümkündür. Bu makalelerden elde edilen sonuçlar ve bizim yaklaşımımızın karşılaştırılması Çizelge 5.9’da verilmiştir. Çizelgeye bakıldığında CBA’nın diğer yaklaşımlar ile benzer sonuçlar verdiğini söylemek mümkündür. Sonuçların elde edilmesinde yaklaşımımız tüm kısıtlayıcı fonksiyonlar için 0’dan küçük olma şartını sağlamaktadır. Probleme ait yakınsama grafiği Şekil 5.8’de verilmektedir.



Şekil 5.8. Yay dizayn problemi yakınsama grafiği

5.5. Hız düşürücü optimizasyon problemi:

Bu problem Şekil 5.9’da gösterilen hız düşürücünün ağırlığının minimum olması esasına dayanır. Bu problemde $x_1 \dots x_7$ 7 tane dizayn değişkeni bulunmaktadır. Bunlar yüz genişliği (x_1), dişli modülü (x_2), küçük dişli içerisindeki diş sayısı (x_3), mil yatağı içerisindeki birinci şaftın uzunluğu (x_4), mil yatağı içerisindeki ikinci şaftın uzunluğu (x_5), birinci şaftın çapı (x_6) ve ikinci şaftın çapı (x_7). Bu değişkenlerden x_3 hariç diğerleri sürekli değer alırlar. x_3 ise ayrık değer alır. Bu problem aşağıdaki gibi formülize edilir (Lin vd. 2013).



Şekil 5.9. Hız düşürücü

Uygunluk fonksiyonu:

$$f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \quad (5.27)$$

Kısıtlayıcı fonksiyonlar:

$$g_1 = \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \quad (5.28)$$

$$g_2 = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0 \quad (5.29)$$

$$g_3 = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0 \quad (5.30)$$

$$g_4 = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0 \quad (5.31)$$

$$g_5 = \frac{\sqrt{(745x_4 / x_2x_3)^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \leq 0 \quad (5.32)$$

$$g_6 = \frac{\sqrt{(745x_5 / x_2x_3)^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \leq 0 \quad (5.33)$$

$$g_7 = \frac{x_2x_3}{40} - 1 \leq 0 \quad (5.34)$$

$$g_8 = \frac{5x_2}{x_1} - 1 \leq 0 \quad (5.35)$$

$$g_9 = \frac{x_1}{12x_2} - 1 \leq 0 \quad (5.36)$$

$$g_{10} = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0 \quad (5.37)$$

$$g_{11} = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \quad (5.38)$$

Tasarım deęişkenleri sınır deęerleri:

$$2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4, x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, 5 \leq x_7 \leq 5.5.$$

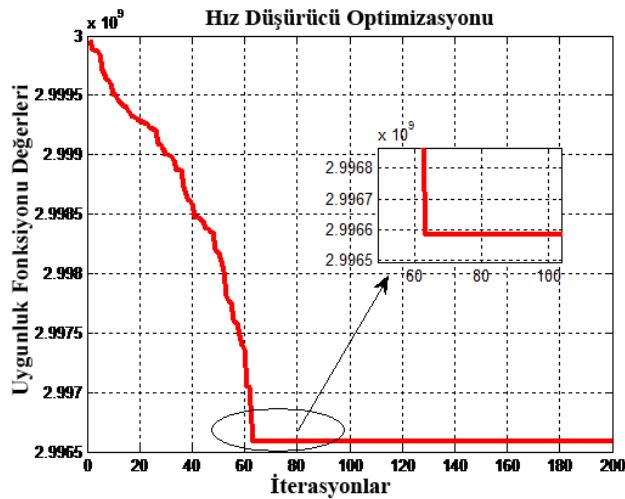
Çizelge 5.10. Hız düşürücü optimizasyon problem sonuçları

$x [x_1, x_2, x_3, x_4, x_5]$	F_{min}	Popülasyon Sayısı	Evrimsel Gelişim Sayısı	Ortalama Süre (s)
[3.5, 0.7, 17.00, 7.3, 7.7, 3.35, 5.28]	2996.5818090	10	15000	1.7

Çizelge 5.11. Hız düşürücü optimizasyon problem için en iyi çözümler

Tasarım Değişkenleri	En İyi Çözümler				
	(Gandomi vd. 2013)	(Lee vd. 2010)	(Akay Karaboğa, 2012)	(Kumar vd. 2012)	CBA
x_1	3.5	3.5	3.49	3.4999	3.500734
x_2	0.7	0.7	0.7	0.7	0.700000
x_3	17	17	17	17	17.00
x_4	7.6	7.3	7.3	7.3	7.346570
x_5	7.81	7.8	7.8	7.8	7.751806
x_6	3.35	3.36	3.35	3.35021	3.351603
x_7	5.28	5.29	5.28	5.28668	5.286819
g_1	-0.0743	-2.155	-0.073915	---	-0.0741
g_2	-0.1983	-98.135	-0.197999	---	-0.1982
g_3	-0.4349	-8568.2	-0.499172	---	-0.4904
g_4	-0.9008	-8403.12	-0.901555	---	-0.9033
g_5	-0.0011	-9.58263	0.0000	---	-0.0012
g_6	-0.0004	-1.59782	0.0000	---	-8.6351
g_7	-0.7025	-28.1	-0.7025	---	-0.7025
g_8	-0.0004	0	0.0000	---	-2.0967
g_9	-0.5832	-7	-0.583333	---	-0.5832
g_{10}	-0.0890	-0.36	-0.051326	---	-0.0571
g_{11}	-0.0130	-0.081	-0.010695	---	-0.0047
$f(x)$	3000.981	3000.83	2997.058412	2996.31	2996.5818090

Çizelge 5.10’da bu probleme ait yaklaşımımızdan elde edilen sonuçlar gösterilmektedir. Çizelge 5.11’de ise cırcır böceği yaklaşımımızın diğer yaklaşımlar ile karşılaştırma sonuçları gösterilmektedir. Elde edilen sonuçlardan bizim yaklaşımımızın diğer yaklaşımların çoğundan daha iyi sonuçlar verdiği görülmektedir. Probleme ait yakınsama grafiği Şekil 5.10’da verilmektedir.



Şekil 5.10. Hız düşürücü problemi yakınsama grafiği

Kullanılan problemlere ait performans değerlendirilmesinin yapıldığı toplu sonuçlar Çizelge 5.12’de verilmektedir.

Çizelge 5.12. Problemlerden elde edilen toplu sonuçlar

Problemler	Kaynaklar	Optimum Çözümler	Sonuçlar	Gelişim Sayısı	CPU (sn)
Problem 2	(Ray ve Saini, 2001)	(0.79500, 0.39500)	264.3000	---	---
	(Gandomi vd. 2013)	(0.78867,0.40902)	263.9716	15000	0.52
	(Yang ve Gandomi, 2012)	(0.78863,0.40838)	263.8962	15000	0.72
	(Mirjalili, 2015)	(0.78866,0.40828)	263.8958	14000	---
	(Sadollah vd. 2013)	(0.78856,0.40824)	263.8958	17610	---
	CBA	(0.788633, 0.408368)	263.8958	25000	5.8
Problem 3 Sürüm 1	(Gandomi vd. 2013)	(78, 33, 29.99, 45, 36.77)	-30665.2	5000	3.26
	(Yang ve Gandomi, 2012)	(78,33,29.99,45,36.77)	-30665.4	15000	2.76
	(Sadollah vd. 2013)	(78,33,29.99,45,36.77)	-30665.5	41750	----
	(Garg, 2014)	(78,33,29.99,45,36.77)	-30665.5	-----	----
Problem 3 Sürüm 2	CBA	(78, 33, 29.99, 45, 36.77)	-30665.481	20000	3.78
	(Garg, 2014)	(78,33,27.07,45,44.96)	-31025.57	---	---
	(Fesanghary vd. 2008)	(78,33,27.08,45,44.92)	-31024.31	28000	---
Problem 4	CBA	(78, 33, 27.07, 45, 44.96)	-31025.55	20000	3.78
	(Sadollah vd. 2013)	(0.0516, 0.3559,11.3446)	0.012665	7650	---
	(Jaberipour ve Khorram, 2010)	(0.0500, 0.3173, 14.0375)	0.012723	100000	0.77
	(Gong vd. 2014)	---	0.012665	19565	---
	(Haipeng vd. 2015)	(0.0516, 0.3567, 11.2889)	0.12665	---	---
Problem 5	CBA	(0.0520, 0.3651, 10.8153)	0.012670	20000	3.4
	(Gandomi vd. 2013)	(3.5, 0.7, 17, 7.6, 7.81, 3.35, 5.28)	3000.981	5000	0.65
	(Lee vd. 2010)	(3.5, 0.7, 17, 7.3, 7.8, 3.3, 5.29)	3000.83	105288	---
	(Akay ve Karaboğa, 2012)	(3.49, 0.7, 17, 7.3,7.8, 3.35, 5.28)	2997.058	30000	---
	(Kumar vd. 2012)	(3.49, 0.7, 17, 7.3, 7.8, 3.35, 5.28)	2996.31	15200	0.1
CBA	(3.5, 0.7, 17, 7.3, 7.7, 3.35, 5.28)	2996.5818090	15000	1.7	

6. CBA'NIN İMGE İŞLEME UYGULAMALARINDAKİ BAŞARI ANALİZİ

Meta-sezgisel algoritmaların imge işleme uygulamalarında kullanımı konusunda yapılan araştırmalarda genellikle imge eşikleme ve imge sıkıştırma alanlarında algoritmaların uygulandığını görülmektedir. Geliştirmiş olduğumuz yaklaşımın imge işleme alanındaki performansını göstermek amacıyla bu bölümde imge işleme uygulamalarındaki bu alanlarda yapılan optimizasyon işlemlerinde CBA çalıştırılarak sonuçlar gösterilecektir.

