

**BENZETİLMİŐ TAVLAMALI HİBRİT BALINA  
OPTİMİZASYON ALGORİTMASININ KARESEL  
ATAMA PROBLEMİNE UYGULANMASI**

**2019  
YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĐİ**

**Adem BÜYÜKÖZER**

**BENZETİLMİŞ TAVLAMALI HİBRİT BALİNA OPTİMİZASYON  
ALGORİTMASININ KARESEL ATAMA PROBLEMİNE UYGULANMASI**

**Adem BÜYÜKÖZER**

**Karabük Üniversitesi**

**Fen Bilimleri Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Yüksek Lisans Tezi**

**Olarak Hazırlanmıştır**

**KARABÜK**

**Haziran 2019**

Adem BÜYÜKÖZER tarafından hazırlanan “BENZETİLMİŞ TAVLAMALI HİBRİT BALİNA OPTİMİZASYON ALGORİTMASININ KARESEL ATAMA PROBLEMİNE UYGULANMASI” başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Dr. Öğr. Üyesi Emrullah SONUÇ  
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

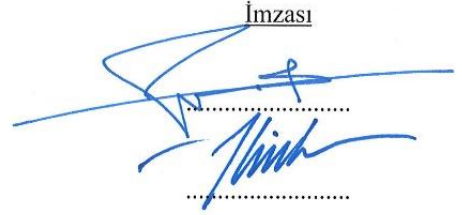


Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 19/06/2019

Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Doç. Dr. Baha ŞEN (AYBÜ)



Üye : Doç. Dr. İlker TÜRKER (KBÜ)



Üye : Dr. Öğr. Üyesi Emrullah SONUÇ (KBÜ)

19/06/2019

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Filiz ERSÖZ  
Fen Bilimleri Enstitüsü Müdürü V.





*Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”*

Adem BÜYÜKÖZER

## ÖZET

**Yüksek Lisans Tezi**

### **BENZETİLMİŞ TAVLAMALI HİBRİT BALİNA OPTİMİZASYON ALGORİTMASININ KARESEL ATAMA PROBLEMİNE UYGULANMASI**

**Adem BÜYÜKÖZER**

**Karabük Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı:  
Dr. Öğr. Üyesi Emrullah SONUÇ  
Haziran 2019, 33 sayfa**

Bilgisayar bilimlerinde kullanılan problem çözme yöntemlerinden klasik yöntemlerin çoğu zaman yetersiz ve performans açısından düşük kalması sezgisel (heuristics) algoritmaların doğmasına neden olmuştur. Sezgisel yöntemlerin kullanılmasında uygulanan yöntemin doğruluğunun ispat edilmesi gerekmez, amaç; karmaşık bir problemi daha basit bir hale getirmek veya tatmin edici bir sonuç elde etmektir.

Balina Optimizasyon Algoritması (BOA) kambur balinaların avlanırken kullandıkları kabarcık avlanma stratejisinden esinlenerek 2016 yılında yeni bir sezgisel algoritma olarak ortaya çıkmıştır. BOA yeni olmasına rağmen sezgisel algoritmalar arasında önemli bir yere sahiptir. Literatürde birçok optimizasyon problemine BOA uygulanmış ve başarılı sonuçların elde edildiği görülmüştür. Benzetilmiş Tavlama (BT) algoritması sezgisel algoritmalar arasında önemli bir yere sahiptir. Geçmişten günümüze birçok problem üzerinde uygulanmış ve uygulanmaya devam etmektedir.

Karesel Atama Problemi (KAP) polinomial zamanda çözülemeyen NP-Zor sınıfına ait bir kombinatoriyal optimizasyon problemidir. Bu çalışmada BOA, BT algoritması ile birlikte kullanılarak geliştirilen yöntemle KAP'a uygulanmıştır. Geliştirilen yöntemin başarısı standart sapma ve bağıl yüzde sapma kriterleri kullanılarak test edilmiştir.

**Anahtar Sözcükler :** Balina Optimizasyon Algoritması (BOA), Benzetilmiş Tavlama (BT), Karesel Atama Problemi (KAP), optimizasyon.

**Bilim Kodu** : 924.1.014



## **ABSTRACT**

**M. Sc. Thesis**

### **A HYBRID ALGORITHM BASED ON WHALE OPTIMIZATION ALGORITHM AND SIMULATED ANNEALING AND ITS APPLICATIONS FOR QUADRATIC ASSIGNMENT PROBLEM**

**Adem BÜYÜKÖZER**

**Karabük University**

**Graduate School of Natural and Applied Sciences**

**Department of Computer Engineering**

**Thesis Advisor:**

**Assist. Prof. Dr. Emrullah SONUÇ**

**June 2019, 33 pages**

In computer science, heuristic methods were arised because of the weakness and slowness of classical solution methods in artificial intelligence and optimization problems. It is not necessary to prove the accuracy of the method applied in the use of heuristic methods. Its aim that to make a complex problem simpler or to achieve a satisfactory result.

Whale Optimization Algorithm (WOA) is inspired by the bubble hunting strategy used by humpback whales, is presented in 2016. WOA has an important place among the algorithms even it is a recent developed algoritm. WOA has been applied to many optimization problems in the literature and has achieved successful results. Simulated Annealing (SA) algorithm is one of the heuristic methods. SA algorithm has been

applied to many optimization problems from past to present day and continues to be applied.

The Quadratic Assignment Problem (QAP) is a combinatorial optimization problem and it is NP-Hard that means can not be solved in polynomial time. In this study, a hybrid algorithm based on WOA and SA is applied to QAP. The developed method was tested using standard deviation and relative percentage deviation criteria to show its performance.

**Key Word** : Whale Optimization Algorithm (WOA), Simulated Annealing (SA), Quadratic Assignment Problem (QAP), optimization.

**Science Code** : 924.1.014



## TEŐEKKÜR

Bu tez alıŐmasının en baŐından beri planlama, araŐtırma, yürütme ve tamamlama süreçlerinde ilgi, alaka, güler güz ve desteęini esirgemeyen danıŐman hocam Sayın Dr. Öğr. Üyesi Emrullah SONUÇ'a sonsuz teşekkürlerimi sunarım.

Sevgili aileme bu kadar yoğun alıŐmalarım katlandıkları ve manevi desteklerini her zaman hissettirdikleri için teşekkür ederim.

## İÇİNDEKİLER

	<u>Sayfa</u>
KABUL.....	ii
ÖZET .....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER .....	ix
ŞEKİLLER DİZİNİ.....	xi
ÇİZELGELER DİZİNİ .....	xii
SİMGELER VE KISALTMALAR DİZİNİ.....	xiii
BÖLÜM 1 .....	1
GİRİŞ .....	1
1.1. ARKAPLAN .....	1
1.2. PROBLEMİN TANIMLANMASI.....	1
1.3. ÖNERİLEN ÇÖZÜM.....	2
1.4. ORGANİZASYON VE YOL HARİTASI.....	2
BÖLÜM 2 .....	4
LİTERATÜR ÇALIŞMALARINA GENEL BAKIŞ .....	4
BÖLÜM 3 .....	7
BALİNA OPTİMİZASYON ALGORİTMASI.....	7
3.1. GİRİŞ VE ARKAPLAN .....	7
3.2. BALİNA OPTİMİZASYON ALGORİTMASININ MATEMATİKSEL MODELİ .....	9
3.2.1. Avın Çevresini Sarma.....	9
3.2.2. Ava Doğru Hareket Etme .....	10
3.2.3. Av Arama .....	11
3.3. KALİTE TEST FONKSİYONLARI.....	13
3.3.1. Griewank Fonksiyonu.....	13

	<b><u>Sayfa</u></b>
3.3.2. Sphere Fonksiyonu .....	14
3.3.3. Schwefel Fonksiyonu .....	15
3.3.4. De Jong Fonksiyonu .....	15
3.3.5. Kalite Test Fonksiyonlarının Sonuçları.....	16
BÖLÜM 4 .....	17
BENZETİLMİŞ TAVLAMA ALGORİTMASI .....	17
4.1. GİRİŞ VE ARKAPLAN .....	17
4.2. METROPOLİS ALGORİTMASI VE BOLTZMANN DAĞILIMI .....	18
4.3. ALGORİTMANIN ÇALIŞMA PRENSİBİ .....	19
4.3.1. Başlangıç Sıcaklığı .....	21
4.3.2. Soğutma Katsayısı .....	21
4.3.3. Hedef Sıcaklık .....	22
4.3.4. İterasyon Sayısı.....	22
4.4. ALGORİTMANIN AVANTAJLARI VE DEZAVANTAJLARI .....	23
4.4.1. Algoritmanın Güçlü Yönleri.....	23
4.4.2. Algoritmanın Zayıf Yönleri.....	23
BÖLÜM 5 .....	24
KARESEL ATAMA PROBLEMİ.....	24
5.1. KARESEL ATAMA PROBLEMİ (KAP) .....	24
5.1.1. KAP'ın Çözüm Yöntemleri .....	25
BÖLÜM 6 .....	26
DENEYSEL ÇALIŞMALAR .....	26
KAYNAKLAR .....	31
ÖZGEÇMİŞ .....	33

## ŞEKİLLER DİZİNİ

### Sayfa

Şekil 3.1. Kambur balinaların avlama yöntemi.....	8
Şekil 3.2. Arama ajanlarının 2 ve 3 boyutlu pozisyon vektörleri ve olası konumları	10
Şekil 3.3. Spiral hareket .....	11
Şekil 3.4. BOA'nın global arama davranışı. ....	12
Şekil 3.5. Griewank fonksiyonu.....	14
Şekil 3.6. Sphere fonksiyonu .....	14
Şekil 3.7. Schwefel fonksiyonu.....	15
Şekil 3.8. De Jong fonksiyonu. ....	16
Şekil 4.1. Katı bir cisim için ısı işlem süreci. ....	18
Şekil 5.1. KAP için kullanılan sezgisel yöntemlerin dağılımı. ....	25
Şekil 6.1. BOA-BT'nin akış diyagramı.....	26
Şekil 6.2. BOA-BT'nin QAPLIB üzerindeki performansı.....	30

## ÇİZELGELER DİZİNİ

### Sayfa

Çizelge 3.1. BOA'nın kalite test fonksiyonları üzerindeki sonuçları .....	16
Çizelge 4.1. Tavlama süreci ve BT algoritması arasındaki ilişki.....	20
Çizelge 6.1. LOV'un çalışma prensibi.....	27
Çizelge 6.2. BOA'nın QAPLIB veri setlerine uygulama sonucu çıkan sonuçlar.....	28



## SİMGELER VE KISALTMALAR DİZİNİ

### KISALTMALAR

- BOA : Balina Optimizasyon Algoritması  
CPU : Central Processing Unit (Merkezi İşlemci Ünitesi)  
NP : Non-deterministic Polynomial-time (Deterministik-olmayan Polinomial-zaman)  
KAP : Karesel Atama Problemi (Quadratic Assignment Problem)  
BT : Benzetilmiş Tavlama  
RPD : Bağlı Yüzde Sapma (Relative Percentage Deviation)  
QAPLIB : Quadratic Assignment Problem Library (Karesel Atama Problemi)

## BÖLÜM 1

### GİRİŞ

Bu bölümde ilk olarak tezin arkaplanı ve motivasyon kaynağından bahsedilecek, devamında problemin tanımı yapılarak tezin amaçları detaylandırılacaktır. Önerilen çözüm sunulduktan sonra ana katkılar listelenerek tezin ana hatları ile bölüm sonlandırılacaktır.

