



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

**ROBOT YOL PLANLAMASI İÇİN GRİ KURT
OPTİMİZASYON ALGORİTMASI**

Lokman DOĞAN

Yüksek Lisans Tezi

Tez Danışmanı

Doç. Dr. Uğur YÜZGEÇ

Bilecik, 2018

Ref.No: 10198046



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

**ROBOT YOL PLANLAMASI İÇİN GRİ KURT
OPTİMİZASYON ALGORİTMASI**

Lokman DOĞAN

Yüksek Lisans Tezi

Tez Danışmanı

Doç. Dr. Uğur YÜZGEÇ

BİLECİK, 2018



BİLECİK ŞEYH EDEBALI UNIVERSITY

Graduate School of Science

Department of Computer Engineering

**GREY WOLF OPTIMIZATION ALGORITHM FOR
ROBOT PATH PLANNING**

Lokman DOĞAN

Master's Thesis

Thesis Advisor

Assoc. Prof. Uğur YÜZGEÇ

BİLECİK, 2018



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS

JÜRİ ONAY FORMU

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun 13/06/2018 tarih ve 5 sayılı kararıyla oluşturulan jüri tarafından 29/06/2018 tarihinde tez savunma sınavı yapılan Lokman DOĞAN'ın "ROBOT YOL PLANLAMASI İÇİN GRİ KURT OPTİMİZASYON ALGORİTMASI" başlıklı tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak oy birliği/ oy çokluğu ile kabul edilmiştir.

JÜRİ

ÜYE (TEZ DANIŞMANI) : Doç. Dr. Uğur YÜZGEÇ

ÜYE : Doç. Dr. Cihan KARAKUZU

ÜYE : Dr. Öğr. Üyesi Suhap ŞAHİN

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANI:

Doç. Dr. Uğur YÜZGEÇ

ONAY

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun tarih ve sayılı kararı.

* İMZA/ MÜHÜR

TEŐEKKÖR

Çalışmalarım boyunca yardım, katkı ve tecrübeleriyle beni yönlendiren, tavsiyeleri ve rehberliğı sayesinde çalışmalarımı kolaylaştıran ve bir hocadan çok bana bir arkadaş gibi yaklaşan danışman hocam Doç. Dr. Uğur YÜZGEÇ'e, teşekkürlerimi bir borç bilirim. Ayrıca çalışma arkadaşlarıma, yoğun çalışmalarım yüzünden fazla zaman ayıramama rağmen sabır gösteren manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme canı gönülden teşekkür ederim.

Lokman DOĞAN



ÖZET

Yol planlama problemi, mobil robot çalışmalarında önemli bir rol oynamaktadır. Robot sistemlerinde, robotun yolunu bir noktadan diğerine planlamak için farklı teknikler kullanılmaktadır. Yol planlama probleminde ana amaç, bir robotun engeller bulunan bir ortamda, başlangıç konumundan hedef konumuna kadar engellere çarpmadan en kısa yolu bulmaktır. Bu çalışmada, Gri Kurt Optimizasyon algoritması robot yol planlama problemine uyarlanmıştır. Gri Kurt Optimizasyon algoritması (Gray Wolf Optimizer-GWO), doğadaki gri kurtların avlanma davranışını ve sosyal liderliğini taklit eder. Liderlik hiyerarşisi dört gruptan oluşur. Bunlar alfa, beta, delta ve omega gruplarıdır. Bu algorithmada avlanma mekanizması üç aşamadan meydana gelir. Bunlar avı aramak, avı kuşatmak ve ava saldırmak. Robot yol planlamasının test çalışmalarında, üç, dört ve beş dairesel engelli haritalar kullanılmıştır. Yol planlamasında aday çözümler bulunurken, başlangıç ve hedef noktalar arasında üç koordinat noktası kullanılmıştır. Her yinelemede, bu koordinat noktaları GWO algoritması tarafından çözüme daha yakın olacak şekilde güncellenir. Çözüm noktası engel bölgesinde ise, maliyet fonksiyonuna ihlal değeri eklenir. Bu tez çalışmasında, GWO algoritmasının performansı, robot yol planlama problemini çözmek için diğer meta-sezgisel algoritmalarla beraber değerlendirilmiştir. GWO algoritması ile elde edilen sonuçlar, kullanılan test haritası için en uygun yolun bulunduğunu göstermektedir.

Anahtar sözcükler: Robot Yol Planlaması, Meta-Sezgisel Algoritmalar, Gri Kurt Optimizasyon Algoritması.

ABSTRACT

The path planning problem plays an important role in mobile robot work. Robot systems use different techniques to plan the path of the robot from one point to another. The main purpose of the path planning problem is to find the shortest path that a robot does not hit obstacles from the start position to the target position in an environment with obstacles. In this study, the Gray Wolf Optimization algorithm is adapted to the robot path planning problem. The Gray Wolf Optimizer (GWO) mimics the hunting behavior and social leadership of gray wolves in nature. The leadership hierarchy consists of four groups. These are alpha, beta, delta and omega groups. In this algorithm, the hunting mechanism comprises in three stages. These are searching for prey, encircling prey and attacking prey. Three, four, and five circular disabled maps were used in the test work of robot path planning. While there are candidate solutions in the path planning, three coordinate points are used between the start and destination points. At each iteration, these coordinate points are updated to be closer to the solution by the GWO algorithm. If the solution point is in the obstacle area, the value of the infringement is added to the cost function. In this thesis study, the performance of the GWO algorithm is evaluated together with other meta-heuristic algorithms to solve the robot path planning problem. The results obtained with the GWO algorithm show that the most suitable path is used for the test map used.