6.1. Entropi temelli imge eşikleme

İmge eşiklemede kullanılan eşik değerlerinin belirlenmesi problemi k boyutlu optimizasyon problemi olarak ele alınır. Bu belirleme işlemi entropi temelli problemin optimizasyonu sağlanarak gerçekleştirilebilir. Bilindiği üzere imge eşiklemesinde maksimum entropi kriteri için optimum eşik değerlerinin bulunmasında kullanılan birçok yöntem vardır. Kapur tarafından geliştirilen yöntemde en yaygın olarak kullanılan yöntemlerdendir (Kapur, 1985). Bu yöntemde verilen N piksel sayısına sahip A imgesi $[0,1,\dots,L-1]$ aralığında L gri seviye ile temsil edilmektedir. Optimizasyon problemlerinde esas olan uygunluk fonksiyon değerinin bulunmasında imgenin gri seviyelerdeki histogram değerleri kullanılır. Bu yöntemde kullanılan matematiksel Denklemler (6.1-6.3)'de verilmektedir (Canayaz ve Karcı, 2015b).

$$p_i = h(i) / N \quad (6.1)$$

Buradaki $h(i)$ i . gri seviyedeki piksel sayısını

N ise imgenin sahip olduğu tüm piksel sayısını ifade etmektedir. P_i ile i . seviyedeki normalleştirilmiş olasılık bulunmaktadır.

Entropi temelli yöntemde asıl amaç f fonksiyonunu maksimum yapan eşik değerlerinin bulunmasını sağlamaktır.

$$f(t_1, t_2, \dots, t_k) = H_1 + H_2 + \dots + H_k \quad (6.2)$$

$$H_1 = -\sum_{i=0}^{t_1-1} \frac{P_i}{w_0} \ln \frac{P_i}{w_0}, w_0 = \sum_{i=0}^{t_1-1} P_i \quad (6.3)$$

$$H_2 = -\sum_{i=t_1}^{t_2-1} \frac{P_i}{w_1} \ln \frac{P_i}{w_1}, w_1 = \sum_{i=t_1}^{t_2-1} P_i$$

$$H_k = -\sum_{i=t_k}^{L-1} \frac{p_i}{w_k} \ln \frac{p_i}{w_k}, w_k = \sum_{i=t_k}^{L-1} p_i$$

İmge eşiklemede kullanılan CBA'nın amacı entropi temelli yöntemdeki f uygunluk fonksiyonunu maksimum yapan k boyutlu vektördeki eşik değerlerinin bulunmasını sağlamaktır. Bu amaçla yapılacakları adım adım sıralayacak olursak;

Adım 1: Öncelikle $[1, L]$ aralığında rastgele eşik değerleri üretilerek bu eşik değerlerine göre histogram değerleri hesaplanır.

Adım 2: Bu histogram değerleri entropi temelli yöntem kullanılarak hedef fonksiyon için uygunluk değerleri hesaplanır.

Adım 3: Bu uygunluk değerleri arasından maksimum olanı en iyi uygunluk değeri olarak atanır (F_{max}). Bunu sağlayan çözümlerde en iyi çözümler olarak atanır (x_{best}).

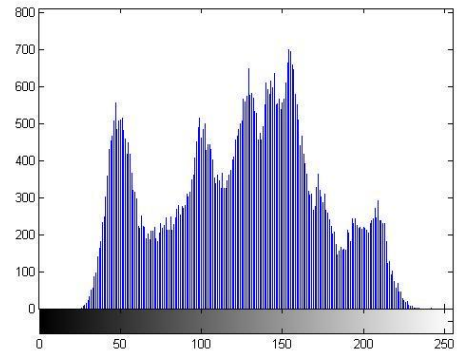
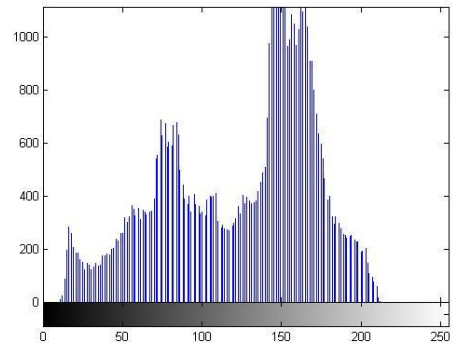
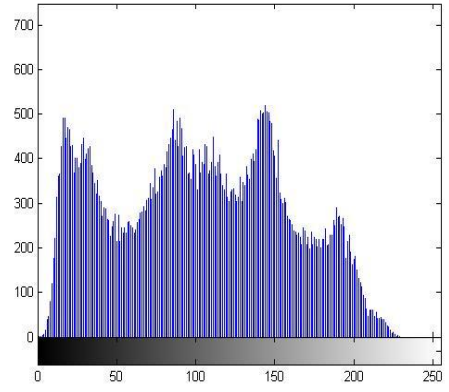
Adım 4: Algoritma belirlenen maksimum iterasyona kadar çalışacak ve bu iterasyona ulaşana kadar elde edilen uygunluk değerleri arasından maksimum sonucu veren değerler optimum eşik değeri olarak belirlenecektir.

Adım 5: Algoritma sonucunda belirlenen eşik değerlerine göre imge eşiklenerek bölütleme sağlanır.

Bu yöntem için kullanılan standart test imgeleri ve histogramları Şekil 6.1'de verilmektedir. Bu test imgeleri kullanılarak elde edilen uygunluk fonksiyonu değerleri Çizelge 6.1. verilmektedir. Eşiklenmiş imgeler Şekil 6.2.'de gösterilmektedir.

Çizelge 6.1. Entropi temelli eşikleme için CBA'dan elde edilen değerler

İmgeler	K	Sonuçlar		
		Eşik Değerleri	Hedef Fonksiyon	Zaman (saniye)
Lena	1	128	6.4448	0.287543
	2	156,181	13.5169	0.374309
	3	140,176,236	20.6402	0.463006
	4	171,203,215, 246	28.3028	0.559729
Bird	1	85	5.6616	0.924327
	2	182,210	13.1633	1.026984
	3	168,185,206	19.5565	1.101153
	4	116,206, 213, 226	25.7858	1.176400
Barbara	1	122	6.5452	1.012018
	2	131,205	13.8831	1.108714
	3	182,163,253	21.5667	1.193324
	4	168,201,216,223	28.8316	1.301844



Şekil 6.1. Test imgeleri ve histogram değerleri



Şekil 6.2. Eşiklenmiş test imgeleri

6.2. Entropi temelli imge sıkıştırma

İmgelerin her geçen gün artması ve bu imgelerin depolanması ihtiyacı imge sıkıştırma uygulamalarının önemini artırmaktadır. Bu bölümde bir imge sıkıştırma uygulamasında CBA'nın kullanımı gösterilecektir. Öncelikle imgenin özelliklerinin çıkarılması için imgeye ayrık dalgacık dönüşümü uygulanacaktır. Daha sonra çok seviyeli eşik değerlerinin kullanıldığı Shannon ve Tsallis entropiden faydalanılacaktır. Her iki entropi metodunda maksimum değerler aranmaktadır. Bu maksimumu arama işlemi CBA yardımıyla yapılarak işlem karmaşıklığı ve arama zamanı minimuma indirilmektedir. Yeni test imgeleri üzerinde uygulama yapılarak performansı gösterilecek ve sıkıştırma oranları verilecektir. Entropi temelli imge sıkıştırma açıklanmadan önce kullanılan yöntemdeki bazı bileşenler açıklanacaktır (Canayaz ve Karcı, 2015c);

6.2.1. Ayrık dalgacık dönüşümü

İşaretlerin analizinde belirli ölçütler kullanarak dalgacığı analiz etmek için birçok teknik kullanılmaktadır. Ayrık dalgacık dönüşümü de bu teknikler arasında yer alır. Ayrık dalgacık dönüşümünde sinyalin analizini kolaylaştırmak amacıyla sürekli sinyal parçalara ayrılarak analiz edilmekte ve bu sayede zaman karmaşıklığı azaltılmaktadır. Bu dönüşüm metodunda, tek boyutlu bir işaretin birbirine dik iki filtreden geçirilmesi sonucu işaret alçak ve yüksek frekans bileşenlerine ayrılmaktadır. (Ergen ve Baykara, 2011). Bu filtrelerden en yaygın kullanılanları Haar, Daubechies, Biorthogonal, gibi filtrelerdir.

$$W_{\psi} f(a,b) = \langle f, \psi(a,b) \rangle = \int_{-\infty}^{+\infty} f(x) \psi_{(a,b)}^*(x) dx \quad (6.4)$$
$$\psi_{(a,b)} = a^{-1/2} \psi\left(\frac{x-b}{a}\right)$$

Bir boyutlu $f(x)$ fonksiyonuna ait dalgacık dönüşümü Denklem 6.4'de verilmektedir (Şengür, 2008). Burada “ a ” ölçek faktörü, $\psi^*(x)$ ise dönüşüm fonksiyonudur. Kullanılan yüksek ve alçak filtre denklemleri Denklem 6.5'de verilmektedir.