#### 1.1. ARKAPLAN

Bilişim sektörünün hızla gelişmesi sezgisel (heuristic) algoritmaların doğmasına neden olmuştur. Sezgisel algoritmalar bir problemin çözümüne uygulandığı algoritmalar. Bu yöntem ile problemin doğruluğunun ispat edilmesi gerekmez, problemi daha basit bir hale getirmesi ya da algoritmanın olabildiğince optimum bir sonuç üretmesi yeterlidir.

#### 1.2. PROBLEMİN TANIMLANMASI

Bilgisayar bilimlerinde kullanılan problem çözme yöntemlerinden klasik yöntemler, çoğu zaman yetersiz ve yavaş kalmıştır. Bu da sezgisel (heuristics) yöntemlerin doğmasına neden olmuştur. Sezgisel yöntemlerin kullanılmasında uygulanan yöntemin doğruluğunun ispat edilmesi gerekmez, amaç; karmaşık bir problemi daha basit bir hale yetirmek veya tatmin edici bir sonuç elde etmektir.

Sezgisel algoritmalar, bir problemi çözmek için rastgele bir çözüm üreterek yöntemi başlatırlar. Daha sonra her bir adımda algoritmaya has tekniklerle komşu çözüm ararlar. Bulunan komşu çözümü yine kendine has tekniklerle kabul veya ret ederler. Bu amaç doğrultusunda algoritmaya ait sonlandırma kriterleri sağlanana kadar en iyi çözümü ararlar. Algoritma işlemini tamamlandığında ise bulunan en iyi çözümü

çözüm olarak sunar. Bu bağlamda bakıldığında, sezgisel algoritmalar genel olarak bir sonraki çözümü bulmak için bir önceki çözüme ihtiyaç duyarlar [1].

Balina Optimizasyon Algoritması (BOA), çok yeni olmasına rağmen sezgisel algoritmalar arasında önemli bir yer edinmiştir. Bu algoritma, diğer sezgisel algoritmalarından farklı olarak kambur balinaların avlanırken kullandıkları kabarcık avlanma stratejisinden esinlenerek 2016 yılında yeni bir sezgisel algoritma olarak geliştirilmiştir. Geliştirilen bu algoritmanın performansını ölçmek için optimizasyon problemleri üzerlerinde denenmesi gerekmektedir. BO algoritması ile ilgili detaylı ve geniş bilgi Bölüm 3’de sunulacaktır.

BOA’da balinanın avı, ulaşılabilecek optimum nokta olarak kabul edilir. Çözüm uzayını genişletmek için algoritmayı olabildiğince fazla çalıştırmak veya algoritmanın kullandığı iterasyon sayısını artırmak daha iyi çözümler elde etmemizi sağlayabilir. Fakat bu durum, algoritmanın çalışma süresini artıracığından optimum sonucu bulmak için zaman problemine yol açacaktır.

### **1.3. ÖNERİLEN ÇÖZÜM**

Çalışmada kullanılan BOA, karesel atama problemine uygulanmış ve bu probleme çözüm üretilmesi amaçlanmıştır. Bu aşamada sonuçların kalitesi ve çalışma sürelerinin sunulması hedeflenmiştir. Bu bağlamda tez çalışmasında elde edilen katkıların sunulması amaçlanmıştır.

### **1.4. ORGANİZASYON VE YOL HARİTASI**

Bu tez 6 bölümden oluşmakta ve her bir bölümde hangi konulara değinildiği aşağıda listelenmiştir.

Bölüm 1: Giriş bölümünde ilk olarak tezin arkaplanı ve motivasyonuna odaklanılmış, devamında problem tanımlanarak bu bağlamda probleme çözüm olarak sunulan tezin amaçlarından bahsedilmiştir.



Bölüm 2: Bu bölümde BOA ve Karesel Atama Problemi hakkında literatür çalışmalarına genel bakış çalışması yapılmıştır.

Bölüm 3: Bu bölümde BOA hakkında bilgiler verilmiştir. Algoritmanın çalışma prensibi anlatılmış, avantajları ve dezavantajları kapsamında algoritma hakkında değerlendirmeler sunulmuştur.

Bölüm 4: Bu bölümde BT algoritması hakkında bilgiler verilmiştir. Algoritmanın çalışma prensibi anlatılmış, diğer algoritmalarla performans karşılaştırılması yapılmış, avantajları ve dezavantajları kapsamında algoritma hakkında değerlendirmeler sunulmuştur.

Bölüm 5: Bu bölümde karesel atama problemi hakkında bilgi verilmiştir.

Bölüm 6: Tezin bu son bölümünde, hibrit BOA-BT algoritmasının probleme uygulanmasını şekilsel olarak anlatılmıştır. Tez bağlamında yapılan çalışmanın önemi, literatüre katkısı ve genel değerlendirmeler sunulmuştur.

## BÖLÜM 2

### LİTERATÜR ÇALIŞMALARINA GENEL BAKIŞ

Literatürde BOA ile ilgili yapılan çalışmalara bakıldığında algoritma çok yeni olduğu için henüz çok fazla bir çalışma yapılmamıştır. Yapılan çalışmalarda, CPU ile program çalıştırılarak performans değerlendirmesi yapılmış, kriter olarak süre ve sonuçların doğruluğu değerlendirilerek analizlerde bulunulmuştur. Bu çalışmalar, farklı optimizasyon problemleri üzerinde gerçekleştirilmiştir.

Bu bölümde literatürde konu ile ilgili yapılmış çalışmalar sunulmuştur.

Erkan T. ve Tuncay C. [1] kaotik haritalı balina optimizasyon algoritmaları üzerinde çalışmışlar, balina optimizasyon algoritmasının detaylı açıklanması kaotik haritalar yardımıyla daha iyi bir yakınsama sağlamayı amaçlamışlardır. BOA'da bir sonraki pozisyonu belirlemede rastgele sayılar kullanılır. Bu çalışmada ise beş farklı kaotik harita yardımıyla bu sayıların çeşitliliğin artırılması gerçekleştirilmiştir. Geliştirilen yöntemin performansın testi için matematiksel test fonksiyonları ile kısıtlı mühendislik problemi kullanılmıştır.

Mohamed A.-B. vd. [2] BOA'yı permütasyon akış tipi çizelgeme probleminin üstesinden gelmek için yerel bir arama stratejisiyle birleştiren yeni bir algoritma üretmeyi amaçlamışlardır. Yapılan çalışmada BOA'yı literatürdeki diğer algoritmalarla kıyaslamışlar ve BOA'nın diğer algoritmalara nazaran başarılı sonuçlar elde ettiğini göstermişlerdir.

Majdi M.M. ve Seyedali M. [3] yaptıkları çalışmada balina BOA'ya dayalı farklı özellik seçim tekniklerini tasarlamak için iki adet hibridizasyon modeli kullanmayı amaçlamışlardır. İlk modelde benzetilmiş tavlama algoritması, balina optimizasyon algoritmasına gömülürken, ikinci modelde ise balina optimizasyon algoritmasının her

iterasyondan sonra bulduđu en iyi çözüm modelini geliřtirmek için kullanmışlardır. Burada BT'yi kullanmalarının amacı, BOA tarafından bulunan en umut verici bölgeleri arayarak sömürüyü arttırmaktır. Önerilen yaklaşımların performansı, UCI deposundan alınan 18 standart kriter veri seti üzerinde değerlendirilmiş ve literatürde iyi bilinen üç sarmalayıcı özellik seçme yöntemi ile karşılařtırmışlardır. Yapılan çalışmanın sonucunda BOA'nın özellik seçme için uygun bir algoritma olduğunu kanıtlamışlardır.

Tansel D. [4] yapmış olduđu çalışmada Teaching-Learning-Based Optimizasyon (TLBO) yöntemini kullanarak iki aşamalı bir şekilde KAP problemini bir dizi TLBO tabanlı hibrit algoritma ile çözülmeyi amaçlamıştır. Öğrenme ve öğretim aşamaları ile her bir öğrencinin bilgisinin artırılması hedeflenmiş daha sonra Robust Tabu Search ile bu bilgilerin işlenmesi amaçlamıştır. Ardışık ve paralel TLBO tabanlı hibrit algoritmaların performanslarını, en iyi çözüm ve hesaplama açısından en gelişmiş meta-sezgisel özelliklerle karşılařtırmıştır. Deneysel olarak, önerilen algoritmaların performansının, birçok gerçek hayat probleminin modellenebileceđi KAP'ın çözümünü için test edilen en iyi algoritmalarla rekabetçi olduđu gösterilmiştir.

Emrullah S. vd. [5] yapmış oldukları çalışmada kombinatoriyel optimizasyon problemlerinden biri olan ve NP-Hard sınıfında yer alan KAP'ı çözmek için CUDA mimarisiyle Parallel Multistart Simulated Annealing (PMSA) yöntemini geliřtirilmişlerdir. İş parçacıkları arasında çoklu başlangıç tekniđi ve işbirliđi sağlanarak verimli bir yöntem geliřtirmişlerdir. İşbirliđi aynı ve farklı bloklardaki iş parçacıklarıyla gerçekleştirilmiştir. Bu makale hem hızlanmaya hemde çözüm kalitesine odaklanılmaktadır. KAP Kütüphanesi (QAPLIB) örnekleri üzerinde yapılan deneyler, PMSA'nın tek çekirdekli bir işlemciden 29 kata kadar daha hızlı çalıştığını ve birçok kıyaslama veri setinde kısa sürede en iyi bilinen çözümü elde ettiđini göstermiştir.

Mohamed A.-B vd. [6] yapmış oldukları çalışmada farklı ölçeklerde hem tek hem de çok boyutlu 0-1 sırt çantası problemlerini çözmek için geliřtirilmiş bir balina optimizasyon algoritması geliřtirmişlerdir ve IWOA adıyla sunmuşlardır. Deđerlendirme işlevine, uygulanabilir çözümlerin uygunluđunun, uygulanabilir

olanların uygunluğundan daha iyi performans gösterebilmesi için bir ceza işlevi eklemiştir. Sigmoid işlevi gerçek değerli çözümleri girdi olarak alabilir ve ikili çözümleri çıktı olarak üretir. Uygun olmayan çözümlerin ele alınması için iki aşamalı bir durum operatörü kullanılır. IWOA, iki strateji kullanarak çeşitlendirme ve yoğunlaşma arasında daha iyi bir denge sağlayabilir. Bu stratejiler Yerel Arama Stratejisi (LSS) ve Lévy uçuşu yürüyüşleridir. IWOA verimliliğini test etmek amacıyla, önerilen algoritma literatürdeki güncel algoritmalar ile karşılaştırılmıştır ve geliştirilen algoritmanın rekabetçi olduğu gözlemlenmiştir.