Keywords: Robot Path Planning, Meta-Heuristic Algorithm, Gray Wolf Optimizer Algorithm.

İÇİNDEKİLER

Sayfa No

JURİ ONAY SAYFASI

TEŞEKKÜRLER

ÖZET..... I

ABSTRACTII

ŞEKİLLER DİZİNİ V

ÇİZELGELER DİZİNİ VII

SİMGELER VE KISALTMALAR VIII

1. GİRİŞ 1

2. SEZGİSEL ALGORİTMALARIN SINIFLANDIRILMASI4

2.1 Farksal Gelişim Algoritması (Differential Evolution-DE)..... 5

2.2 Parçacık Sürü Optimizasyonu Algoritması (Particle Swarm Optimization Algorithm-PSO) 6

2.3 Yapay Arı Koloni Algoritması (Artificial Bee Colony Algorithm-ABC)..... 8

2.4 Ateş Böceği Optimizasyonu Algoritması (Firefly Optimization Algorithm-FA)... 9

2.5 Yabani Ot Optimizasyonu Algoritması (Invasive Weed Optimization Algorithm-IWO) 10

2.6 Emperyalist Rekabetçi Algoritma (Imperialist Competitive Algorithm-ICA) 11

2.7 Karıştırılmış Kurbağa Sıçraması Algoritması (Shuffled Frog Leaping Algorithm-SFLA)..... 12

2.8 Biyocoğrafya Tabanlı Optimizasyon Algoritması (Biogeography Based Optimization Algorithm-BBO) 13

2.9 Kovaryans Matris Adaptasyonu Evrimsel Stratejisi (Covariance Matrix Adaptation Evolution Strategy-CMAES) 14

2.10 Uyum Arama Algoritması (Harmony Search Algorithm-HSA) 14

2.11 Kültür Optimizasyonu Algoritması (Cultural Optimization Algorithm-COA) .. 15

2.12 Yusufçuk Optimizasyonu Algoritması (Dragonfly Optimization Algorithm-DOA)..... 15

2.13 Çekirge Optimizasyonu Algoritması (Grasshopper Optimization Algorithm-GOA).....	16
2.14 Güve-Alev Optimizasyonu Algoritması (Moth-Flame Optimization Algorithm (MFO)	18
2.15 Karınca Aslanı Optimizasyonu Algoritması (Ant Lion Optimization Algorithm-ALO)	19
3. GRİ KURT OPTİMİZASYONU (GREY WOLF OPTİMİZER-GWO).....	21
3.1 Matematiksel Model.....	22
3.2 Sosyal hiyerarşi	24
3.3 Avın Kuşatılması.....	24
3.4 Avlanma	25
4. YOL PLANLAMA PROBLEMİ	27
5. YAPILAN DENEYSEL ÇALIŞMALAR.....	29
5.1 Geliştirilen Grafiksel Arayüzü	33
5.2 Bir Tekrarlı Robot Yol Planlama Sonuçları	34
5.3 Çok Tekrarlı Robot Yol Planlama Sonuçları.....	34
6. SONUÇ.....	38
KAYNAKLAR	39
EK 1: Sezgisel algoritmaların yol planlama problemi-1 için başarımlarının karşılaştırılması (A-P).....	46
EK 2: Sezgisel algoritmaların yol planlama problemi-2 için başarımlarının karşılaştırılması (A-P).....	52
EK 3. Sezgisel algoritmaların yol planlama problemi-3 için başarımlarının karşılaştırılması (A-P).....	58