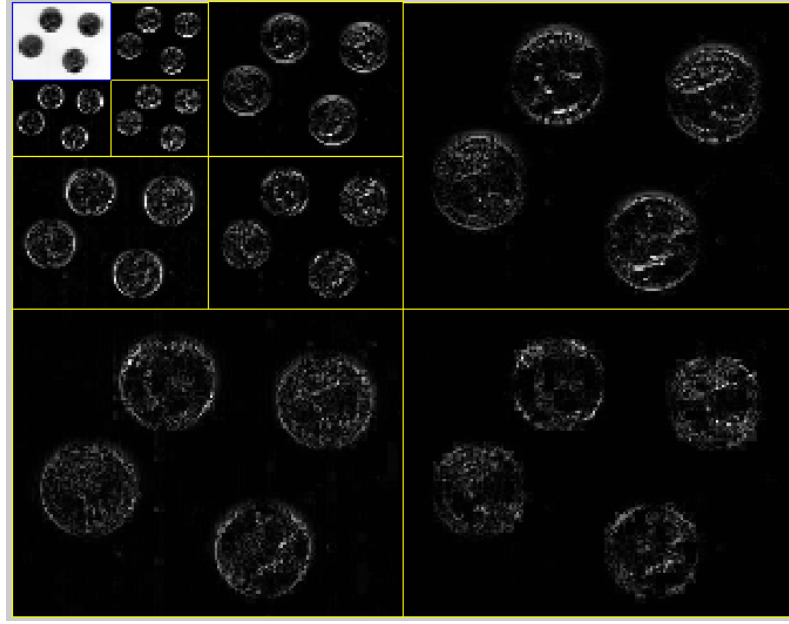
$$y_{yüksek}[k] = \sum_n x[n]g[2k-n]$$
$$y_{alçak}[k] = \sum_n x[n]h[2k-n] \quad (6.5)$$

Burada $y_{yüksek}(k)$ ve $y_{alçak}(k)$, sırasıyla yüksek geçiren (g) ve alçak geçiren (h) süzgeç çıkışlarıdır. Filtreler uygulandıktan sonra işaret kaba bir yaklaşıklık ve ayrıntı katsayılarına ayrılır ve bu sayede farklı frekans bantlarında işlem görür (Koçyiğit ve Korürek, 2005). Bu metod imge işleme uygulamalarında sıklıkla kullanılmaktadır. Bu dönüşüm imgeye uygulandığında sinyaldeki gibi imgeyi de alt bantlara ayırdığı görülmektedir. İmge için Biorthogonal 1.1 filtresinden elde edilen sonuçlar kullanılacaktır. İmgenin bu ayırma işlemi sonundaki ayrıldığı alt bantların temsili gösterimi Şekil 6.3’de verilmektedir. Alt bantlar LH, HL, HH ve LL’dir. İmgeye ait doku bilgilerinin çoğu LH ve HL alt bantlarında yer alır. Bu amaçla imgenin doku özelliklerinin elde edilmesinde bu bantlar kullanılarak ortalama enerjileri LHE ve HLE elde edilecektir.

LL	HL	LL1	HL1	HL
		LH1	HH1	
LH	HH	LH	HH	

Şekil 6.3. Alt bantlar temsili gösterimi

Şekil 6.4’de üç seviyeli Biortogonal 1.1 filtresi uygulanarak ayrık dalgacık dönüşümüne tabi tutulmuş imge gösterilmektedir. En üstte ve solda yer alan imge orijinal imgeyi temsil etmektedir.



Şekil 6.4. Üç seviyeli ayrık dalgacık dönüşümü uygulanmış imge

6.2.2. Shannon ve Tsallis entropi metotları

Bu bölümde CBA'nın uygunluk fonksiyonlarında yer alan Shannon ve Tsallis entropi modelleri açıklanmaya çalışılacaktır. Entropi istatistiksel manada belirsizliğin ölçüsü olarak kabul edilmekte ve birçok alanda kullanımı mevcut bulunmaktadır. İmge işleme alanı da entropinin sıklıkla kullanıldığı alanlar arasında yer almaktadır. Entropi hesaplanmasında önerilen Shannon (Beenamol vd. 2012) ve Tsallis (Agrawal vd. 2013) yöntemleri en bilinen yöntemlerdendir. Shannon entropi genel olarak Denklem 6.6'daki gibi tanımlanmaktadır.

$$S = -\sum_{i=1}^L p_i \log_2(p_i) \quad (6.6)$$

p_i : her i durumunun olma olasılığı, S : Olası tüm durumların toplam sayısı

Verilen sistemin birbirinden bağımsız X ve Y gibi iki istatistiksel alt parçadan ibaret olduğu düşünüldüğünde Shannon entropi modelini Denklem 6.7'deki gibi ifade etmek mümkün olmaktadır.

$$S(X + Y) = S(X) + S(Y) \quad (6.7)$$

Bu aşamada hem uzun zaman gerektiren işlemler hem de bazı kısıtlar bulunduğundan dolayı Tsallis bu kısıtların üstesinden gelmek amacıyla Denklem 6.8'deki gibi bir entropi hesaplama yöntemi önermiştir.

$$S_{\varphi} = \frac{1 - \sum_{i=1}^{\varphi} (p_i)^{\varphi}}{\varphi - 1} \quad (6.8)$$

Bu entropideki φ entropik indekstir. Tsallis (Albuquerque vd. 2004; Zhang ve Wu, 2011) mevcut sistemin iki alt parçaya bölüdüğü durumu Denklem 6.9'daki gibi ifade etmektedir.

$$S_{\varphi}(X + Y) = S_{\varphi}(X) + S_{\varphi}(Y) + (1 - \varphi)S_{\varphi}(X)S_{\varphi}(Y) \quad (6.9)$$

Bu denklemlerdeki φ 'nin aldığı değerlere göre 3 farklı entropi hesaplanabilmektedir.

Eğer $\varphi < 1$ alt geniş entropi $S_{\varphi}(X + Y) < S_{\varphi}(X) + S_{\varphi}(Y)$;

Eğer $\varphi = 1$ standart entropi $S_{\varphi}(X + Y) = S_{\varphi}(X) + S_{\varphi}(Y)$;

Eğer $\varphi > 1$ süper geniş entropi $S_{\varphi}(X + Y) > S_{\varphi}(X) + S_{\varphi}(Y)$;

İmgedeki nesnelere sahip olduğu pikseller arasında belirli bir korelasyonun olması ve imgenin analizinde rahatlıkla kullanılabilir olması nedeniyle bu modellerin imge işleme uygulamalarında sıklıkla kullanıldığını görmek mümkündür. Örneğin imge bölütleme işlemlerinde imge üzerindeki her nesnenin imgenin bir alt parçası gibi düşünülüp, bu alt parçalara ait entropinin hesaplanması daha verimli uygulamaların ortaya çıkmasını sağlamaktadır. İmgede bulunan arkaplan ve nesneyi iki sınıf gibi düşündüğümüzde, bu sınıfları X ve Y olarak isimlendirsek bunların Tsallis entropi modelindeki karşılığı 6.10 ve 6.11 denklemlerindeki gibi olur (Sayed, 2011).

$$p_X = \frac{p_1}{p_X}, \frac{p_2}{p_X}, \dots, \frac{p_t}{p_X}, \text{ ve } p_Y = \frac{p_{t+1}}{p_Y}, \frac{p_{t+2}}{p_Y}, \dots, \frac{p_L}{p_Y}, \quad (6.10)$$

$$p_X = \sum_{i=1}^t p_i \text{ ve } p_Y = \sum_{i=t+1}^L p_i$$

$$S_{\varphi}^X(t) = \frac{1 - \sum_{i=1}^t (p_i^X)^{\varphi}}{\varphi - 1}, S_{\varphi}^Y(t) = \frac{1 - \sum_{i=t+1}^L (p_i^Y)^{\varphi}}{\varphi - 1} \quad (6.11)$$

Kullanacağımız yöntemde $\varphi=0.5$ olarak alınacaktır.