## BÖLÜM 3

### BALİNA OPTİMİZASYON ALGORİTMASI

Bu bölümde BOA tanımından ve tarihçesinden kısaca bahsedilerek çalışma prensibi anlatılacaktır.

#### 3.1. GİRİŞ VE ARKAPLAN

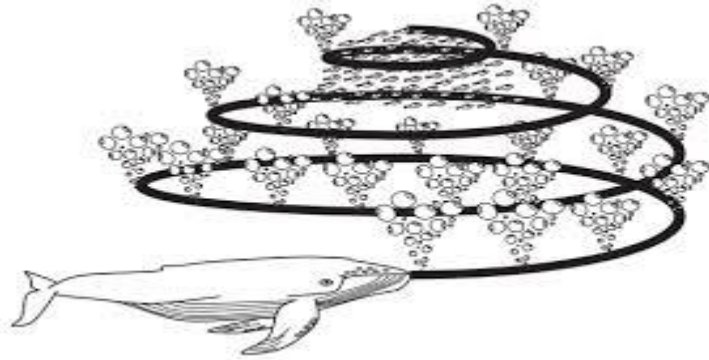
Balinalar, dünyadaki en büyük memeli canlı olarak bilirlir. Yetişkin bir balina 25-30 M (metre) uzunluğunda ve ağırlıkları 150 tonu bulan en büyük deniz canlılarıdır. Katil, Kuzey, Minke, Buzul, Mavi, Oluklu ve Kambur balina olmak üzere 7 türdür. Sürüler haline gezer ve kendilerine özgü çıkardıkları sesler sayesinde haberleşirler. Çok az uyurlar ve beyinlerinin sadece yarısı uyur.

Kambur balinaların beyin ağırlığının yaklaşık %20'sini beyincik oluşturur. Çok az koku alırlar, hatta hiç koku almadıklarına dair bazı kuşkular vardır. Gözleri küçüktür ve su basıncına karşı dayanıklıdır. Kambur balinaların en belirgin özelliği, yüzgeç büyüklükleri ve şekli, kuyruk rengi ve şekilleridir. Yüzgeçleri uzundur ve uç bölgelerde küçük yumrular vardır. Yüzgeçlerinin rengi genelde beyaz renklidir. Kelebeğe benzeyen kuyrukları, kolayca ayırt edilebilen beyaz ve gri renklerdeki desenler üzerlerinde bulundurlar. Kambur ismi, balinaların üçgeni ya da kancayı andıran yüzgeç şekillerinin basamak veya hörgüce benzemesinden ileri gelmektedir. [7].

Bazı kambur balinalar hava kabarcığı davranışı denilen avlanma metodunu kullanırlar. Suda nefes vererek hava kabarcıkları oluştururlar. Bu büyük ve biribine bağlı hava kabarcıkları kümeleri, avları birbirine yaklaştırıp bir arada toplar. Henüz kesin bir bilgi olmamasına rağmen beslenme olayının su altında olduğu varsayılmaktadır. Daha sonra

kambur balina, hava kabarcığının içinden su yüzüne yavaşça çıkar. Bu durumu bir kaç kez tekrarlar. Amacı, avın bulunması yada bulunan avın hareketsiz kalmasını ve şaşırtılmasını sağlayıp avı yakalamak olduğu tahmin edilmektedir. Ayrıca bu hareket, avın irkilmesiyle hareket etmesine, kambur balinanın avı kolayca bulmasına neden olmaktadır. Sütunlar, kambur balinanın su altında bir çember çizerek yüzerken soluk vermesi ile oluşmaktadır. Tek sütun, sütunlar, yarım daireler ya da tam daireler oluşturulabilir. Bu daireler, avları bir araya getirmede ya da sürü halinde toplamaya yarar.

Kambur balinaların beslenme davranışları, su yüzeyinde bulunurken ve daire çizerek yüzerlerken karmaşık bir yapıya benzer. Bu yapıda, balinalar kuyruklarını kullanırlar ve suya vurarak su yüzeyinde köpükler oluştururlar ve bu köpükler avlarının etrafını sararlar. Oluşturulan bu daire sayesinde suyun altından yukarıya daire merkezine doğru ağızlarını açarak yüzerler ve avlarını yemeyi amaçlarlar. [7] (Şekil 3.1).



Şekil 3.1. Kambur balinaların avlama yöntemi.

Kambur balinalar bazen farklı bir avlanma tekniği de sergilerler. Suyun altından yüzeye doğru dik ya da eğik bir şekilde yüzerler ve bu sayede Plankton ya da balık sürülerini avlarlar. Bu beslenme çeşidi avın bol olduğu durumlarda olur.

Bu çalışmada üzerinde çalışılan tür, kambur balinadır. Kambur balinaların dişileri ortalama 15-16 metre, erkekleri ise ortalama 13-14 metredir ve küçük balık sürülerini avlarlar.

## 3.2. BALİNA OPTİMİZASYON ALGORİTMASININ MATEMATİKSEL MODELİ

Bu bölümde kambur balinaların avlanma yöntemi; avın çevresini sarma, ava doğru ilerleme ve avı arama olmak üzere üç kısımda ele alınmıştır.

### 3.2.1. Avın Çevresini Sarma

Kambur balinalar yetenekleri sayesinde avlarının konumunu belirleyebilir ve bu sayede avlarının etrafında kabarcıklar çıkararak onları kuşatırlar. BOA'da çözülmek istenen problemin optimum değeri av olarak nitelenir. Optimizasyon problemlerinin optimum değeri genel olarak bilinmediğinden, bu algoritmalar optimum değeri bulabildikleri en iyi değer veya ona yakın bir değer olarak kabul edebilirler. Bu en iyi değere sahip arama ajanı bulunur ve her bir adımda en iyi değere sahip arama ajanı güncellenir. Avın etrafını sarmanın matematiksel modeli Denklem 3.1 ve Denklem 3.2'de gösterilmiştir [8].

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \quad (3.1)$$

$$\vec{X}(t+1) = |\vec{X}(t) - \vec{A} \cdot \vec{D}| \quad (3.2)$$

Denklem 3.1 ve 3.2'de  $t$  bulunan iterasyonunu,  $\vec{A}$  ve  $\vec{C}$  yakınsama vektörlerini,  $\vec{X}^*$  bulunan en iyi çözüm vektörü ve '.' ise matris çarpımını ifade etmektedir [8].

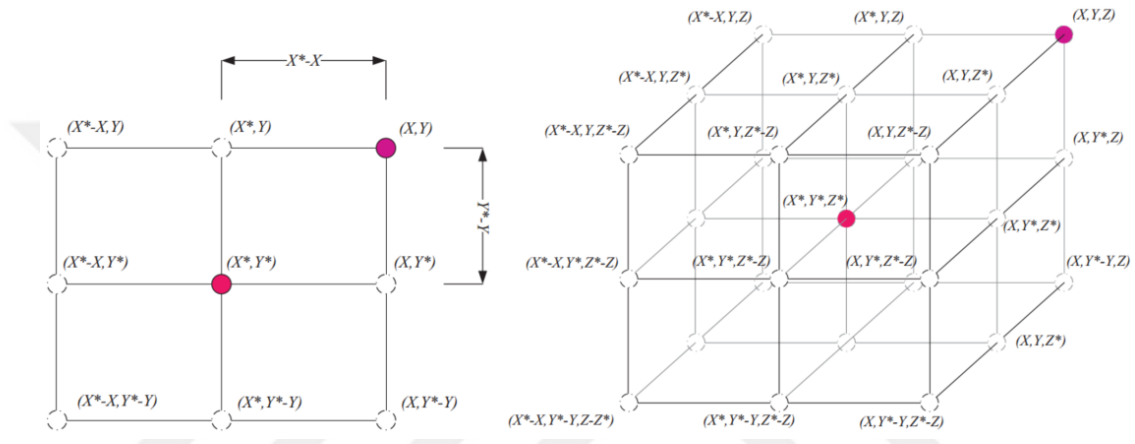
$\vec{A}$  ve  $\vec{C}$  yakınsama vektörleri, Denklem 3.3 ve Denklem 3.4 'teki gibi hesaplanmaktadır.

$$\vec{A} = 2a \cdot r - a \quad (3.3)$$

$$\vec{C} = 2 \cdot r \quad (3.4)$$

Denklem 3.3 ve 3.4'te,  $r$  rastgele vektörünü ve  $\vec{a}$  iterasyonlar boyunca 2'den sıfıra doğru lineer olarak azalan vektörü göstermektedir [8].

Şekil 3.2 (a), 2 boyutlu problem için Denklem 3.2 'nin mantığını göstermektedir. Rastgele bir arama ajanı  $(X, Y)$  pozisyonunu en iyi arama ajanına  $(X^*, Y^*)$  göre belirleyebilir. Bilinen en iyi ajanın etrafındaki noktalara ise  $\vec{A}$  ve  $\vec{C}$  sabitleri değiştirilerek erişilebilir. Şekil 3.2 (b)' de ise aynı durumun 3 boyutlu arama uzayı gösterilmiştir.



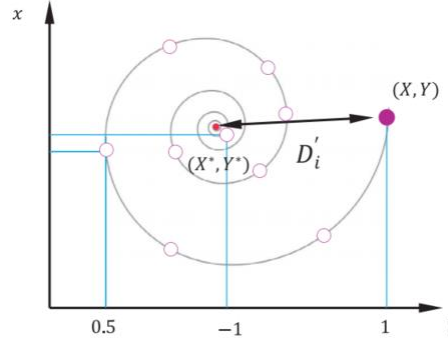
Şekil 3.2. Arama ajanlarının 2 ve 3 boyutlu pozisyon vektörleri ve olası konumları.

### 3.2.2. Ava Doğru Hareket Etme

Bu işlem, av etrafındaki dairenin daralması ve spiral hareketin durumu olmak üzere iki farklı şekilde modellenmiştir. Av etrafındaki daireyi küçültme, Denklem 3.3'teki  $\vec{a}$  vektörünün değerini düşürmekle olabilmektedir. Şekil 3.3'te en iyi arama ajanın konumu ve yaptığı spiral hareketi gösterilmiştir. Buradaki hareketten yola çıkılarak en iyi arama ajanı ile diğer arama ajanı arasındaki bağıntı Denklem 3.5 ile formülize edilmiştir.