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 2.1. Sezgisel algoritmaların sınıflandırılması (Canayaz, 2015).....	5
Şekil 2.2. Farksal gelişim algoritması (Price vd., 2005).....	6
Şekil 2.3. Standart parçacık sürüşü optimizasyonu (PSO algorithm, 2018).....	7
Şekil 2.4. Yapay arı koloni algoritması (Kavian, vd., 2013).	8
Şekil 2.5. Ateş böceği algoritması (Ateş Böceği Algoritması, 2014).....	9
Şekil 2.6. Emperyalist rekabet algoritması (Nazari, vd., 2010).....	12
Şekil 2.7. SFLA optimizasyon algoritması (Melia, vd., 2016).....	13
Şekil 2.8. BBO algoritması çalışma prensibi (Rifai, vd., 2018).	14
Şekil 2.9. (a) Gerçek yusufoçuk, (b) yusufoçuğun yaşam döngüsü (Mirjalili, 2016).	16
Şekil 2.10. (a) Gerçek çekirge, (b) Çekirgelerin yaşam döngüsü (Saremi, vd., 2017)..	17
Şekil 2.11. Enine yönelim (Mirjalili, 2015).....	18
Şekil 2.12. Yakın ışık kaynaklarının etrafında spiral uçan yol (Mirjalili, 2015).	19
Şekil 2.13. Koni şeklindeki tuzaklar ve karınca aslanlarının avlanma davranışları (Mirjalili, 2015).....	20
Şekil 3.1. Gri kurt hiyerarşisi (dominantlık yukarıdan aşağıya doğru azalır).....	21
Şekil 3.2. Gri kurtların avlama stratejisi (Mirjalili, 2016; Doğan and Yüzgeç, 2018). .	23
Şekil 3.3. 2D ve 3D uzayında pozisyon vektörleri ve muhtemel bir sonraki konumları (Mirjalili, 2014).....	25
Şekil 3.4. GWO'da konum güncellemesi (Mirjalili, 2014).	26
Şekil 3.5. Gri kurtların avlana davranışları: (A) av peşinde koşmak, yaklaşmak ve izlemek (B – D) takip etmek, taciz etmek ve çevrelemek (E) durağan durum ve saldırı (Mirjalii, 2014).....	26
Şekil 4.1. Bir robot için yol planlaması senaryosu.	28
Şekil 5.1. Çalışmada temel alınan robot yol problemi (Yarpiz, 2015).	29
Şekil 5.2. Yol planlama problemi için kullanılan örnek senaryolar.	30
Şekil 5.3. GWO algoritması ile elde edilen yol planlama problemi (1) için elde edilen tüm çözümler.....	30
Şekil 5.4. Sezgisel algoritmaların robot yol planlama problemi için en iyi maliyet değerlerinin karşılaştırılması.	31

Şekil 5.5. Matlab grafik arayüzü.....	33
Şekil 5.6. Matlab grafik arayüzünde algoritma seçimi.	34



ÇİZELGELER DİZİNİ**Sayfa No**

Çizelge 2.1. Ateş böceklerinin davranışlarının algoritmadaki karşılığı (Ateş Böceği Algoritması, 2014).	10
Çizelge 5.1. Meta-sezgisel algoritma parametreleri.	32
Çizelge 5.2. Yol planlama problemi-1 için sezgisel algoritmaların başarımlarını karşılaştırması.	35
Çizelge 5.3. Yol planlama problemi-2 için sezgisel algoritmaların başarımlarını karşılaştırması.	36
Çizelge 5.4. Yol planlama problemi-3 için sezgisel algoritmaların başarımlarını karşılaştırması.	37

SİMGELER VE KISALTMALAR

Simgeler

$\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$	: Av ve Kurt (alfa, beta, delta) Arasındaki Vektör Mesafesi
$\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$	: Alfa, Beta, Delta Kurtları İçin Avın Pozisyon Vektörü
$\vec{C}_\alpha, \vec{C}_\beta, \vec{C}_\delta, \vec{A}_\alpha, \vec{A}_\beta, \vec{A}_\delta$	: Alfa, Beta, Delta Kurtları İçin Katsayı Vektörleri
$\vec{U}_\alpha, \vec{U}_\beta, \vec{U}_\delta$	: Alfa, Beta, Delta Kurtları İçin Deneme Vektörleri
$\vec{r}_{\alpha 1}, \vec{r}_{\beta 1}, \vec{r}_{\delta 1}$	: [0, 1] 'de İlk Rastgele Vektörler
$\vec{r}_{\alpha 2}, \vec{r}_{\beta 2}, \vec{r}_{\delta 2}$	: [0, 1] 'de İkinci Rastgele Vektörler

Kisaltmalar

GWO	: Gray Wolf Optimization : Gri Kurt Optimizasyonu
DE	: Differential Evolution : Farksal Gelişim
PSO	: Particle Swarm Optimization : Parçacık Sürü Optimizasyonu
ABC	: Artificial Bee Colony : Yapay Arı Kolonisi
FA	: Firefly Algorithm : Ateş Böceği Algoritması
IWO	: Invasive Weed Optimization : Yabani Ot Optimizasyonu
ICA	: Imperialist Competitive Algorithm : Emperyal Rekabetçi Algoritma
SFLA	: Shuffled Frog Leaping Algorithm : Karıştırılmış Sıçramalı Kurbağa Algoritması
BBO	: Biogeography-Based Optimization : Biyocoğrafyaya Dayalı Optimizasyon
CMAES	: Covariance Matrix Adaptation Evolution Strategy : Kovaryans Matris Adaptasyonu Evrimsel Stratejisi
HSA	: Harmony Search Algorithm : Uyum Arama Algoritması
DOA	: Dragonfly Optimization Algorithm : Yusufçuk Optimizasyonu Algoritması
GOA	: Grasshopper Optimization Algorithm : Çekirge Optimizasyonu Algoritması
MFO	: Moth-Flame Optimization : Güve-Alev Optimizasyonu
ALO	: Ant Lion Optimization : Karınca Aslanı Optimizasyonu