6.2.3. İmge sıkıştırma yöntemi

İmge işleme uygulamalarında kullanılan entropi hesaplaması gri tonlamalı imgeler için gri seviyedeki histogram değerleri göz önünde bulundurularak yapılmaktadır. I imgesi M piksel sayısına sahip olup $[0,1,\dots,L-1]$ aralığında L gri seviye ile temsil edilmektedir. İmge sıkıştırma işlemini adım adım özetleyecek olursak;

Adım 1: İmge gri tonlamalı değilse gri tonlamalı yapılır.

Adım 2: İmgeye ayrık dalgacık dönüşümü uygulanarak altbantların enerji katsayı matrisi bulunur.

Adım 3: Bu katsayı matrisi için histogram değerleri hesaplanır ve entropi modelleri uygulanır.

Adım 4: Shannon veya Tsallis entropi yönteminde maksimum entropiyi sağlayan çok seviyeli eşik değerleri CBA yardımıyla belirlenir.

Adım 5: Bu eşik değerleri için histogram değerleri alınır ve bu değerler gri seviye değerleri ile ilişkilendirilir.

Adım 6: Bu gri seviye değerleri imgenin temsilinde kullanılır.

Adım 5 imge sıkıştırma için önemli olduğundan biraz daha ayrıntılı açıklayacak olursak;

CBA yardımıyla Shannon veya Tsallis entropi modellerinde kısa sürede bulunan eşik değerleri imgenin sıkıştırılması işleminde kullanılır. $n+1$ seviyede eşiklenen imgenin gösterimi için $\log_2(n+1)$ bit gerekmektedir. Gri ölçekli bir imgede 256 gri seviye bulunduğundan bu imgenin temsili ancak 8 bit ile sağlanmaktadır. Kullanılan yöntemde her imge için 15 eşik değeri bulunarak $n+1$ yani 16 seviyede imge temsil edilecektir. Bu kadar fazla eşik değerinin kullanılması sonucunda imgedeki piksel değerlerinin değişimi azalır. Bu yöntem (Paul ve Bandyopadhyay, 2014) tarafından da kullanılmıştır.

Bir örnek üzerinden açıklayacak olursak; [100, 130, 212] gibi üç eşik değeri bulduğumuzu düşünelim. Öncelikle 0-100,100-130,130-212 arası histogram değerleri hesaplanır. İki eşik değeri arasındaki frekans değerleri benzer değerlere sahip olduğu durumlarda bu değerler alınarak sıkıştırılmış imgenin temsili için gri seviye değerleri olarak ilişkilendirilir ($0 < E_1 < E_2 < \dots < E_{L-1} < L-1$). Bu sayede imge kayıplı bir şekilde sıkıştırılmış olur. Bu eşik değerleri Denklem 6.12 ve 6.13’de kullanılarak her bir $H(E)$ değeri hesaplanır. Daha sonra bu değerler Denklem 6.14’te yerine koyulur. Buradaki amaç $H(E)$ ’yi maksimum yapan eşik değerlerinin bulunmasıdır. CBA her iki modelde de bu maksimumu bulmak için kullanılmaktadır.

$$H_1(E) = -\sum_{i=0}^{E_1} \frac{P_i}{P_1} \ln \frac{P_i}{P_1}, H_2(E) = -\sum_{i=E_1+1}^{E_2} \frac{P_i}{P_2} \ln \frac{P_i}{P_2}, \dots, \quad (6.12)$$

$$H_{n+1}(E) = -\sum_{i=E_n+1}^{L-1} \frac{P_i}{P_n} \ln \frac{P_i}{P_n}$$

$$P_1(E) = \sum_{i=0}^{E_1} h_i, P_2(E) = \sum_{i=E_1+1}^{E_2} h_i, \dots, \quad (6.13)$$

$$P_{n+1}(E) = -\sum_{i=E_n+1}^{L-1} h_i$$

$$H(E) = H_1(E) + H_2(E) + \dots + H_{n+1}(E) \quad (6.14)$$

$$E^* = \max_E \{H(E)\}$$

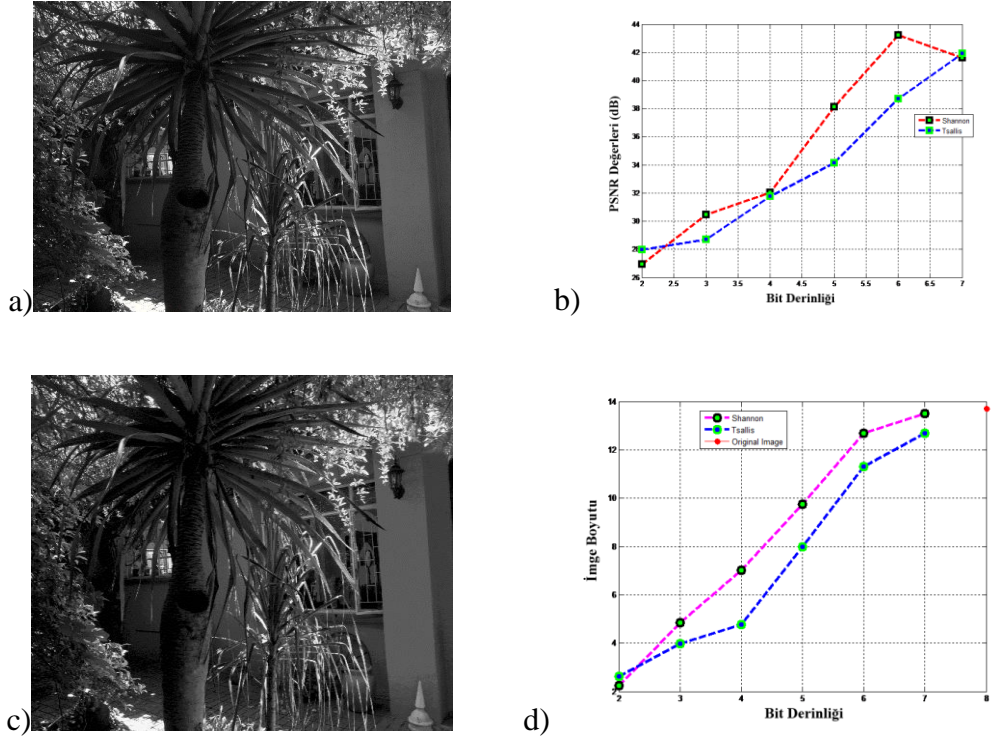
Sıkıştırma sonrası elde edilen imge ile orjinal imge için Doruk Sinyal Gürültü Oranı (Peak Signal to Noise Ratio (PSNR)) değerleri Denklem 6.15’deki denklemler kullanılarak hesaplanır (Jaiswal vd. 2014).

$$PSNR(dB) = 10 * \log \left(\frac{255^2}{MSE} \right) \quad (6.15)$$

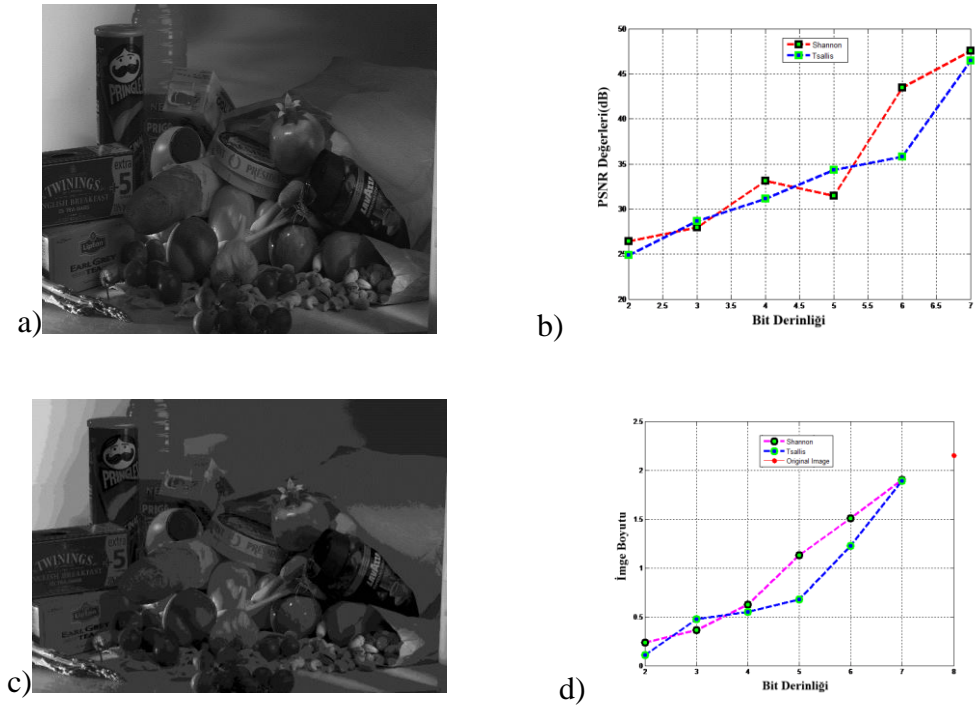
$$MSE = \sum_{i=1}^x \sum_{j=1}^y \frac{(|A_{ij} - B_{ij}|)^2}{x * y}$$

Burada A orjinal imgeyi, B ise sıkıştırılmış imgeyi göstermektedir. İmge sıkıştırma uygulamasında kullandığımız imgeler (Garg, 2015), sıkıştırma oranları, imge boyutları

ve sıkıştırılmış imgeler Şekil 6.5-6.8’de gösterilmektedir. CBA’dan elde edilen uygunluk fonksiyonu değerleri Çizelge 6.2’de verilmektedir.