Denklem 3.5'te  $\vec{D}^i = \vec{X}^* * (t) + \vec{X}(t)$ 'dir. Bu ifade arama ajanı ile bilinen en iyi nokta arasındaki mesafeyi vermektedir.  $b$  logaritmik spiral sabiti,  $l$  ise  $[-1,1]$  arasında rastgele sayıdır. Kambur balinalar ava doğru hareket ederken paralelinde spiral hareket etmeye devam eder. Algoritmada spiral mı doğrusal hareketmi hangi hareketin yapılacağı Denklem 3.6'daki gibi %50 olasılıkla belirlenir.





Şekil 3.3. Spiral hareket.

$$\vec{X}(t+1) = \vec{D}^l \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (3.5)$$

$$\vec{X}(t+1) = \begin{cases} \vec{X}^x(t) - \vec{A} \cdot \vec{D} & , p < 0,5 \\ \vec{D}^l \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), & p \geq 0,5 \end{cases} \quad (3.6)$$

Denklem 3.6'da  $p$ ,  $[0,1]$  arasında rastgele değerdir.

### 3.2.3. Av Arama

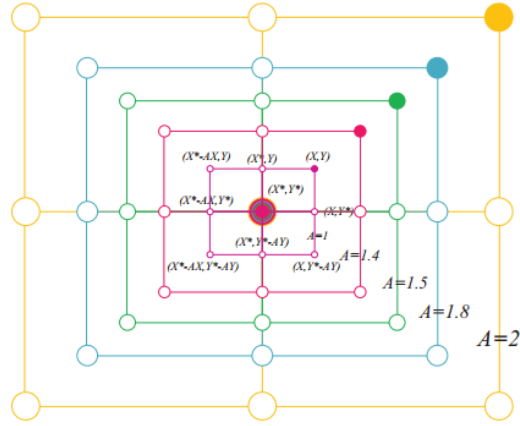
Bu adımdan arama ajanlarının yeni konumları belirlenir. Bu işlem için rastgele bir arama ajanı seçilir ve bu ajan üzerinden bir hesaplama gerçekleştirilir. Bu hesaplamalar Denklem 3.7 ve Denklem 3.8'de gösterilmiştir [8].

$$\vec{D}^l = \vec{C} \cdot \overrightarrow{X_{rank}} - \vec{X} \quad (3.7)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rank}} - \vec{A} \vec{D} \quad (3.8)$$

Denklem 3.7 ve 3.8'de  $\overrightarrow{X_{rank}}$  rastgele seçilen bir arama ajanını göstermektedir.

Global veya yerel aramalardan hangisinin yapılacağına ise  $\vec{A}$ 'nın değerine göre karar verilir. Şekil 3.4'te gösterildiği gibi,  $\vec{A} > 1$  veya  $\vec{A} < -1$  olduğunda en iyi noktadan daha uzak nokta seçilebileceğinden, bu durumlar global arama olarak kabul edilir ve Denklem 3.7 ve Denklem 3.8 uygulanır [8].



Şekil 3.4. BOA'nın global arama davranışı.

BOA'nın matematiksel modeli, kambur balinanın avın etrafını sarması durumunu vektörlerle ifade etmiştir. BOA'nın sözde kodu gösterilmiştir.

```

Balina popülasyonunu başlat  $X_i$  ( $i = 1, 2, \dots, n$ )
Her arama ajanının uygunluk durumunu hesapla
 $X^*$  = Bilinen en iyi arama ajanı değeri
while ( $t <$  maksimum iterasyon sayısı)
for (her bir arama ajanı için)
    Güncelle  $a$ ,  $A$ ,  $C$ ,  $l$ , ve  $p$ 
    if ( $p < 0.5$ )
        if ( $|A| < 1$ )
            Denklem 3.1 ile arama ajanını güncelle
        else if ( $|A| < 1$ )
            Rastgele bir arama ajanı seç ( $\overrightarrow{X_{rank}}$ )
            Denklem 3.8 ile arama ajanını güncelle
        end if
    else if ( $p \geq 0.5$ )
        Denklem 3.5 ile arama ajanını güncelle
    end if
end for
Herbir arama ajanının, arama alanının dışına çıkıp çıkmadığını kontrol et ve değeri değiştir

```

*Herbir arama ajanının uygunluk durumunu hesapla*  
*Daha iyi çözüm varsa, en iyi ajanın değerini güncelle*  
*t=t+1*  
*end while*  
*Sonuç X\**

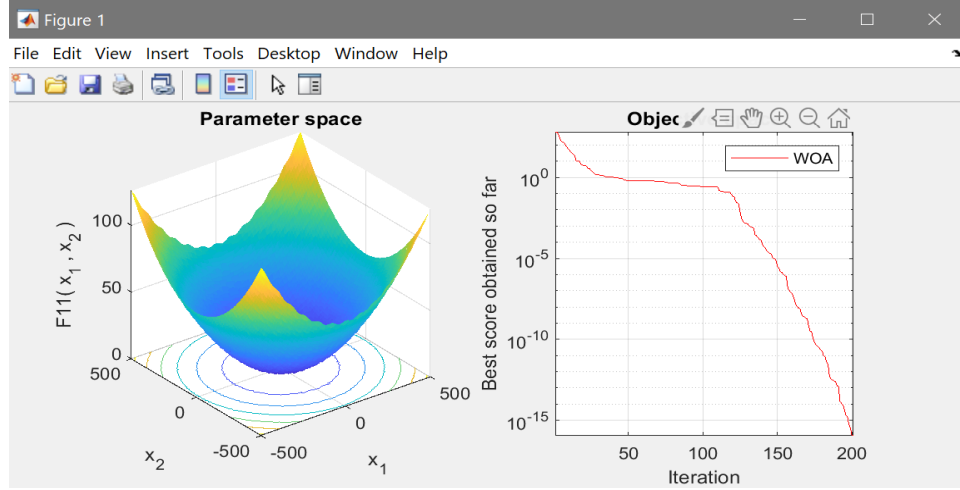
### 3.3. KALİTE TEST FONKSİYONLARI

Matematiksel test fonksiyonları, optimizasyon algoritmalarının performanlarını test etmek için kullanılan matematiksel ifadelerdir. Bu fonksiyonlar ile hem hız açısından hemde elde edilen çözüm açısından testler yapılabilir. Bir çok kalite test fonksiyonunun zorluk dereceleri, fonksiyonların parametreleri değiştirilerek ayarlanabilir. Tek modlu (tek optimum değeri olan) ve çok modlu (birden fazla lokal optimum değere ama sadece bir global optimum değeri olan) olmak üzere iki tane kalite test fonksiyonları vardır. Kalite test fonksiyonları, genetik algoritmalarda sık kullanılmaktadır.

#### 3.3.1. Griewank Fonksiyonu (F1)

Griewank fonksiyonu [-600,600] aralığında tanımlanmıştır. Matematiksel modeli Denklem 3.9'da verilmiştir.

$$f(\vec{x}) = \frac{1}{400} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (3.9)$$

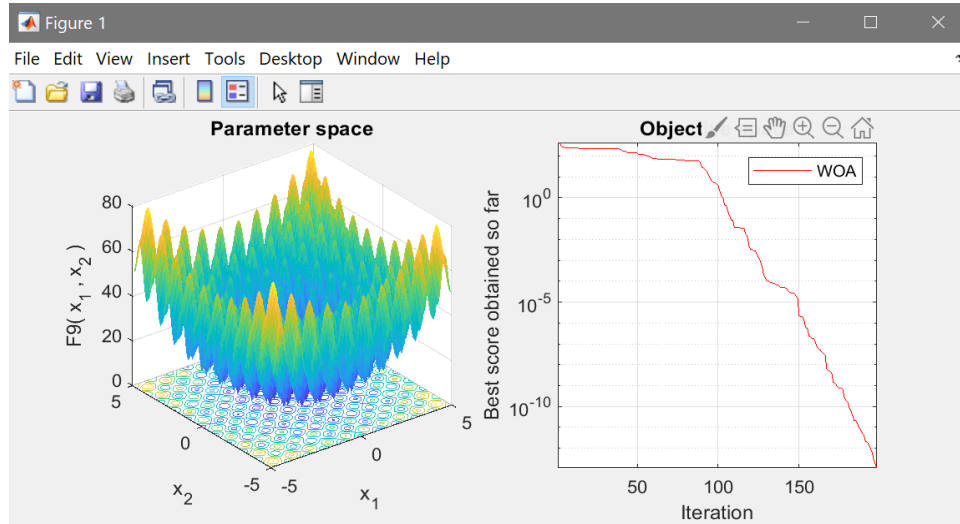


Şekil 3.5. Griewank fonksiyonu.

### 3.3.2. Sphere Fonksiyonu (F2)

Griewank fonksiyonu  $[-5.12, 5.12]$  aralığında tanımlanmıştır. Matematiksel modeli Denklem 3.10'da verilmiştir.

$$f(\vec{x}) = \frac{1}{899} \sum_{i=1}^6 x_i^2 2^i - 1745 \quad (3.10)$$

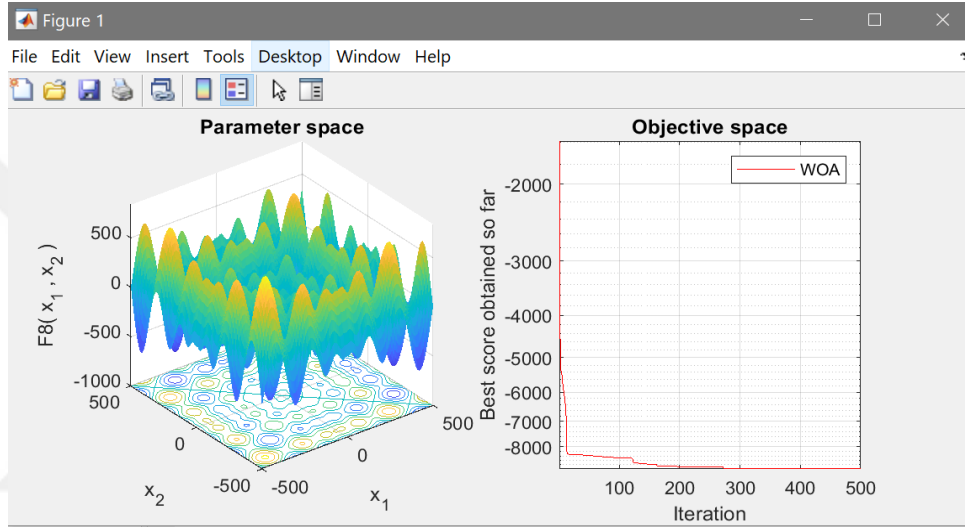


Şekil 3.6. Sphere fonksiyonu.