1. GİRİŞ

Son otuz yılda, meta-sezgisel algoritmalar optimizasyon problemlerinin çözümünde çok popüler hale gelmiştir. Meta-sezgisel algoritmalar, evrim kuralları, fiziksel denklemler, sürülerin sosyal davranışları ve doğadaki hayvan sürülerinden ilham alır. Meta-sezgisel algoritmalar, fiziksel temelli algoritmalar, evrimsel algoritmalar, sürü zekası algoritmaları, biyocoğrafikilhamlı algoritmalar (Beheshti and Shamsuddin, 2013) ve doğadan ilham alan diğer algoritmalar olarak sınıflandırılabilirler (Blum and Roli, 2010). Fiziksel tabanlı algoritmalarda, optimizasyon probleminin çözümü tek bir çözümle başlar ve her bir yinelemede fiziksel denklemlerle güncellenir. Tabu arama algoritması (Nowicki and Smutnicki, 1996), benzetimli tavlama algoritması (Kirkpatrick, vd., 1983) fiziksel tabanlı algoritmalarla örnek olarak verilebilir. Genetik algoritma ve farksal gelişim algoritması (Storn, vd., 1997; Price, vd., 2005; Yüzgeç, 2010), evrimsel meta-sezgisel algoritmaların en iyi bilinen örnekleridir. Sürü zekası algoritmalarının bazıları, parçacık sürü optimizasyonu algoritması Kennedy ve Eberhart tarafından (Kennedy and Eberhart, 1995), yapay arı kolonisi algoritması Karaboğa tarafından (Karaboğa, 2005), karınca aslanı optimizasyonu algoritması Dorigo ve arkadaşları tarafından (Dorigo, vd., 1996; Dorigo, vd., 2006) ve balık sürüsü algoritması Li ve arkadaşları tarafından ortaya çıkarılmıştır (Li, 2002). Biyo-ilhamlı algoritmalar, biyolojik organizmaların aktivitelerini taklit eder. Bu algoritmaların en önemli örnekleri yapay bağışıklık algoritması (Timmis, vd., 2010) ve bakteriyel toplama optimizasyonu algoritmasıdır (Das, vd., 2009; Passino, vd., 2002; Passino, vd., 2010). Diğer doğadan esinlenen meta-sezgisel algoritmalar guguk kuşu arama algoritması (Yang and Deb, 2014; Yang, 2010), ateş böceği algoritması (Yang, 2007, Yang, 2009), meyve sineği optimizasyonu algoritması (Iscan, Gunduz, 2014), yerçekimi arama algoritması (Rashedi, vd., 2009), emperyalist rekabetçi algoritma (Gargari and Lucas, 2007; Hosseini and Khaled, 2014), karınca aslanı optimizasyonu (Mirjalili, 2015; Kiliç and Yüzgeç, 2017), yusufoçuk optimizasyonu (Mirjalili, 2016; Ranjini and Murugan, 2017) ve balina optimizasyon algoritmasıdır (Mirjalili and Lewis, 2016).

Bu çalışmada, başlangıç noktasından hedef noktasına herhangi bir engelle çarpmadan en uygun yolu bulmak için GWO algoritması önerilmiştir. GWO algoritmasının performansını değerlendirmek için farklı yarıçaplı üç, dört ve beş daire şeklindeki engellerden oluşan haritalar tasarlandı. GWO algoritması, farksal gelişim

algoritması, parçacık sürü optimizasyonu algoritması, yapay arı kolonisi algoritması, ateş böceği optimizasyonu algoritması, yabani ot optimizationu algoritması, emperyal rekabetçi algoritma, karıştırılmış sıçramalı kurbağa optimizasyonu algoritması, biyocoğrafya ilhamlı optimizasyon algoritması, kovaryans matris adaptasyonu evrimsel stratejisi algoritması, uyum arama algoritması, kültür optimizasyonu algoritması, yusufçuk optimizasyonu algoritması, çekirge optimizasyonu algoritması, güve-alev optimizasyonu algoritması, karınca aslanı optimizasyonu algoritması gibi iyi bilinen meta-sezgisel algoritmalar ile karşılaştırılmıştır. Sezgisel algoritmalar, doğruluğu, yüzde yüz olmayan fakat geleneksel yöntemlerden daha hızlı ve daha verimli yöntemlerle problemleri çözmek için tasarlanmış algoritmalarlardır. Sezgisel algoritmalar belirli verilere bağlı kalarak bir problemin çözümüne optimum derecede yaklaşım tarzıdır. Sezgisel algoritmalar, genellikle yaklaşık çözümlerin yeterli olduğu ve kesin çözümlerin maliyetli olduğu durumlarda sıkça kullanılır. Sezgisel algoritma, en basit anlamda bir probleme bağımlı olarak bazı tahminlerde bulunulmasıdır.