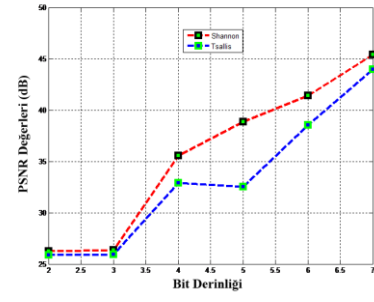
Şekil 6.5. (a) Orjinal imge (b) PSNR değerleri; (c) Shannon entropi kullanılarak sıkıştırılmış imge; (d) İmge boyutları



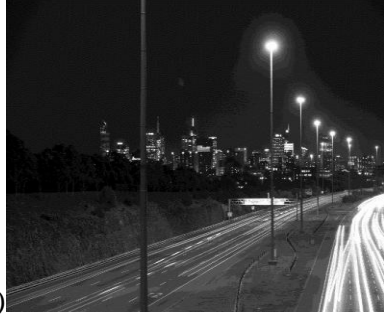
Şekil 6.6. (a) Orjinal imge (b) PSNR değerleri; (c) Tsallis entropi kullanılarak sıkıştırılmış imge; (d) İmge boyutları



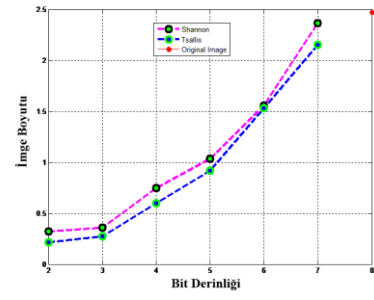
a)



b)

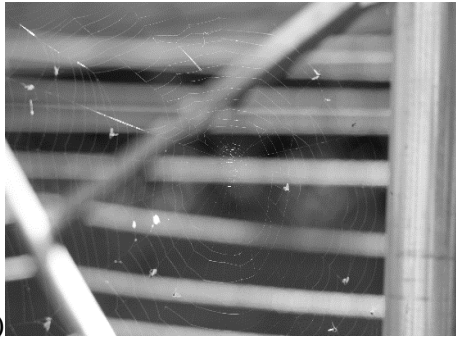


c)

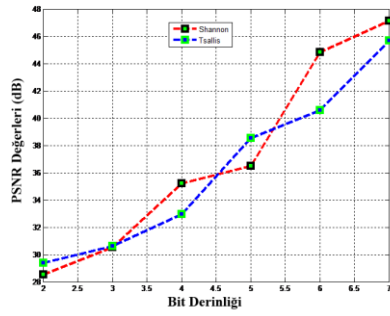


d)

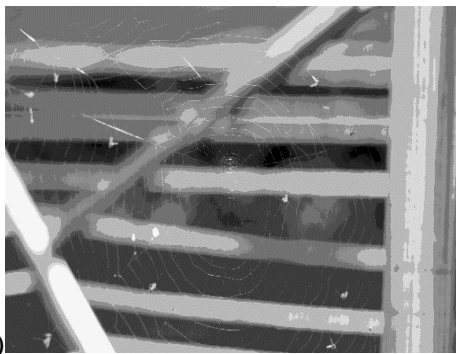
Şekil 6.7. (a) Orjinal imge (b) PSNR değerleri; (c) Shannon entropi kullanılarak sıkıştırılmış imge; (d) İmge boyutları



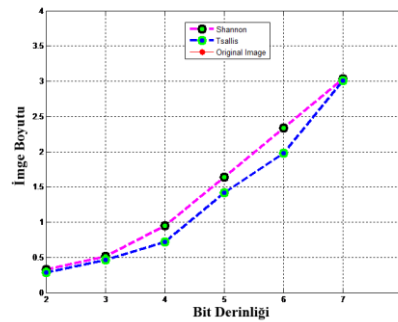
a)



b)



c)



d)

Şekil 6.8. (a) Orjinal imge (b) PSNR değerleri; (c) Tsallis entropi kullanılarak sıkıştırılmış imge; (d) İmge boyutları

Çizelge 6.2. Entropi temelli sıkıştırma için CBA'dan elde edilen değerler

İmgeler	Uygunluk Fonksiyon Değerleri (Shannon Entropi)	Uygunluk Fonksiyon Değerleri (Tsallis Entropi)
big_tree.png(13.713 Kb)	39.2339	30.9195
hdr.png(2.151 Kb)	42.2015	27.3813
nightshot_iso_100.png(2.472Kb)	36.6877	21.5728
spider_web(3.517 Kb)	40.0398	20.8671

7. TARTIŞMA VE SONUÇ

Bilindiği üzere meta-sezgisel algoritmalar problemler için çözüm arayışında oldukça verimli sonuçlar verebilen sezgisel yöntemlerdendir. Literatüre bakıldığında yüzlerce meta-sezgisel algoritma bulmak mümkündür. Bu algoritmaların temel amacı problemlerin çözümü için her zaman en iyiyi elde etmeye çalışmaktır. Tüm algoritmaların bu amacı her zaman sağladığını söylemek mümkün değildir. Yeni algoritmaların ortaya çıkma sebepleri arasında her problem için en iyi sonucu sağlayacak bir algoritmanın henüz keşfedilememiş olması sayılabilmektedir. Literatürde yer alan birçok algoritma kendisinin daha üstün olduğunu söylemektedir. Üstünlük kavramı kişiden kişiye değişebilir bir kavram olduğundan araştırmacıların optimizasyon alanında yeni algoritmalar ortaya çıkaracağı öngörülmektedir.

Bu tez çalışmasında genel olarak meta-sezgisel algoritmalar incelenerek belirli ölçütlere göre gruplandırılmış algoritmalar araştırmacılara tanıtılmaya çalışılmıştır. Tez kapsamında yaptığımız çalışmalarda yeni bir meta-sezgisel algoritma yaklaşımı sunulup sunulamayacağı üzerinde durulmuştur. Bu amaçla, tabiatta ismi cırcır böceği olarak anılan böcek türünün davranışları incelenmiştir. Bu inceleme sürecinde cırcır böcekleri ile tabiatın esinlenilen bazı sürü tabanlı meta-sezgisel algoritmalarda yer alan canlı türleri arasında ortak özellikler olduğu görülmüştür. Bu ortak özelliklerin yer aldığı matematiksel yöntemler önermiş olduğumuz yaklaşımın modellenmesinde kullanılmıştır. Bunların haricinde cırcır böcekleri doğada ses ile iletişime geçtiklerinden dolayı önermiş olduğumuz yaklaşımın doğaya yakınlığını artırmak amacıyla sesin doğadaki yayılımı ile alâkalı fiziki kanunlarda göz önünde bulundurulmuştur.

Önerilen yeni meta-sezgisel algoritma yaklaşımının 3 temel uygulama ile başarımları analizi yapılmıştır. Öncelikle bölüm 4'te tüm algoritmaların performans değerlendirilmesinde ortak olarak kullanılan standart test fonksiyonları üzerinde algoritma çalıştırılmıştır. CBA'nın test fonksiyonları üzerinde elde ettiği sonuçlara bakıldığında global optimumları rahatlıkla sağladığı görülmektedir. Algoritma ikinci olarak bölüm 5'te literatürdeki bilinen mühendislik optimizasyon problemlerine uygulanmış, bu problemler üzerindeki performansı gözlemlenmiştir. Çizelge 5.12'de bu problemlerden elde edilen sonuçlara bakıldığında algoritmanın karşılaştırıldığı diğer yöntemlerin bazılarında daha iyi sonuçlar verdiği bazılarında ise benzer

sonular verdiđi grlmektedir. Son olarak, algoritmanın performansı imge iřleme uygulamalarında denenmiř ve uygulama sonuları blm 6’da sunulmuřtur.