### 3.3.3. Schwefel Fonksiyonu (F3)

Griewank fonksiyonu  $[-5.12,5.12]$  aralığında tanımlanmıştır. Matematiksel modeli Denklem 3.11’de verilmiştir.

$$f(\vec{x}) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|}) \quad (3.11)$$



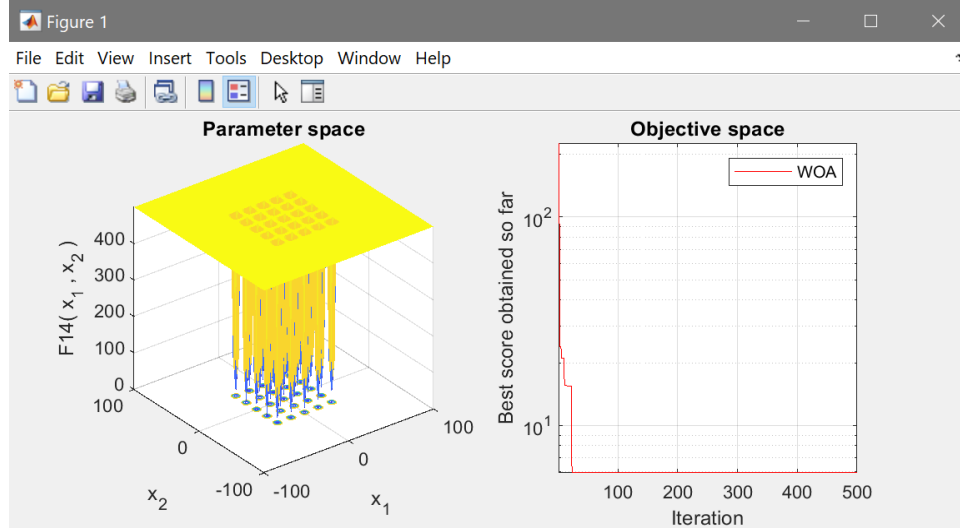
Şekil 3.7. Schwefel fonksiyonu.

### 3.3.4. De Jong Fonksiyonu (F4)

Griewank fonksiyonu  $[-65536,65536]$  aralığında tanımlanmıştır. Matematiksel modeli Denklem 3.12’de verilmiştir.

$$f(\vec{x}) = \left(0.002 + \sum_{i=1}^{25} \frac{1}{1+(x_1-a_{1i})^6+(x_2-a_{2i})^6}\right)^{-1}, \quad (3.12)$$

$$a = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{pmatrix}$$



Şekil 3.8. De Jong fonksiyonu.

Çizelge 3.1. BOA'nın kalite test fonksiyonları üzerindeki sonuçları.

Fonksiyon	BOA
Griewank Fonksiyonu (F1)	0
Sphere Fonksiyonu (F2)	0
Schwefel Fonksiyonu (F3)	-8999.7182
De Jong Fonksiyonu (F4)	5.9288

### 3.3.5. Kalite Test Fonksiyonlarının Sonuçları

Balina Optimizasyon Algoritmasını, kalite test fonksiyonlarının her biri için 500 iterasyon ve 30 defa çalıştırılarak ortalamaları bulunmuştur. Sonuçlar Çizelge 3.1'de verilmiştir. Ayrıca Şekil 3.5, Şekil 3.6, Şekil 3.7, Şekil 3.8'te de, test fonksiyonları ile çalıştırılan algoritmaların optimuma yakınsama grafikleri verilmiştir.

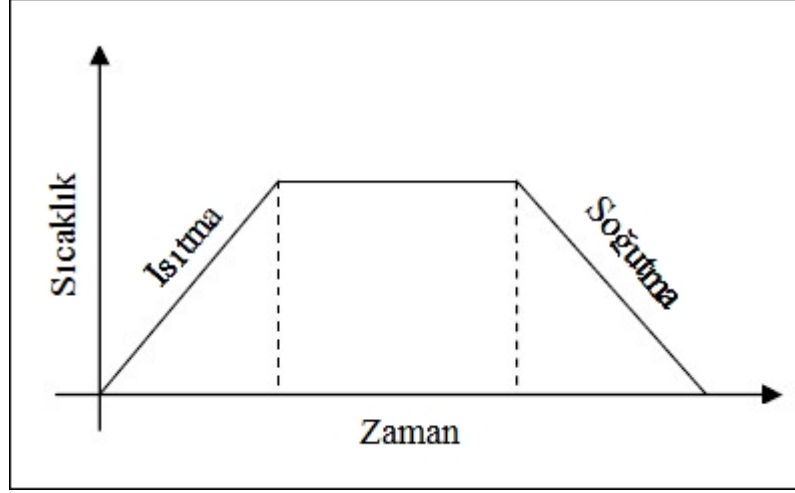
## BÖLÜM 4

### BENZETİLMİŞ TAVLAMA ALGORİTMASI

Bu bölümde BT algoritmasının tanımından ve tarihçesinden kısaca bahsedilerek çalışma prensibi anlatılacaktır. BT algoritmasına ait parametrelerin tanımları yapılacak, algoritmanın avantaj ve dezavantajlarından bahsedilecektir.

#### 4.1. GİRİŞ VE ARKAPLAN

Benzetilmiş Tavlama (Simulated Annealing) algoritması ismini demir tavlama anlamına gelen tavlama (annealing) kelimesinden almıştır. Tavlama, sistemin iç yapısındaki fiziksel veya kimyasal özelliklerini değiştirmek için uygulanan ısıtma ve soğutma işlemi olarak tanımlanır. Tavlama işleminin temel amacı, sistemin sıcaklığını arttırmak ve daha sonra yavaş yavaş soğutmak suretiyle mevcut veya rastgele oluşturulmuş durumlardan sistemin amaçlanan yapıya kavuşmasını sağlamaktır. Dolayısıyla, tavlama işlemi, fiziksel bir sistemin ısıtılması ve soğutulması süreçleri olarak iki adımdan oluşmaktadır. Öncelikle, fiziksel sistemin enerjisini yükseltmek için ısıtılma işlemi gerçekleştirilir, böylece atomların sistem içerisinde serbestçe dağılması sağlanarak dengesiz bir yapı elde edilir. Daha sonra sistemin istenilen yapısını elde etmek amacıyla sistem yavaş bir şekilde soğutulur. Soğutulma işlemi sürecinde atomların farklı enerji seviyelerine geçmesi sağlanarak sistemin yapısı üzerinde değişiklik sağlanır. Bu süreç, sistemin daha düzgün ve kararlı bir fiziksel yapıya dönüşmesine imkan tanır. Soğutma işlemi, fiziksel sistemin minimum enerji seviyesine ulaşana kadar devam eder. Bu enerji seviyesi ancak başlangıç veya maksimum sıcaklık yeterince yüksek olduğunda ve fiziksel sistem çok yavaş soğutulduğunda elde edilir. Katı bir cisim üzerinde gerçekleştirilen ısıtma işlemi süreci Şekil 4.1’de gösterilmiştir.



Şekil 4.1. Katı bir cisim için ısıl işlem süreci.

Genel olarak bir sezgisel algoritma, rastgele bir çözüm üretip arama uzayı alanındaki komşu çözümleri sistematik olarak araştırarak en iyi çözümü elde etmeyi amaçlar. SA algoritması, yukarıda bahsedilen tavlama işleminden esinlenerek geliştirilen yerel bir arama algoritmasıdır.

Metropolis vd. [9] 1953 yılında BT algoritmasının temellerini atmışlardır. Çalışmalarında, bir sistemin sabit sıcaklıkta termal denge düzeyine ulaşması için farklı enerji seviyeleri arasındaki geçişini incelemişlerdir. BT algoritmasının anahtar parçası olarak kabul edilen ve enerji seviyeleri arasındaki geçiş ihtimali için kullanılan Boltzmann dağılımını sunmuşlardır. Boltzmann dağılımına göre, Kirkpatrick vd. [10] 1983 yılında, Cerny vd. [11] 1985 yılında optimizasyon problemlerini çözmek için birbirlerinden bağımsız olarak BT tekniğini arama algoritması olarak kullanmışlardır.

## 4.2. METROPOLİS ALGORİTMASI VE BOLTZMANN DAĞILIMI

Metropolis algoritması ve Boltzmann dağılımı SA algoritmasının sezgisel bir algoritma olarak kullanılmasında önemli yer tutarlar. Metropolis vd. [9] yaptıkları çalışmada bir sistemin sabit sıcaklıktaki termal dengeye ulaşması için farklı enerji seviyeleri arasındaki değişimi incelemişlerdir. Bunun üzerine bir sistemin farklı enerji seviyeleri arasındaki termal denge geçişini referans alarak bir algoritma geliştirmişlerdir. Bu çalışmaların sonucunda Boltzmann dağılımı ortaya çıkmıştır. SA



algoritmasında arama uzayında gerçekleştirilen rastgele hareketler Boltzmann dağılımına bağlı olarak gerçekleştirilir. Termal denge durumunda, bir sistemin  $i$  durumundaki konfigürasyonuna ait  $E_i$  enerjisi ile olasılığı ( $P_i$ ) Boltzmann dağılımı ile aşağıdaki gibi gösterilir:

$$P_i = \frac{\exp\left(-\frac{E_i}{kT}\right)}{S} \quad (4.1)$$

$S$  olası tüm konfigürasyonları içermektedir:

$$S = \sum_{i=1}^N \exp(-E_i/kT) \quad (4.2)$$

Burada  $T$  tavlama sürecinde sistemin o anki sıcaklığı,  $k$  Boltzmann sabiti ve  $N$  ise toplam konfigürasyon sayısını ifade etmektedir ve olasılıklar toplamı aşağıdaki gibidir:

$$\sum_{i=1}^N p_i = 1 \quad (4.3)$$

### 4.3. ALGORİTMANIN ÇALIŞMA PRENSİBİ

Optimizasyon problemi çözümünde BT algoritmasının kullanılması, katı bir cismin tavlama sürecindeki düşük enerji seviyesini bulması arasındaki benzerliğe bağlı olarak geliştirilmiştir. Tavlama işleminde amaç, optimizasyon işleminde hedef fonksiyon değerini tavlama sürecindeki enerji seviyesi için yapıldığı gibi en aza indirmektir. Fiziksel sistemin atomlarının konfigürasyonu (dağılımı), optimizasyon problemine ait değişkenlere karşılık gelir. Sistemin enerji durumu, optimizasyon probleminin olası çözümleri ile aynıdır. Fiziksel sistemin mevcut ve aday (alternatif) durumları, optimizasyon probleminde mevcut ve aday (alternatif) çözümlerle çakışmaktadır. Tavlama işlemindeki sıcaklık, hedef çözüme yakınsama seviyesini belirlemek için optimizasyon işleminde bir kontrol ve sonlandırma parametresi olarak kullanılır. Son

olarak, yerel ve global minimum enerji durumları sırasıyla, optimizasyon problemindeki yerel ve global optimum çözümlere karşılık gelir. Çizelge 4.1 tavlama süreci ile BT algoritması arasındaki ilişkiyi göstermektedir [12].

Çizelge 4.1. Tavlama süreci ve BT algoritması arasındaki ilişki.