Sezgisel (heuristic) kavramının Türkçedeki karşılığı yaklaşık, tahmini, deneye dayalı veya bulucu olarak tanımlanır. Sezgisel optimizasyonlarda bir problem için uygulanan çözüm yönteminin kesinliği garanti edilmez. Burada amaç karmaşık bir problemi çözebilmeyi ve sezgisel algoritmanın yeterli veya tatmin edici bir sonucu bulabilmesidir. Sezgisel algoritmalarda iki durumdan biri seçilir. Yani ya problem için hızlı bir çözüm üretilmeli fakat bunun problemi her zaman çözeceği garanti edilemez ya da problemi uygun bir sürede çözer fakat her zaman aynı hızda çözeceği garanti edilemez. Sezgisel terim, mümkün olan her şey arasında çözümler bulabilen algoritmalar için kullanılır, ancak bunların en iyi bulunacağını garanti etmezler, bu nedenle, yaklaşık olarak doğru algoritmalar olarak kabul edilebilirler. Bu algoritmalar, genellikle en iyiye yakın bir çözüm bulurlar. Sezgisel algoritmalar her gün tecrübe ettiğimiz yaklaşımlardır. Örnek vermek gerekirse, bir yere giderken yön sezgimize dayanarak, yolun bizi nereye çıkaracağını bilmeden hareket etmek ve yol ayrımlarında sezgisel olarak bir seçim yapmak böyle bir yaklaşımdır.

Sezgisel optimizasyon teknikleri son yıllarda çok daha da önem kazanmış genetik algoritma, karınca kolonisi optimizasyonu ve parçacık sürü optimizasyonu gibi bazı algoritmalar, sadece bilgisayar bilimcileri değil, aynı zamanda diğer bilim adamları tarafından da bilinir hale gelmiştir. Çok sayıda teorik çalışmanın yanı sıra, çeşitli

alışma alanlarında bu optimizasyon teknikleri uygulanmıřtır. Sezgisel algoritmaların sade, esnek vb. özellikleri neden dikkate değeri ölçüde yaygın hale geldiğini göstermektedir.

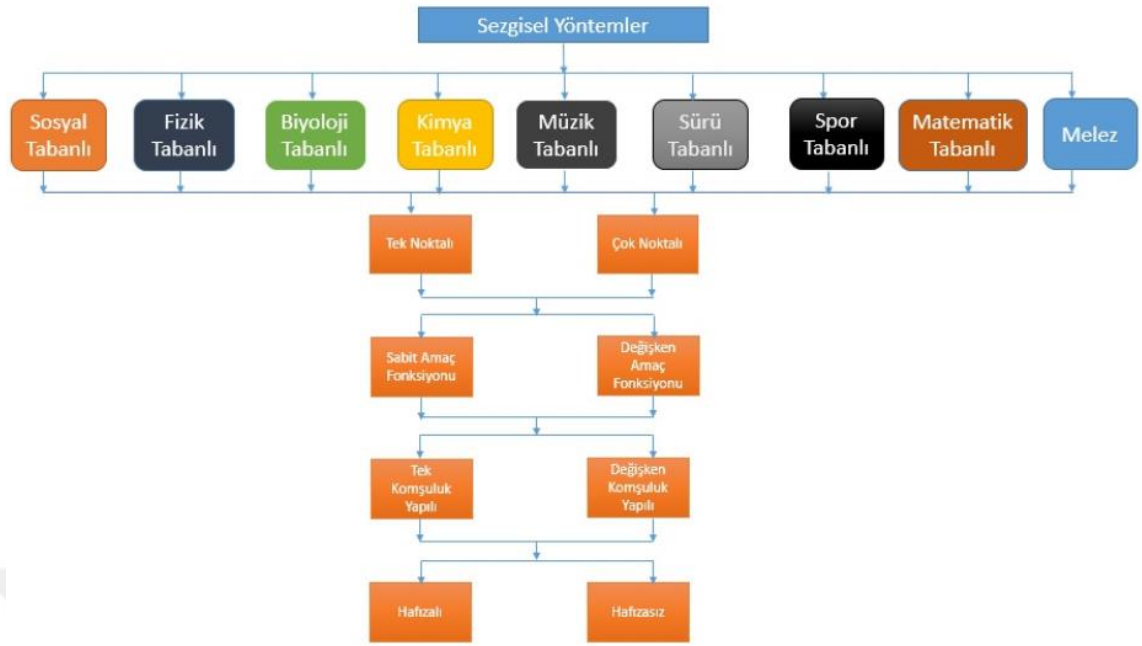
Optimizasyon süreci rastgele çözümler ile başlar ve optimum çözümler bulmak için arama alanlarının türevini hesaplamayı gerektirmez. Bu da sezgisel algoritmanın gerçek problemlere oldukça uygun hale getirir. Gerçek sorunların arama alanı genellikle çok karmaşıktır, bu yüzden üst-sezgisel bu zorlu gerçek problemleri optimize etmek için iyi bir seçenektir. Genel anlamda, meta-sezgisel tek çözüm tabanlı ve popölasyon tabanlı olmak üzere iki ana sınıfa ayrılabilir. Örneğın, benzetimli tavlama (simulated annealing) arama süreci bir aday çözümle başlar. Bu tek aday çözüm daha sonra tekrarlamalar boyunca geliştirilir.

2. SEZGİSEL ALGORİTMALARIN SINIFLANDIRILMASI

Sezgisel algoritmalar, herhangi bir amacı gerçekleştirmek veya hedefe varmak için doğal olaylardan esinlenen algoritmalarlardır. Bu algoritmaların, çözüm uzayında optimum çözüme yakınsaması garanti edilememektedir. Yani sezgisel algoritmalar yakınsama özelliğine sahiptirler ama kesin çözümü garanti etmezler.