Sonu olarak, nerilen CBA meta-sezgisel yaklařımı kullanarak, gerekleřtirilen uygulamalardan elde edilen sonuların geneline bakıldıđında, nermiř olduđumuz yaklařımın performansı konusunda literatrdeki diđer meta-sezgisel algoritmalar ile yariřabilecek dzeyde olduđunu sylemek mmkndr. Algoritmanın pozitif diđer bir yn de problemlerin zmnde daha az evrim sayısı ve alıřma zamanı ile zm karmařıklıđını azaltmasıdır. nerilen algoritmanın yapısı itibariyle fazla parametre gerektirmemesi ve kolay uygulanabilir olması gibi zellikleri dřnldđnde, arařtırmacılar tarafından optimizasyon gerektiren bir ok alanda kullanılabileceđi ngrlmektedir.

8. KAYNAKLAR

- Agrawal, S., Panda, R., Bhuyan, S., Panigrahi, B.K. (2013). Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. *Swarm Evol. Comput.* **11**, 16–30, doi:10.1016/j.swevo.2013.02.001
- Akay, B. (2009). Nümerik optimizasyon problemlerinde yapay arı kolonisi (artificial bee colony) algoritmasının performans analizi. Doktora tezi, Erciyes Üniversitesi, Kayseri.
- Akay, B., Karaboga, D. (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *J. Intell. Manuf.* **23**, 1001–1014, doi: 10.1007/s10845-010-0393-4
- Akyol, S., Alataş, B. (2012). Güncel sürü zekâsı optimizasyon algoritmaları. *Nevşehir Üni. Fen Bil. Ens. Der.* **1**, 36-50.
- Alataş B. (2007). Kaotik haritalı parçacık sürü optimizasyonu algoritmaları geliştirme. Doktora tezi, Fırat Üniversitesi, Elazığ.
- Alataş, B. (2011). ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert Syst. Appl.* **38**, 13170–13180, doi:10.1016/j.eswa.2011.04.126
- Alataş, B. (2012). A novel chemistry based metaheuristic optimization method for mining of classification rules, *Expert Syst. Appl.* **39**, 11080–11088, doi:10.1016/j.eswa.2012.03.066
- Alba, E. (2005). *Parallels and Metaheuristic*. John Wiley, New Jersey.
- Albuquerque, M. P., Esquef, I.A., Gesualdi, M.A.R. (2004). Image thresholding using Tsallis entropy. *Pattern Recogn. Lett.* **25**, 1059–1065, doi:10.1016/j.patrec.2004.03.003
- Altunbey F., Alataş B. (2015). Overlapping community detection in social networks using parliamentary optimization algorithm. *IJCNA* **2**, 12-19.
- Atashpaz-Gargari, E., Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition (pp: 4661 – 4667). *Proc. Evolutionary Computation IEEE Congress*, September 25-28, Singapore doi:10.1109/CEC.2007.4425083
- Aydoğdu, M. (2015). Çevresel gürültünün değerlendirilmesi ve yönetimi. http://web.deu.edu.tr/cevre/belge/dersler/.../ders1_cevresel_gurultu.pps (on-line access on 01 May, 2014).
- Beenamol, M., Prabavathy, S., Mohanalin, J. (2012). Wavelet based seismic signal denoising using Shannon and Tsallis entropy. *Comput. Math. Appl.* **64**, 3580–3593, doi:10.1016/j.camwa.2012.09.009

- Belen, M.A., Alıcı, M., Çor, A., Güneş, F. (2014). Ateşböceği algoritması ile mikrodalga transistör performans karakterizasyonu (pp: 491-494). *Proc. Elektrik – Elektronik – Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu*, November 27 – 29, Bursa.
- Bingöl, H., Alataş, B. (2015). Metasezgisel optimizasyon tekniklerine spor tabanlı yeni bir yaklaşım: lig şampiyonası algoritması. *Fırat Üniv. Fen Bil. Derg.* **27**, 1-11.
- Birbil, S. I., Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. *J. Global Optim.* **25**, 263–282.
- Boschetti, M. A., Maniezzo, V., Roffilli, M., Röhler, A.B.(2009). Matheuristics: optimization, simulation and control. *Lect. Notes Comput. Sc.* 171-177, doi: 10.1007/978-3-642-04918-7_13
- Brown, D.W. (1999). Mate choice in tree crickets and their kin. *Annu. Rev. Entomol.* **44**, 371-396.
- Can, Ü. (2014). Yerçekimsel arama algoritmasıyla nicel birliktelik kurallarının keşfi, Yüksek Lisans tezi, Tunceli Üniversitesi, Tunceli.
- Canayaz, M., Karcı, A. (2015a). Investigation of cricket behaviours as evolutionary computation for system design optimization problems. *Measurement*, **68**, 225-235.
- Canayaz, M., Karcı, A. (2015b). İmge işleme uygulamalarında cırcır böceği algoritması. *Proc. Akademik Bilişim Konferansı*, February 4-6, Eskişehir.
- Canayaz, M., Karcı, A. (2015c). A novel approach for image compression based on multi-level image thresholding using discrete wavelet transform and cricket algorithm (pp: 224-227). *IEEE 23. Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, May 16-19, Malatya, Türkiye.
- Canayaz, M., Karcı, A. (2013). A new metaheuristic cricket-inspired algorithm (p: 176). *Proc. 2nd International Eurasian Conference on Mathematical Sciences and Applications*, August 26-29, Sarajevo, Bosna-Herzegovina.
- Çevik, K. K., Koçer, H.E. (2013). Parçacık sürü optimizasyonu ile yapay sinir ağları eğitimine dayalı bir esnek hesaplama uygulaması. *Süleyman Demirel Üni. Fen Bil. Ens. Der.* **17**, 39-45.
- Chu, S.-A., Tsai, P.-W., Pan, J.-S. (2006). Cat swarm optimization. *Lect. Notes Comput. Sc.* 854-858, doi: 10.1007/978-3-540-36668-3_94
- Crocker, M. J. (2008). Theory of sound predictions and measurement. *Handbook of Noise and Vibration Control*, John Wiley & Sons, Inc., Hoboken, NJ, USA.
- Cui, Z., Shi, Z., Zeng, J. (2010). Using social emotional optimization algorithm to direct orbits of chaotic systems. *Swarm, Evolutionary, and Memetic Computing* Springer. doi:10.1007/978-3-642-17563-3_47

- Dolbear, E. A. (1897). The cricket as a thermometer. *Amer. Nat.* **31**, 970-971.
- Dorigo, M., Stützle, T. (2004). *Ant colony optimization*. MIT Press, Cambridge.
- El-Sayed, M.A. (2011). A new algorithm based entropic threshold for edge detection in images. *IJCSI* **8**, 71-78.
- Erbatur, F., Hasançebi, O., Tütüncü, İ., Kılıç, H. (2000). Optimal design of planar and space structures with genetic algorithms. *Comput. Struct.* **75**, 209-224.
- Ergen, B., Baykara, M. (2011). Dalgacık ve dalgacık paket ayrıştırması ile imgelerden gürültü temizlemesi analizi. *NWSA* **6**, 518-526.
- Erol, O. K., Eksin, I. (2006). A new optimization method: Big bang-big crunch. *Adv. Eng. Softw.* **37**, 106–111.
- Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M. (2012). Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **110**, 151–166
- Fesanghary, M., Mahdavi, M., Minary-Jolandan, M., Alizadeh, Y. (2008). Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. *Comput. Methods in Appl. Mech. Eng.* **197**, 3080-3091, doi: 10.1016/j.cma.2008.02.006
- Formato, R. A. (2007). Central force optimization: A new metaheuristic with applications in applied electromagnetics. *PIER* **77**, 425–491.
- Gandomi, A. H., Yang, X. S., Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **29**, 17–35, doi: 10.1007/s00366-011-0241-y
- Garg, H. (2014). Solving structural engineering design optimization problems using an artificial bee colony algorithm. *JIMO* **10**, 777-794, doi:10.3934/jimo.2014.10.777
- Garg, S. (2015). The new image compression test set, http://www.imagecompression.info/test_images (on-line access on 10 Feb, 2015).
- Geem, Z.W., Kim, J. H., Loganathan. G.V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, **76**, 60–68.
- Glover, F. (1989). Tabu search-part 1. *Orsa J. Comput.* **1**, 190-206.
- Goldberg, D. E. (1989). *Genetic algorithm in search: optimization and machine learning*. Kluwer Academic Publishers, Boston, USA.
- Gong, W., Cai, Z., Liang, D. (2014). Engineering optimization by means of an improved constrained differential evolution. *Comput. Methods Appl. Mech. Engrg.* **268**, 884–904, doi:10.1016/j.cma.2013.10.019