<b>Tavlama Süreci</b>	<b>BT Algoritması</b>
Enerji Durumu	Olası Çözüm
Enerji Seviyesi	Amaç Fonksiyonu
Atomlarının Konfigürasyonu	Problem Değişkenlerinin Kümesi
Mevcut Durum	Mevcut Çözüm
Aday Durum	Aday Çözüm
Yerel Minimum Enerji	Yerel Minimum Çözüm
Global Minimum Enerji	Global Minimum Çözüm

BT algoritmasının sözde kodu (pseudocode) aşağıda gösterilmiştir:

$s \leftarrow s_0; e \leftarrow E(s)$	// İlk durum, enerji
$s_{best} \leftarrow s; e_{best} \leftarrow e$	// Başlangıcı en iyi çözüm ata
$k \leftarrow 0$	// İterasyon sayısı
<i>while</i> $k < k_{max}$ <i>and</i> $e > e_{max}$	// İterasyon bitene devam et
$T \leftarrow \text{temperature}(k/k_{max})$	// Sıcaklığı azalt
$s_{new} \leftarrow \text{neighbor}(s)$	// Yeni bir komşu çözüm seç
$e_{new} \leftarrow E(s_{new})$	// Enerjiyi hesapla
<i>if</i> $P(e, e_{new}, T) > \text{random}()$ <i>then</i>	// Yeni durum kabul edilebilir mi?
$s \leftarrow s_{new}; e \leftarrow e_{new}$	// Evet ise mevcut durumu değiştir
<i>if</i> $e < e_{best}$ <i>then</i>	// Enerji daha düşükse en iyi çözüm
<i>yap</i>	
$s_{best} \leftarrow s_{new}; e_{best} \leftarrow e_{new}$	// Mevcut durumu en iyisiyle
<i>değiştir</i>	
$k \leftarrow k + 1$	// İterasyon sayısını artır
<i>return</i> $s_{best}$	// En iyi sonuçla bitir

Sözde kodda görüldüğü üzere bir iterasyon sayısının varlığı bir sonraki adım için mevcut adımın tamamlanmasını gerektirmektedir. Seri algoritmada her adımda tek bir çözüm bulunur ve mevcut ile karşılaştırılır. Günümüzde kullanılan paralelleştirme yöntemleri kullanılarak her adımın iş parçacıklarına dağıtılmasıyla birden fazla çözüm üretilip bunlar için en uygun olanı seçebilmek mümkündür. Bu sayede seri algoritma için gereken süre içerisinde paralel bir yöntemde birden fazla çözüm üretilerek daha iyi sonuçların bulunabilme ihtimali de artmış olmaktadır.

BT algoritması temelde dört adet parametreye sahiptir ve bu parametre değerlerinin seçimi problemin çözüm kalitesinde yüksek derecede etkilidir. Bu parametreler ise şunlardır:

- Başlangıç Sıcaklığı,
- Soğutma Katsayısı,
- Hedef Sıcaklık ve
- İterasyon Sayısıdır.

#### **4.3.1. Başlangıç Sıcaklığı**

BT algoritmasında başlangıç sıcaklığının belirlenmesi önemli bir olgudur. Başlangıç sıcaklığının yüksek bir değer seçilmesi işlem süresini uzatır ve gereksiz hesaplama neden olur. Diğer yandan düşük bir sıcaklık ile algoritmanın çalıştırılması yerel optimum değerlere takılma olasılığını artırır. Bu parametrenin seçilmesi için bir kestirimde bulunmak mümkün değilse yüksek bir sıcaklığın seçilmesi optimizasyon probleminin en iyi çözümüne ulaşabilmek adına faydalı olacaktır.

#### **4.3.2. Soğutma Katsayısı**

Soğutma katsayısı  $\alpha$  ile temsil edilmektedir. Genellikle 0 ile 1 arasında herhangi bir değer olarak mevcut sıcaklık ile çarpılarak düşürülür. Sıcaklığın düşürüldüğü farklı yöntemler de mevcuttur. Soğutma katsayısı 0'a yaklaştıkça sistem daha hızlı soğumakta, 1'e yaklaştıkça ise daha yavaş soğumaktadır.

Sıcaklık azaltma fonksiyonu farklı biçimlerde kurulabilir:

- Aritmetik ( $T_k = T_{k-1} - C$ )
- Geometrik ( $T_k = T_{k-1} * \alpha$ )
- Ters fonksiyon ( $T_k = C / (1 + k)$ )
- Logaritmik ( $T_k = C / (\text{Log}(1 + k))$ )

Burada  $T_k$  azaltmadan sonraki sıcaklığı temsil etmektedir.  $C$  başlangıç sıcaklığından küçük bir değer ve  $\alpha$  da 0 ile 1 arasında soğutma katsayısı olarak ifade edilen bir değerdir. Yapılan çalışmalarda  $\alpha$  değeri genellikle 0.8 ile 0.99 arasında bir değer almaktadır.

#### 4.3.3. Hedef Sıcaklık

Hedef sıcaklık değeri durdurma kuralı olarak kullanılan değerdir. Algoritma başlangıç sıcaklığı ile çözüm adımlarına başlayarak her adımda belirlenen soğutma katsayısı ile sıcaklığı düşürür. BT algoritması belirlenen bir hedef sıcaklığa ulaşıncaya kadar işlemlerine devam eder. Hedef sıcaklığa erişilince yöntem durdurulur. Bu kuralla beraber belli bir döngü sonucunda herhangi bir iyileştirme olmuyorsa, gereksiz işlemleri bitirmek adına yöntem durdurulabilir. Durdurma kriteri için kullanılacak bir diğer yöntem ise zamandır. Yöntem zaman yönünden kısıtlanıp belli bir zaman aralığı içinde bulunan en iyi sonuca odaklanabilir.

#### 4.3.4. İterasyon Sayısı

İterasyon sayısı, BT algoritmasının birden fazla çalıştırılmasını ifade etmektedir. Her bir iterasyonda BT belirlenen parametrelerle tekrar adımlarına başlar. İlk iterasyondan sonraki her iterasyon başlangıcında bir önceki iterasyonda elde edilen çözüm ile devam edilir. Bu sayede global en iyi çözüme ulaşmak amaçlanır.

#### **4.4. ALGORİTMANIN AVANTAJLARI VE DEZAVANTAJLARI**

BT algoritması birçok çalışmada farklı yöntemler ile karşılaştırılmış, problemin özgünlüğüne ve parametrelerine göre başarılı ya da başarısız sonuçlar vermiştir. Yine BT algoritmasının performansını artırmak amaçlı birçok hibrit yöntem geliştirilmiş ve bu yöntemler de yine diğer sezgisel yöntemlerle karşılaştırılarak analiz edilmiştir.

##### **4.4.1. Algoritmanın Güçlü Yönleri**

BT algoritması bir çok optimizasyon problemleri ile başa çıkabilen, yerel optimuma takılma riskine karşı dayanıklı bir algoritmadır. Diğer algoritmalar ile karşılaştırıldığında en önemli farkı, esnek yani probleme uyarlanabilir olması ve en iyi çözüme yakın çözümü elde etmedeki yeteneğidir. Algoritma, kısıtlayıcı özelliklere sahip olmamasından ötürü çok yönlü bir algoritmadır. Algoritmanın adımları her problem için rahatlıkla düzenlenebilir yapıdadır. Yine bir başka önemli özelliği bir çok optimizasyon algoritması ile entegreli biçimde çalışarak algoritmaya güç katarak performansı artırabilme yeteneğine sahip olmasıdır [13].

##### **4.4.2. Algoritmanın Zayıf Yönleri**

BT algoritması sezgisel bir yöntem olduğundan, gerçek bir algoritmaya dönüştürebilmek için birçok seçenek gereklidir. En önemli dezavantajlarından birisi hesaplama için fazla zamana ihtiyaç duymasıdır. Parametreler değerlerinin belirlenmesi hassas bir işlem olup sonuç üzerinden direkt olarak etki ettiği için bu da algoritmanın eksi bir yönü olarak söylenebilir. Yine hassas hesaplama isteyen modellerde BT algoritması sonucun kalitesine erişmek için fazlaca işlem ve zamana ihtiyaç duyabilir [13].

## BÖLÜM 5

### KARESEL ATAMA PROBLEMİ

Bu bölümde Karesel Atama Problemi ile ilgili bilgiler verilmiştir.

#### 5.1. KARESEL ATAMA PROBLEMİ (KAP)

Karesel Atama Problemi (Quadratic Assignment Problem, QAP) kombinatoriyal optimizasyon problemlerinden biri olan ve NP-zor sınıfında gösterilen uygulama alanı geniş bir problemdir. Problem, eşit sayıda tesis (facility) ve bölge (location) içermekte ve tesislerin bölgelere yerleşimini en az maliyet ile gerçekleştirmeyi hedeflemektedir. İlk olarak 1957 yılında Koopmans ve Beckmann tarafından sunulmuştur [14]. Problemin çözümünden sonra her tesisin bir bölgeye atanması sağlanmalı ve hiçbir tesisin boşta kalmaması gerekmektedir. KAP için toplam maliyet, probleme atanan herhangi bir permütasyon ( $\pi$ ) için aşağıdaki gibi formülize edilebilir:

$$c(\pi) = \sum_{n=1}^n d_{i,j} \cdot f_{\pi(i),\pi(j)} \quad (5.1)$$

*d*: *i* bölgesi ile *j* bölgesi arasındaki mesafe,

*f*: *i* tesisi ile *j* tesisi arasındaki birim uzaklığın maliyeti,

*n*: *d* ve *f* değerlerini içeren matrislerin boyutu,

*c*: maliyet.

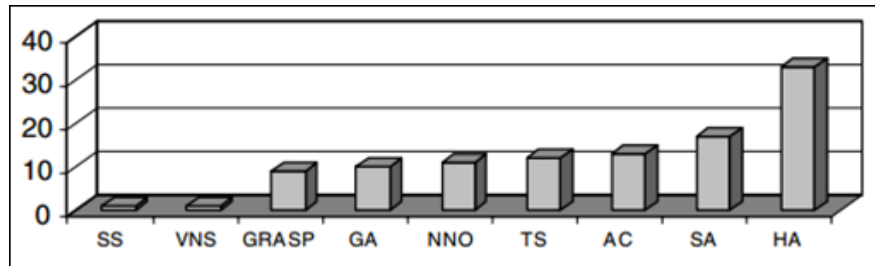
KAP; pota kablolama bileşenleri arasındaki bağlantı sayısını tespit etmede [15], zaman çizelgeleme probleminde [16], daktilo klavyesi ve kontrol paneli tasarımında [17], bilgisayar destekli plan tasarımında [18], nümerik analizde [19] kullanılmıştır. Bununla birlikte en yaygın kullanım alanı tesis yerleşim problemi olarak görülmektedir. Dickey ve Hopkins üniversite yerleşkesi içerisindeki binaların

yerleşiminde [20], Elshafei hastane yerleşim planında [21], Bos ise orman parklarının imarı ile ilgili bir problemde [22] KAP'dan yararlanmışlardır.