Optimizasyon, en iyileme anlamına gelmektedir. Bir problem için, verilen şartlar altında tüm çözümler arasından en iyi çözümü elde etme işidir. Belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilir. Bazen tek başlarına hiçbir iş yapamayan varlıklar, toplu hareket ettiklerinde çok zekice davranışlar sergileyebilmektedirler. Genel amaçlı sezgisel yöntemler; biyoloji tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı ve kimya tabanlı olmak üzere altı farklı grupta değerlendirilmektedir. Ayrıca bunların birleşimi olan melez yöntemler de vardır. Bahsedilen bu yöntemler Şekil 2.1’de gösterilmektedir (Canayaz, 2015).

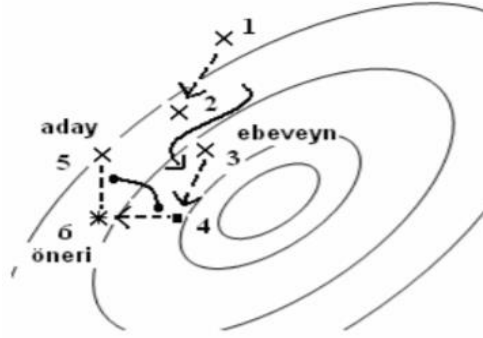
Genetik algoritma, diferansiyel gelişim algoritması, karınca koloni algoritması, yapay sinir ağları, arı koloni algoritmaları ve yapay bağışıklık sistemleri biyolojik tabanlı; emperyalist rekabetçi algoritma, parlamenter optimizasyon algoritması ve tabu arama sosyal tabanlı; yapay kimyasal reaksiyon algoritması kimya tabanlı; uyum arama algoritması müzik tabanlı; ısıtma işlemi, büyük patlama büyük sıçrama, yerçekimsel arama algoritması, merkez kuvvet optimizasyonu, zeki su damlacıkları algoritması ve elektromanyetizma algoritması fizik tabanlı ve parçacık sürü optimizasyonu, sürü tabanlı algoritma ve modellerdir. Kültürel algoritma hem biyoloji hem de sosyal tabanlı algoritma olduğundan melez olarak sınıflandırılabilir.



Şekil 2.1. Sezgisel algoritmaların sınıflandırılması (Canayaz, 2015).

2.1 Farksal Gelişim Algoritması (Differential Evolution-DE)

Farksal gelişim algoritması 1995'te Price ve Storn tarafından ortaya çıkarılan özellikle sürekli verilerin söz konusu olduğu problemlerde etkin sonuçlar verebilen, işleyiş ve operatörleri itibariyle genetik algoritmaya dayanan popülasyon temelli sezgisel optimizasyon tekniğidir. Günümüzde gerçek sayı fonksiyonu optimizasyonu için en güçlü evrimsel algoritmalarından biri olarak kabul edilir. Farksal gelişim algoritmasını kullanmanın mühendislik, istatistik ve finans gibi alanlardaki kısıtlamalardan kaynaklı, analitik olarak çözülmesi mümkün olmayan problemlere yaklaşık olarak çözüm bulması, basit yapısı, kullanım kolaylığı, hız ve sağlamlığı gibi avantajları vardır. Şekil 2.2.'de farksal gelişim algoritmasının çalışma mantığı gösterilmektedir. DE algoritması aynı anda birçok noktada araştırma yapabilmektedir. DE'nin diğer sezgisellere göre önemli bir üstünlüğü de kolayca kodlanabilir olmasıdır (Storn and Price, 1997; Price vd., 2005).



Şekil 2.2. Farksal gelişim algoritması (Price vd., 2005).

2.2 Parçacık Sürü Optimizasyonu Algoritması (Particle Swarm Optimization Algorithm-PSO)

Parçacık sürü optimizasyonu algoritması 1995–1996 yıllarında sosyolog-psikolog James Kennedy ve Elektrik Mühendisi Russel Eberhart tarafından ortaya çıkarılan ve ilk olarak kuş ve balık sürülerinin hareketlerinden esinlenerek doğrusal olmayan nümerik problemlere optimum sonuçlar bulmak için geliştirilmiş popülasyon tabanlı bir optimizasyon yöntemi olup çok parametrelili ve çok değişkenli optimizasyon problemlerine çözümler üretmek için kullanılmaktadır. Başlangıçta amaç, kuş ya da balık sürü koreografisinin grafiksel olarak benzetimlerini yapmaktır. Ancak grafiksel benzetimden sonra, parçacık sürü modelinin bir optimizasyon yöntemi olarak kullanılabileceği keşfedilmiştir. Kuş toplulukları gerçek yiyecek kaynağını bilmemelerine rağmen, yiyecek kaynağından ne kadar uzakta olduklarını öğrenmeye çalışırlar. Bunun için yiyecek kaynağına en yakın olan kuşu izlerler. PSO’da her bir kuş parçacık olarak, kuş topluluğu da sürü olarak temsil edilir. Parçacık hareket ettiğinde, kendi koordinatlarının uygunluk değeri yani yiyeceğe ne kadar uzaklıkta olduğu hesaplanır. Sistem rastgele çözümlerden oluşan bir popülasyonla başlatılır ve en iyi çözüm için jenerasyonları güncelleyerek arama yapar.