- Green, R. C., Wang, L., Alam, M. (2012). Training neural networks using central force optimization and particle swarm optimization: Insights and comparisons. *Expert Syst. Appl.* **39**, 555–563.
- Gürsu, B. (2014). Elektromanyetizm algoritması ile radyal şebekelerde ters zamanlı yönsüz selektif aşırı akım röle koordinasyonu (pp.24-28). *Proc. Elektrik – Elektronik – Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu*, November 27 – 29 Bursa.
- Haipeng, K., Ni, L., Yuzhong, S. (2015). Adaptive double chain quantum genetic algorithm for constrained optimization problems. *CSAA* **28**, 214-228 doi:10.1016/j.cja.2014.12.010
- Hedrick, A. V., Kortet, R. (2006). Hiding behaviour in two cricket populations that differ in predation pressure. *Anim. Behav.* **72**, 1111-1118.
- Hedrick, A. V., Mulloney, B. (2004). A multichannel electronic monitor of acoustic behaviors, and software to parse individual channels. *J. Neurosci. Meth.* **133**, 201–210.
- Hedwig, B. (2006). Pulses, patterns and paths: neurobiology of acoustic behaviour in crickets. *J. Comp. Physiol. A.* **192**, 677–689.
- Hezer, S., Kara, Y. (2013). Eşzamanlı dağıtım ve toplamalı araç rotalama problemlerinin çözümü için bakteriyel besin arama optimizasyonu tabanlı bir algoritma. *Gazi Üniv. Müh. Mim. Fak. Der.* **28**, 373-382.
- Himmelblau, D. M. (1972). *Applied nonlinear programming*. New York, McGraw-Hill.
- Howard, D., Angus, J. (2009). *Acoustics and psychoacoustics*. Taylor and Francis.
- Hsu, Y-L., Wang, S-G., Yu, C-C. (2003). A sequential approximation method using neural networks for engineering design optimization problems. *Eng. Opt.* **35**, 489-511, doi:10.1080/03052150310001620713
- ISO (1993). *Acoustics attenuation of sound during propagation outdoors part 1: calculation of the absorption of sound by the atmosphere*. ISO 9613-1.
- Jaberipour, M. Khorram, E. (2010). Two improved harmony search algorithms for solving engineering optimization problems. *Commun. Nonlinear Sci.* **15**, 3316-3331, doi: 10.1016/j.cnsns.2010.01.009
- Jaiswal, A., Upadhyay, J., Somkuwar, A. (2014). Image denoising and quality measurements by using filtering and wavelet based techniques. *AEU –Int. J. Electron C.* **68**, 699-705, doi:10.1016/j.aeue.2014.02.003
- Kamışlıoğlu, B., Karaboğa, N. (2012). Armoni arama algoritması ile daha düşük dereceden sistem modelleme (pp. 576-580). *Proc. Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu*, November 29 - December 01, Bursa.

- Kapur, J.N., Sahoo, P.K., Wong, A.K.C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Comput. Vision Graph.* **29**, 273–285.
- Karaboga, D., Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. *J. Global Optim.* **39**,459-171, doi: 10.1007/s10898-007-9149-x
- Karcı, A. (2012). A new metaheuristic algorithm based chemical process: Atom Algorithm (p:85). *Proc. 1st International Eurasian Conference on Mathematical Sciences and Applications*, September 03-07, Pristina, Kosova.
- Kashan, A. H. (2009). League championship algorithm: a new algorithm for numerical function optimization (pp:43-48). *Proc. IEEE Soft Computing and Pattern Recognition*, December 04-07, Malacca.
- Kaveh, A., Talatahari, S. (2010). A novel heuristic optimization method: Charged system search. *Acta Mech.* **213**, 267–289.
- Kennedy, J., Eberhart, R. C. (1995). Particle swarm optimization (pp: 1942-1948). *Proc. IEEE International Conference on Neural Network*, Perth, WA, Australia.
- Kirkpatrick, S., Gelatt, J. D., Vecchi, M.P. (1983). Optimization by simulated annealing. *Science* **220**, 671–680,
- Koçyiğit, Y., Korürek, M. (2005). EMG işaretlerini dalgacık dönüşümü ve bulanık mantık sınıflayıcı kullanarak sınıflama. *ITU Dergisi* **4**.
- Kumar, P., Pant, M., Singh, V.P. (2012). Differential evolution with interpolation based mutation operators for engineering design optimization. *AMEA* **2**, 221-231.
- Larsen, J. L., LeMone, P. (2009). The sound of crickets. *Science Teacher* **76**, 37-41.
- Lee, K-M., Tsai, J-T., Liu, T-K., Chou, J-H. (2010). Improved genetic algorithm for mixed-discrete-continuous design optimization problems. *Eng. Optim.* **42**, 927-941, doi: 10.1080/03052150903505885
- Lin, M. H., Tsai, J. F., Hu, N. Z., Chang, S. C. (2013). Design optimization of a speed reducer using deterministic techniques. *Math. Probl. Eng.* 1-7, doi: 10.1155/2013/419043
- Mishra, S. (2006). Some new test functions for global optimization and performance of repulsive particle swarm method. *MPRA*, <http://mpra.ub.uni-muenchen.de/2718/> (on-line access on 18 August, 2013).
- Özdağ, R., Karcı, A. (2015). Sensor node deployment based on electromagnetism-like algorithm in mobile wireless sensor networks. *Int. J. Distrib. Sens. N.* doi: 10.1155/2015/507967

- Parsopoulos, K.E., Vrahatis, M. N. (2010). *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Reference, New York.
- Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* 52–67.
- Paul, S., Bandyopadhyay, B. (2014). A novel approach for image compression based on multi-level image thresholding using Shannon Entropy and Differential Evolution (pp: 56-61). *Proc. IEEE Students' Technology Symposium*, India, doi: 10.1109/TechSym.2014.6807914
- Pooranian, Z., Shojafar, Abawajy, M., J. H., Singhal, M. (2013). GLOA: a new job scheduling algorithm for grid computing. *IJIMAI* 2, 59-64, doi: 10.9781/ijimai.2013.218
- Ramezani, F., Lotfi, S. (2013). Social-Based Algorithm (SBA). *Appl. Soft Comput.* 13, 2837–2856, doi:10.1016/j.asoc.2012.05.018
- Rao, R.V., Patel, V. (2013). An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *Sci. Iran.* 20, 710–720, doi:10.1016/j.scient.2012.12.005
- Rao, S. S. (1996). *Engineering optimization: theory and practice*. Wiley, New York, USA.
- Rashedi, E., Nezamabadi-pour, H., Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Inform. Sciences* 179, 2232–2248.
- Ray, T., Saini, P. (2001). Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng. Optim.* 33, 735-748, doi: 10.1080/03052150108940941
- Sadollah, A., Bahreininejad, A., Eskandar, H., Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl. Soft Comput.* 13, 2592–2612, doi:10.1016/j.asoc.2012.11.026
- Said, G.A.E.A., Mahmoud, A. M., El-Horbaty, E. M. (2014). A comparative study of meta-heuristic algorithms for solving quadratic assignment problem. *IJACSA* 5, 1-6.
- Şengür, A. (2008). Wavelet transform and adaptive neuro-fuzzy inference system for color texture classification. *Expert Syst. Appl.* 34, 2120–2128, doi:10.1016/j.eswa.2007.02.032
- Shah-Hosseini, H. (2008). The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm. *IJBIC* 1, 71–79.

- Shi, Y., Eberhart, R. C. (1998). A modified particle swarm optimizer (pp: 69 - 73). *Proc IEEE International Conference on Evolutionary Computation*, May 4-9, Alaska.
- Silagadze, Z. K. (2004). Finding two-dimensional peaks. <http://arxiv.org/pdf/physics/0402085.pdf>. (on-line access on 05 Jan, 2015).
- Simon, D. (2008). Biogeography-based optimization. *Evolutionary Computation, IEEE Transactions* **12**, 702–713.
- Stephen, R. O., Hartley, J. C. (1995). Sound production in crickets. *J. Exp. Biol.* **198**, 2139-2152.
- Storn, R., Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**, 341–359.
- Sun, W., Yuan, Y. X. (2006). Optimization theory and methods. *Nonlinear Programming*. Springer verlag.
- Thanedar, P.B., Vanderplaats, G. N. (1995). Survey of discrete variable optimization for structural design. *J. Struct Eng. ASCE* **2**, 301–306.
- Türkay, M. (2015). Optimizasyon modelleri ve çözüm metodlari, internet. (on-line access on 20 May, 2015).
- Xie, L., Zeng, J., Cui, Z. (2009). General framework of artificial physics optimization algorithm (pp: 1321–1326). *Proc Nature & Biologically Inspired Computing World Congress on*, December 9-11, Coimbatore.
- Yan, G.W., Hao, Z.J. (2013). A novel optimization algorithm based on atmosphere clouds model. *Int. J. Comp. Intel. Appl.* **12**, doi: 10.1142/S1469026813500028
- Yang, X. S. (2010a). A new metaheuristic bat-inspired algorithm. *NICSO SCI.* **284**, 65-74.
- Yang, X. S. (2010b). Firefly algorithm, stochastic test functions and design optimization. *IJBIC* 78-84.
- Yang, X. S. (2013). Optimization and Metaheuristic Algorithms in Engineering. *Metaheurstics in Water, Geotechnical and Transport*, Elsevier.
- Yang, X. S., Deb. S. (2009). Cuckoo search via l'evy flights (pp: 210–214). *Proc Nature & Biologically Inspired Computing World Congress on IEEE*, December 9-11, Coimbatore.
- Yang, X. S., Gandomi, A. H. (2012). Bat algorithm: a novel approach for global engineering optimization. *Eng. Comput.* **29**, 464-483, doi: 10.1108/02644401211235834