### 5.1.1. KAP'ın Çözüm Yöntemleri

KAP NP-zor sınıfı bir problem olduğu için problemin boyutu arttıkça çözüm zamanı da üstel olarak artmaktadır. KAP için global en iyi çözüme ulaşmak amaçlı kullanılan değişik yöntemler vardır. Bunlar, dal-sınır, kesen düzlemler ya da bu yöntemlerin kombinasyonları olan dal-kesim yöntemi ve dinamik programlamadır. Dal-sınır en bilinen ve kullanılan yöntemlerin başında yer alır. Son yıllarda, dal-sınır teknikleri ile kombine edilen yöntemlerin paralel uygulamaları literatürde yer almaktadır. Bununla birlikte başarılı sonuçlar elde edilmeye başlanmıştır. Ancak problemin boyutu ve donanım teknolojisi burada önemli rol oynamaktadır [9]. KAP NP-zor bir problem olduğundan ve kesin çözüm yöntemlerinin yüksek boyutlu problemlerdeki sorunlarından dolayı makul bir sürede en iyi ya da en iyi çözüme yakın sonucu sağlayan sezgisel yöntemler tercih edilmektedir.

KAP'ın çözümünde kullanılan sezgisel yöntemlere örnek olarak Benzetilmiş Tavlama (SA), Genetik Algoritma (Genetic Algorithm, GA), Sinir Ağları (Neural Network, NNO), Dağılım Araması (Scatter Search, SS), Karınca Kolonisi (Ant Colony, AC), Tabu Arama (Tabu Search, TS), Açgözlü Rastgele Adaptif Arama Yöntemi (Greedy Randomized Adaptive Search Procedure, GRASP), Değerli Komşuluk Arama (Variable Neighborhood Search, VNS) verilebilir. Bununla birlikte farklı sezgisel yöntemlerin bir arada kullanıldığı hibrit çalışmalar da literatürde mevcuttur. Şekil 5.1'de ise KAP için kullanılan sezgisel yöntemlerin dağılımı gösterilmiştir.

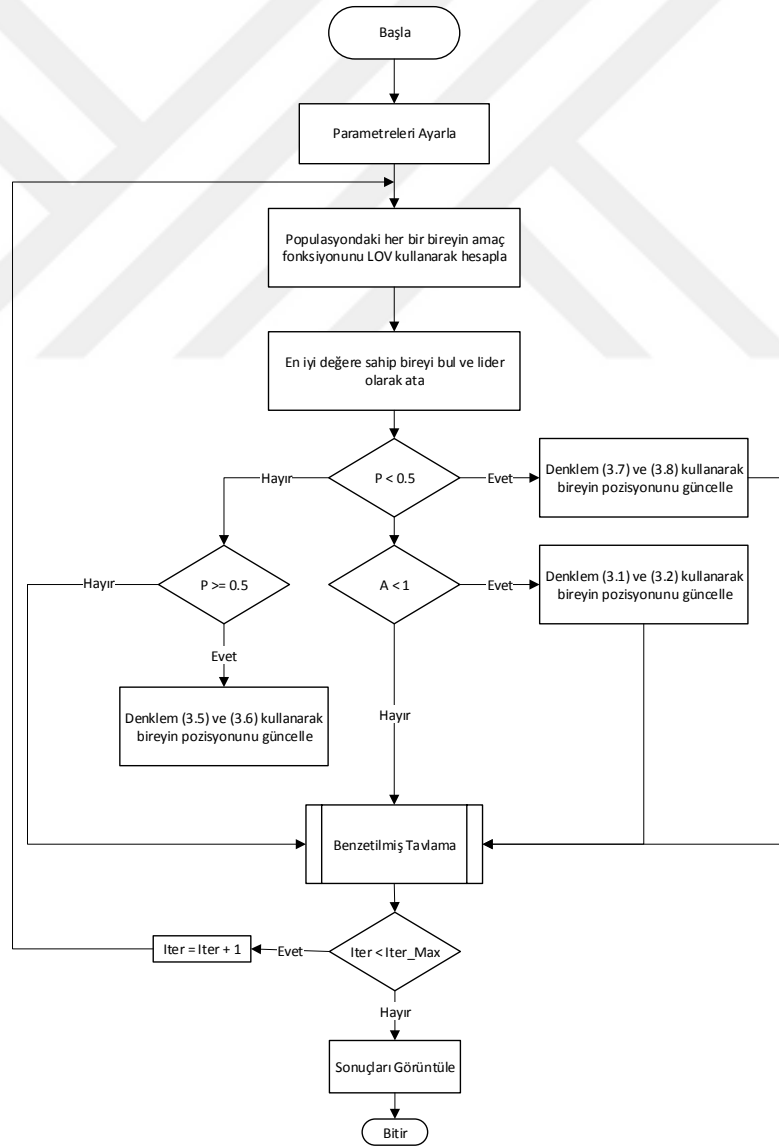


Şekil 5.1. KAP için kullanılan sezgisel yöntemlerin dağılımı [24].

## BÖLÜM 6

### DENEYSEL ÇALIŞMALAR

Bu tezde benzetilmiş tavlama hibrit BOA'nın (BOA-BT) KAP üzerinde nasıl bir sonuç vereceğine yönelik bir yöntem geliştirilmiştir. Geliştirilen yöntemin akış çizelgesi Şekil 6.1'de gösterilmiştir.



Şekil 6.1. BOA-BT'nin akış diyagramı.



Şekil 6.1’de görülen Benzetilmiş Tavlama algoritmasında ise adımlar şunlardır:

- Lider çözüm başlangıç çözümü olarak BT’ye verilir.
- BT bu çözüm üzerinde ikili yer değiştirme uygulayarak yeni bir çözüm üretir.
- Yeni çözüm BT kriterlerine göre kabul ya da ret edilir.
- Yeni çözüm lider çözümden iyi ise lider çözüm güncellenir.
- Bu aşamalar BT’nin durdurma kriterleri bitene kadar devam eder.

BOA sürekli optimizasyon problemleri için geliştirilmiş bir algoritmadır. KAP kombinatoriyel bir optimizasyon problemi olduğu için BOA’nın çözüm uzayını sürekli uzaydan ayrık uzaya dönüştürmek amacıyla Largest Order Value (LOV) yöntemi kullanılmıştır. LOV’un çalışma prensibi (Çizelge 6.1) aşağıdaki gibidir:

- Önce dizi büyükten küçüğe doğru sıralanır.
- Sıralanan dizinin indeks değerleri alınır.
- Bu indeksler kullanılarak dizi aşağıdaki gibi elde edilir.

Çizelge 6.1. LOV’un çalışma prensibi.

İndeks	1	2	3	4	5	6
Sürekli Uzay	1.2	3.6	0.56	8.97	5.43	1.12
LOV Aşama 1.	4	5	2	1	6	3
Kombinatoriyal Uzay	4	3	6	1	2	5

Bu yöntemle 20 balina popülasyonu ve 5000 iterasyon ile her bir QAPLIB veri setinin 10 defa çalıştırılması ile elde edilen sonuçlar Çizelge 6.1’de verilmiştir.

Değerlendirme yapılırken karşılaştırma kriteri olarak Bağlı Yüzde Sapma (Relative Percentage Deviation, RPD) kullanılmıştır. RPD aşağıdaki formül ile ifade edilmektedir:

$$RPD = \frac{cost(\pi) - cost(\pi^*)}{cost(\pi^*)} \times 100\% \quad (6.1)$$

$cost(\pi)$  geliştirilen yöntem ile bulunan sonucu,  $cost(\pi^*)$  ise Bilinen En İyi Sonucu (Best Known Solution, BKS) temsil etmektedir.

Çizelge 6.2. BOA'nın QAPLIB veri setlerine uygulama sonucu çıkan sonuçlar.

Veri Seti	BKS	Bulunan En İyi Değer	RPD (%)	Standart sapma
chr12a	9552	<b>9552</b>	<b>0</b>	<b>0</b>
chr12b	9742	<b>9742</b>	<b>0</b>	<b>0</b>
chr12c	11156	<b>11156</b>	<b>0</b>	<b>12</b>
chr15a	9896	<b>9896</b>	<b>0</b>	<b>18.3303</b>
chr15b	7990	<b>7990</b>	<b>0</b>	<b>464.7615</b>
chr15c	9504	<b>9504</b>	<b>0</b>	<b>281.3701</b>
chr18a	11098	<b>11098</b>	<b>0</b>	<b>656.863</b>
chr18b	1534	<b>1534</b>	<b>0</b>	<b>0</b>
chr20a	2192	2224	0.0146	61.62597
chr20b	2298	<b>2298</b>	<b>0</b>	<b>60.94719</b>
chr20c	14142	<b>14142</b>	<b>0</b>	<b>605.0461</b>
chr22a	6156	<b>6156</b>	<b>0</b>	<b>81.81565</b>
chr22b	6194	<b>6194</b>	<b>0</b>	<b>78.8165</b>
chr25a	3796	<b>3796</b>	<b>0</b>	<b>137.8116</b>
esc16a	68	<b>68</b>	<b>0</b>	<b>0</b>
esc16b	292	<b>292</b>	<b>0</b>	<b>0</b>
esc16c	160	<b>160</b>	<b>0</b>	<b>0</b>
esc16d	16	<b>16</b>	<b>0</b>	<b>0</b>
esc16e	28	<b>28</b>	<b>0</b>	<b>0</b>
esc16f	0	<b>0</b>	<b>0</b>	<b>0</b>
esc16g	26	<b>26</b>	<b>0</b>	<b>0</b>
esc16h	996	<b>996</b>	<b>0</b>	<b>0</b>
esc16i	14	<b>14</b>	<b>0</b>	<b>0</b>
esc32a	130	132	0.0153	6.352952
esc32b	168	184	0.0952	3.919184
esc32c	642	<b>642</b>	<b>0</b>	<b>0</b>
esc32d	200	<b>200</b>	<b>0</b>	<b>0.6</b>
esc32e	2	<b>2</b>	<b>0</b>	<b>0</b>
esc32g	6	<b>6</b>	<b>0</b>	<b>0</b>
esc32h	438	<b>438</b>	<b>0</b>	<b>0.916515</b>
esc64a	116	<b>116</b>	<b>0</b>	<b>0</b>
esc128	64	66	0.0312	4.433319
had12	1652	<b>1652</b>	<b>0</b>	<b>0</b>

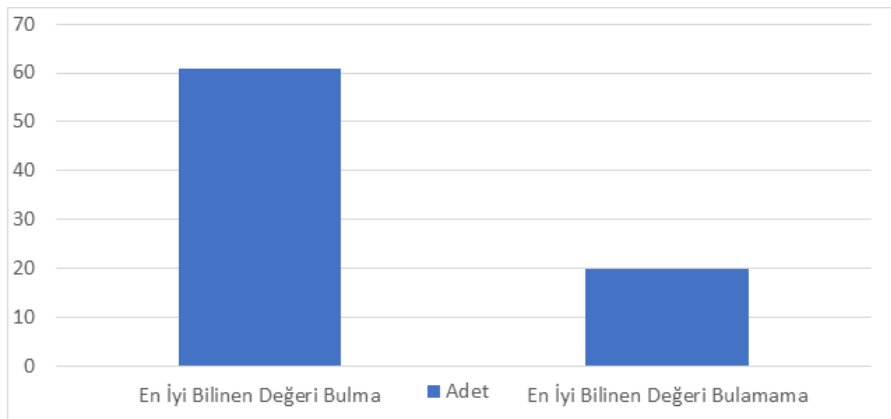
Çizelge 6.2. (devam ediyor).