Burada birtakım yinelemeler ile bir değişken grubunun değeri herhangi bir anda hedefe en yakın olan üyeye daha yakın ayarlanır. Gizli bir besin kaynağının kokusunu alabilecekleri bir alanın etrafında dönen bir kuş sürüsünü ele alalım. Yemeklere en yakın olanı en yüksek ses çıkarır. Diğer kuşlardan herhangi biri, ilkinden daha çok hedefe yaklaşırsa, daha yüksek sesler çıkarır ve diğerleri ona doğru yönelir. Bu

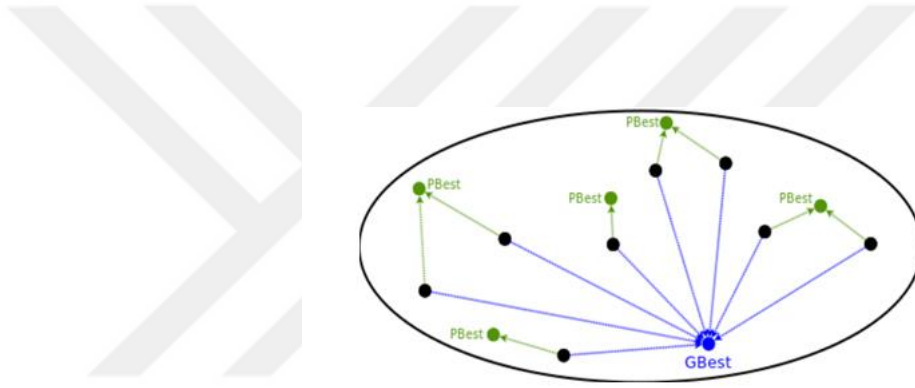
sıkılaştıran örüntü, kuşlardan biri yiyeceğin başına gelene kadar devam eder (Kennedy and Eberhart, 1995; Clerc and Kennedy, 2002).

Algoritma üç genel değişkenin kaydını tutar:

- Hedef değer veya koşul,
- Hangi partikülün verilerinin hedefe en yakın olduğunu gösteren global en iyi (gBest) değeri,
- Hedef bulunamazsa algoritmanın ne zaman duracağını gösteren değer.

Her parçacık şunlardan oluşur:

- Olası bir çözümü temsil eden veriler,
- Verilerin ne kadar değiştirilebileceğini gösteren bir hız değeri,
- Parçacık verilerine en yakın olan en iyi (pBest) değer.



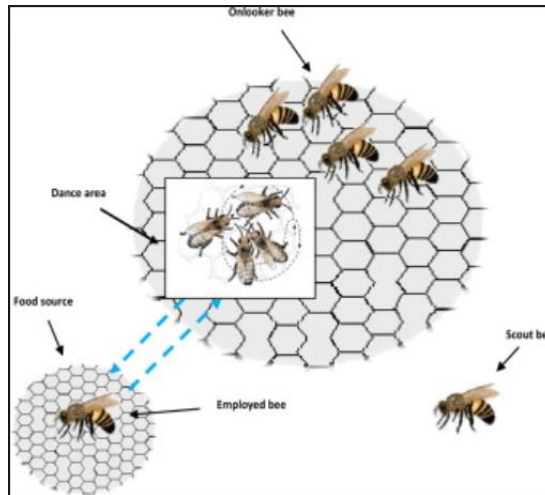
Şekil 2.3. Standart parçacık sürüşü optimizasyonu (PSO algorithm, 2018).

Parçacıkların verileri herhangi bir şey olabilir. Şekil 2.3'teki örnekte, veriler her kuşun X, Y, Z koordinatları olacaktır. Her kuşun kendi koordinatları, en yakın kuşun koordinatlarına (gBest) daha yakın olacak şekilde yaklaşmaya çalışacaktır. Kuşlar örneğinde, yiyeceklerden en uzak olan bireyler, gBest kuşuna doğru daha hızlı uçarak diğerlerine ayak uydurmak için çaba harcarlar. Her parçacığın pBest değeri, algoritmanın başlatılmasından bu yana verinin hedefe en yakın olduğu koordinatı gösterir. gBest değeri herhangi bir parçacığın pBest değerinin gBest'ten daha fazla hedefe yaklaştığı durumda değişir. Algoritmanın her yinelemesi ile gBest, parçacıklardan biri hedefe ulaşana kadar yavaş yavaş hedefe yaklaşır (Mnemstudio, 2012).