Zhan, Z., Zhang, J., Shi, Y., Liu, H. (2012). A modified brain storm optimization (pp: 1–8). Proc. *Evolutionary Computation (CEC) IEEE Congress on*, June 10-15, Brisbane, QLD, doi: 10.1109/CEC.2012.6256594

Zhang, Y., Wu L. (2011). Optimal multi-level thresholding based on maximum Tsallis entropy via an artificial bee colony approach. *Entropy* 13841–859, doi: 10.3390/e13040841

EKLER

Ek 1. ISO Standardına Göre Havanın Sesi Emme Katsayısını Hesaplayan Matlab Kodu

```
function[AbsorbCoef]=CoefCalculate(F,T)
pres=1;
relh=50;
freq_hum=F;
temp=T+273;
C_humid=4.6151-6.8346*((273.15/temp)^1.261);
hum=relh*(10^C_humid)*pres;
tempr=temp/293.15;
frO=pres*(24+4.04e4*hum*(0.02+hum)/(0.391+hum));
frN=pres*(tempr^-0.5)*(9+280*hum*exp(-4.17*((tempr^-1/3)-1)));
alpha=8.686*freq_hum*freq_hum*(1.84e-11*(1/pres)*sqrt(tempr)...
+(tempr^-2.5)*(0.01275*(exp(-2239.1/temp)*1/(frO+freq_hum*freq_hum/frO))...
+0.1068*(exp(-3352/temp)*1/(frN+freq_hum*freq_hum/frN))));
db_humi=alpha;
db_humi =round(1000*db_humi)/1000;
AbsorbCoef=db_humi;
end
```

Ek 2. Cırcır Böceği Algoritması Matlab Kodu

```
function [best,fmin,N_iter]=cricket_algorithm(para)
% Varsayılan Parametreler
if nargin<1, para=[25 0.5]; end
n=para(1); % Population büyüklüğü
alpha=para(2);
betamin=0.2;
d=4;
pi=3.14;
Lb=-1*ones(1,d);%Alt Limit
Ub=1*ones(1,d); % Üst Limit Vektörleri

Qmin=0; % Minimum frekans

% İterasyon parametreleri
tol=10^(-6); % Durdurma Kriteri
```

```

N_iter=0;    % Toplam gelişim sayısını tutan sayacı

% Başlangıç vektörleri
Q=zeros(n,1);
v=zeros(n,d);

    for i=1:n,
        Sol(i,:)=Lb+(Ub-Lb).*rand(1,d);
        Fitness(i)=Fun(Sol(i,:));

    end

% Mevcut en iyiyi bulma
[fmin,I]=min(Fitness);
best=Sol(I,:);
tic;

% Algoritma başlangıç
while(fmin>tol)
    for i=1:n,
        N(i,:)=randi([0 120],1,d);
        T(i,:)=0.891797*N(i,:)+40.0252;
        if(T(i,.)>180)
            T(i,:)=180;
        end
        if(T(i,.)<55)
            T(i,:)=55;
        end
        C(i,)=(5/9)*(T(i,.)-32);
        V(i,:)=20.1*sqrt(273+C(i,:));
        V(i,:)=sqrt(V(i,))/1000;
        Z(i,)=(Sol(i,.)-best);
        if(Z(i,)==0)
            F(i,)=0;
        else
            F(i,)=V(i,)/Z(i,);
        end
        Q(i,)=Qmin+(F(i,)-Qmin)*rand;
        v(i,)=v(i,)+(Sol(i,.)-best)*Q(i)+V(i);
        S(i,)=Sol(i,.)+v(i,);
        SumF=sum(F(i,))/i+10000;
        SumT=sum(C(i,))/i;
        gamma=CoefCalculate(SumF,SumT);

        Solo=Sol;
        scale=(Ub-Lb);
        for j=1:n,
            if(Fitness(i)<Fitness(j))
                distance=sqrt(sum((Sol(i,.)-Solo(j,)).^2));
                PS=Fitness(i)*(4*pi*(distance^2));
            end
        end
    end
end

```

```

Lp=PS+10*log10(1/4*pi*(distance^2));
Aatm = (7.4 * ((F(i,:)^2)*distance)/ (50*(10^(-8))));
RLP=Lp-Aatm;
K=(RLP)*exp(-gamma*distance.^2);
beta=K+betamin;
tmpf=alpha.*(rand(1,d)-0.5).*scale;
M(i,:)=Sol(i,:).*(1-beta)+Solo(j,:).*beta+tmpf;

else
M(i,:)=best+0.01*randn(1,d);
end
end

% Yeni çözümler üretme
if(rand >gamma)

    u1(i,:)=S(i,:);
else
    u1(i,:)=M(i,:);
end
u1(i,:)=simplebounds(u1(i,:),Lb,Ub);

Fnew=Fun(u1(i,:));

if (Fnew<=Fitness(i))
    Sol(i,:)=u1(i,:);
    Fitness(i)=Fnew;

% Mevcut En iyiyi güncelleme

if Fnew<=fmin,
    best=u1(i,:);
    fmin=Fnew;
end
end
alpha=alpha_new(alpha);
end
N_iter=N_iter+n;

end

% Sonuçları Gösterme
disp(['Number of evaluations: ',num2str(N_iter)]);
disp(['Best =',num2str(best)]);
disp(['fmin: ',num2str(fmin)]);
toc;

function [Fonksiyon2]=Fun(u)

```

```
[Fonksiyon2]=Sphere(u);
```

```
function z= Sphere(u)
```

```
n = 2;
```

```
s = 0;
```

```
for j = 1:n
```

```
    s = s+u(j)^2;
```

```
end
```

```
z = s;
```

```
function alpha=alpha_new(alpha)
```

```
delta=0.97;
```

```
alpha=delta*alpha;
```

```
function s=simplebounds(s,Lb,Ub)
```

```
    ns_tmp=s;
```

```
    I=ns_tmp<Lb;
```

```
    ns_tmp(I)=Lb(I);
```

```
    J=ns_tmp>Ub;
```

```
    ns_tmp(J)=Ub(J);
```

```
    s=ns_tmp;
```

ÖZGEÇMİŞ

Ad Soyadı: Murat CANAYAZ

Doğum Yeri ve Tarihi: Van-17.12.1981

Adres: Yüzüncü Yıl Üniversitesi

E-Posta: muratcanayaz@gmail.com

Lisans: Azerbaycan Teknik Üniversitesi-Bilgisayar Mühendisliği-2005

Yüksek Lisans: Yüzüncü Yıl Üniversitesi-Elektrik ve Elektronik Mühendisliği-2010

DOKTORA TEZ KAPSAMINDA TÜRETİLEN AKADEMİK YAYINLAR

Yayın:

- Canayaz, M., Karcı, A. (2015a). Investigation of cricket behaviours as evolutionary computation for system design optimization problems. Elsevier, *Measurement*, **68**, 225-235.
- Canayaz M., Karcı A, Cricket behavior-based evolutionary computation technique in solving engineering optimization problems. Springer, *Applied Intelligence*, doi: 10.1007/s10489-015-0706-6 (Basım aşamasında)

Bildiri:

- Canayaz, M., Karcı, A. (2015b). İmge işleme uygulamalarında cırcır böceği algoritması. Proc. *Akademik Bilişim Konferansı*, February 4-6, Eskişehir.
- Canayaz, M., Karcı, A. (2015c). A novel approach for image compression based on multi-level image thresholding using discrete wavelet transform and cricket algorithm (pp: 224-227). *IEEE 23. Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, May 16-19, Malatya, Türkiye.
- Canayaz, M., Karcı, A. (2013). A new metaheuristic cricket-inspired algorithm (p: 176). Proc. *2nd International Eurasian Conference on Mathematical Sciences and Applications*, August 26-29, Sarajevo, Bosna-Herzegovina.