Veri Seti	BKS	Bulunan En İyi Değer	RPD (%)	Standart sapma
had14	2724	<b>2724</b>	<b>0</b>	<b>0</b>
had16	3720	<b>3720</b>	<b>0</b>	<b>0</b>
had18	5358	<b>5358</b>	<b>0</b>	<b>0</b>
had20	6922	<b>6922</b>	<b>0</b>	<b>0</b>
kra30a	88900	90160	0.0141	807.948
kra30b	91420	<b>91420</b>	<b>0</b>	<b>969.8603</b>
kra32	88700	89870	0.0131	1082.462
lipa20a	3683	<b>3683</b>	<b>0</b>	<b>0</b>
lipa20b	27076	<b>27076</b>	<b>0</b>	<b>0</b>
lipa30a	13178	13241	0.004781	57.23539
lipa30b	151426	172276	0.137691	871.4136
lipa40a	31538	31924	0.012239	25.08087
lipa40b	476581	<b>476581</b>	<b>0</b>	<b>37561.49</b>
lipa50a	62093	62839	0.012014	28.87144
lipa60a	107218	108309	0.010176	55.40794
lipa70a	169755	171298	0.00909	49.41346
lipa80a	253195	255313	0.008365	81.91709
lipa90a	360630	363390	0.007653	127.6896
nug12	578	<b>578</b>	<b>0</b>	<b>0</b>
nug14	1014	<b>1014</b>	<b>0</b>	<b>0</b>
nug15	1150	<b>1150</b>	<b>0</b>	<b>0</b>
nug16a	1610	<b>1610</b>	<b>0</b>	<b>0</b>
nug16b	1240	<b>1240</b>	<b>0</b>	<b>0</b>
nug17	1732	<b>1732</b>	<b>0</b>	<b>0.8</b>
nug18	1930	<b>1930</b>	<b>0</b>	<b>2.4</b>
nug20	2570	<b>2570</b>	<b>0</b>	<b>3.155947</b>
nug21	2438	<b>2438</b>	<b>0</b>	<b>2.749545</b>
nug22	3596	<b>3596</b>	<b>0</b>	<b>3.4</b>
nug24	3488	<b>3488</b>	<b>0</b>	<b>15.65759</b>
nug25	3744	<b>3744</b>	<b>0</b>	<b>9.641577</b>
nug27	5234	<b>5234</b>	<b>0</b>	<b>12.07311</b>
nug30	6124	6166	0.006858	20.59126
rou12	235528	<b>235528</b>	<b>0</b>	<b>1353</b>

Çizelge 6.2. (devam ediyor).

Veri Seti	BKS	Bulunan En İyi Değer	RPD (%)	Standart sapma
rou15	354210	<b>354210</b>	<b>0</b>	<b>4506.217</b>
rou20	725522	726920	0.001927	2827.634
scr12	31410	<b>31410</b>	<b>0</b>	<b>0</b>
scr15	51140	<b>51140</b>	<b>0</b>	<b>0</b>
scr20	110030	<b>110030</b>	<b>0</b>	<b>407.9353</b>
sko42	15812	15894	0.005186	58.33044
tai10a	135028	<b>135028</b>	<b>0</b>	<b>0</b>
tai12a	224416	<b>224416</b>	<b>0</b>	<b>0</b>
tai15a	388214	<b>388214</b>	<b>0</b>	<b>858.4318</b>
tai17a	491812	<b>491812</b>	<b>0</b>	<b>1510.171</b>
tai20a	703482	<b>703482</b>	<b>0</b>	<b>4160.143</b>
Wil50	48816	48984	0.003441	92.21735
Wil100	273038	275060	0.007406	382.5143
tho30	149936	<b>149936</b>	<b>0</b>	<b>1270.858</b>
tho40	240516	241460	0.003925	1663.652
		<b>Ortalama</b>	<b>0.005114222</b>	<b>782.1725036</b>

Bulunan sonuçlara göre toplamda 81 veri seti içinde 61 adet veri setine ait bilinen en iyi sonuçlar BOA-BT tarafından bulunmuştur (Şekil 7.1). 20 adet veri setinde ise bilinen en iyi sonuç bulunmamasına rağmen, ortalama RPD değeri yaklaşık olarak 0.005 bulunmuştur. Bu değer geliştirilen algoritmanın güçlü ve rakabetçi olduğunu ortaya koymaktadır.



Şekil 6.2. BOA-BT'nin QAPLIB üzerindeki performansı.

## KAYNAKLAR

1. Tanyıldızı, E. vd. , “Kaotik Haritalı Balina Optimizasyon Algoritmaları”, *Fırat Üniv. Bil. Dergisi*, S. 309-319 (2017).
2. Abdel-Basset, Mohamed vd., "A Hybrid Whale Optimization Algorithm Based on Local Search Strategy for The Permutation Flow Shop Scheduling Problem", *Future Generation Computer Systems*, S.85:129-145 (2018).
3. Mafarja, Majdi M. vd, " Hybrid Whale Optimization Algorithm with Simulated Annealing For Feature Selection", *Neurocomputing*, S.260:302-312 (2017).
4. Dokeroğlu, Tansel, "Hybrid Teaching–Learning-Based Optimization Algorithms For Quadratic Assignment Problem", *Computers&Industrial Engineering*, S.85:86-101 (2015).
5. Sonuç, Emrullah vd., "A Cooperative GPU-Based Parallel Multistart Simulated Annealing Algorithm For Quadratic Assignment Problem", *Engineering Science and Technology, an International Journal* (2018).
6. Abdel-Basset Mohamed vd., "A Modified Nature Inspired Meta-Heuristic Whale Optimization Algorithm For Solving 0-1 Knapsack Problem", DOI 10.1007/s13042-017-0731-3 (2017).
7. İnternet: Wikipedia, "Kambur Balina | Wikipedia", [https://tr.wikipedia.org/wiki/Kambur\\_balina/](https://tr.wikipedia.org/wiki/Kambur_balina/) (2019)
8. Mirjalili Seyedali vd., "The Whale Optimization Algorithm", *Advances in Engineering Software* 95, S.51-67 (2016).
9. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equation of state calculations by fast computing machines", *The Journal Of Chemical Physics*, 21 (6): 1087–1092 (1953).
10. Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by simulated annealing", *Science*, 220 (4598): 671–680 (1983).
11. Černý, V., "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm", *Journal Of Optimization Theory And Applications*, 45 (1): 41–51 (1985)
12. Şahin, S., "Çelik kafes enerji nakil hattı direklerinin benzetimsel tavlama yöntemi kullanılarak bilgisayar destekli optimizasyonu", Doktora Tezi, *Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İnşaat Mühendisliği A.B.D.*, Ankara, (2016)

13. Īnternet: Busetti, F., "Simulated Annealing Overview", [http://personalpages.to.infn.it/~ferraro/retineurali/articoli\\_rn/simantut.pdf](http://personalpages.to.infn.it/~ferraro/retineurali/articoli_rn/simantut.pdf) (2003).
14. Koopmans, T. C. and Beckmann, M., "Assignment problems and the location of economic activities", *Econometrica: Journal Of The Econometric Society*, 53–76 (1957).
15. Han, Y., Roy, S., and Chakraborty, K., "Optimizing simulated annealing on GPU: A case study with IC floorplanning", *Quality Electronic Design (ISQED), 2011 12th International Symposium on*, 1–7 (2011).
16. Fobel, C., Gréwal, G., Collier, R., and Stacey, D., "GPU Approach to FPGA placement based on star+", *New Circuits and Systems Conference (NEWCAS), 2012 IEEE 10th International*, 229–232 (2012).
17. Li, H. and Liu, C., "Prediction of protein structures using GPU based simulated annealing", *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, 630–633 (2012).
18. Fabry-Asztalos, L., Lörentz, I., and Andonie, R., "Molecular distance geometry optimization using geometric build-up and evolutionary techniques on GPU", *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2012 IEEE Symposium on*, 321–328 (2012).
19. Fobel, C., Grewal, G., and Stacey, D., "A scalable, serially-equivalent, high-quality parallel placement methodology suitable for modern multicore and GPU architectures", *Field Programmable Logic and Applications (FPL), 2014 24th International Conference on*, 1–8 (2014)
20. Kataoka, M., Tsukiyamia, S., Kambe, T., and Fukui, M., "An effective method to use GPU for rectangle packing", *New Circuits and Systems Conference (NEWCAS), 2012 IEEE 10th International*, 129–132 (2012).
21. Ferreiro, A., García, J., López-Salas, J. G., and Vázquez, C., "An efficient implementation of parallel simulated annealing algorithm in GPUs", *Journal Of Global Optimization*, 57 (3): 863–890 (2013).
22. Stivala, A. D., Stuckey, P. J., and Wirth, A. I., "Fast and accurate protein substructure searching with simulated annealing and GPUs", *BMC Bioinformatics*, 11 (1): 446 (2010).
23. Īnternet: Code Capsule, "Simulated Annealing Applied to the Traveling Salesman Problem | Code Capsule", <http://codecapsule.com/2010/04/06/simulated-annealing-traveling-salesman/> (2017).
24. Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., and Querido, T., "A survey for the quadratic assignment problem", *European Journal Of Operational Research*, 176 (2): 657–690 (2007).

## ÖZGEÇMİŞ

Adem BÜYÜKÖZER 1983 yılında Muş'ta doğdu. İlkokulu Hürriyet İlkokulu'nda tamamladı. Ortaokul eğitimini Muş İmam-Hatip Okulu'nda, liseyi ise Muş Süper Lisesi'nde tamamlayarak 2001 yılında mezun oldu. 2001 yılında Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nü kazanarak bu bölümde lisans öğrenimine başladı. 2006 yılında lisans eğitimini tamamladı. 2010 yılında Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda başladığı yüksek lisans eğitimini devam ettirmektedir. Özel sektörde ve kamu görevlisi olarak bir çok görevde yer alıp 2014 yılından beri kurmuş olduğu grup şirketlerini yönetmektedir.

### **ADRES BİLGİLERİ**

Adres : Eymir Mah. Toki Küme Evleri C-60 Blok No:19  
Gölbaşı/ANKARA  
Tel : 0 (532) 292 99 50  
E-posta : adembuyukozer@hotmail.com