2.3 Yapay Arı Koloni Algoritması (Artificial Bee Colony Algorithm-ABC)

Yapay arı koloni algoritması (ABC), 2005 yılında Derviş Karaboğa tarafından ortaya atılan bal arılarının davranışlarını taklitle dayanan bir meta-sezgisel optimizasyon algoritmasıdır (Karaboğa, 2005). Yapay arı koloni algoritması, bal arısı sürülerinin kendilerine özgü davranışlarını örnek alarak, arıların besin ararken kullandıkları yöntemlerden esinlenerek oluşturulmuş bir algoritmadır. Yapay arı koloni algoritmasındaki arıların yiyecek arama davranışı, öğrenme, bilgi paylaşımı gibi özellikler, son zamanlarda sürü zekâsındaki en ilginç araştırma konularından birisi olmuştur. Bal arıları üzerinde çalışmalar son yıllarda literatürde bir artış içindedir. Doğal arı kolonisinde arılar arasında yapılacak işlere göre bir görev paylaşımı vardır. Bu görev paylaşımı Şekil 2.4'te gösterilmektedir. Arılar bu iş paylaşımını merkezi bir birim olmadan, kendi kendilerine gerçekleştirmektedirler. Arılar, polen veya nektar bulmak için yiyecek kaynaklarına gitmektedirler. Yiyecek kaynağının değeri, yuvaya yakınlığı, çeşidi, nektar yoğunluğu, nektarın çıkarılmasının kolaylığı gibi birçok etkene bağlıdır (Karaboğa, 2005; Karaboğa and Akay, 2009).

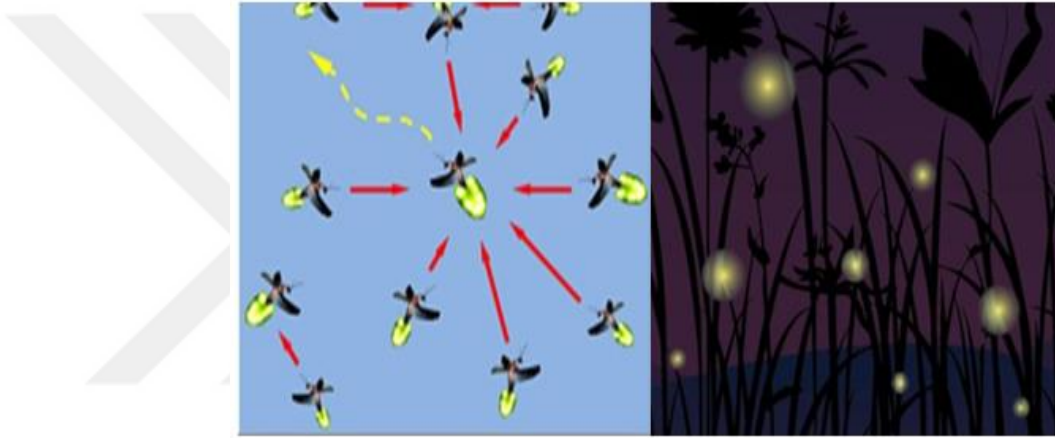
Ortaklaşa bilginin oluşumundaki en önemli etmen arılar arasındaki bilgi paylaşımıdır. Yiyecek kaynağının yeri ve kalitesi hakkındaki bilgi paylaşımı kovandaki dans ile yapılmaktadır. Dans eden arıya diğer arılar antenleri aracılığıyla dokunarak kaynağın tadı ve kokusu hakkında da bilgi alırlar (Karaboğa, 2005). Şekil 2.4'de yapay arı koloni algoritması gösterilmektedir.



Şekil 2.4. Yapay arı koloni algoritması (Kavian, vd., 2013).

2.4 Ateş Böceği Optimizasyonu Algoritması (Firefly Optimization Algorithm-FA)

Ateş Böceği algoritması (FA) 2008 Yılında Xin-She Yang tarafından geliştirilmiş ateş böceklerinin yanıp sönme davranışından esinlenen sürü tabanlı bir optimizasyon algoritmasıdır. Algoritma, ateş böceklerini cinsiyet farkı gözetmeksizin ele alır. Tüm ateş böcekleri birbirlerine yönelebilirler. Daha parlak olan ateş böcekleri daha çekicidir. Daha az parlak olan ateş böcekleri çekici olan diğer ateş böceklerine doğru yönelirler. Parlaklık etkisi uzaklık arttıkça azalacağı için daha uzaktaki ateş böcekleri, uzaktaki parlak ateş böceklerinden daha az etkilenirler. Eğer ateş böceği kendinden daha parlak bir ateş böceği bulamazsa rastgele bir hareket gerçekleştirir (Yang, 2007; Yang, 2009). Şekil 2.5'de Ateş böceği algoritması gösterilmektedir.



Şekil 2.5. Ateş böceği algoritması (Ateş Böceği Algoritması, 2014).

Ateş böceği algoritması, tropikal iklim bölgelerindeki ateş böceklerinin sosyal davranışlarını baz alır ve diğer sürü zekâ tabanlı algoritmalarla bir çok benzerliği bulunmasına rağmen kavram ve uygulamada daha basittir. Bir ateş böceğinin ışıklarını yakıp söndürmesinin birincil amacı, diğer ateş böceklerini kendine doğru çekmektir. Yanıp sönen ışıkların üretimindeki karmaşık biyokimyasal sürecin detayları ve gerçek amacı bilim dünyasında hâlâ bir tartışma konusu olmasına rağmen, birçok araştırmacı yanıp sönen ışıkların ateşböceğine, arkadaşlarını bulmada, olası avlarını çekmede ve avcılarından kendilerini korumada yardımcı olduğuna inanmaktadır. Tablo 2.1'de ateş böceklerinin davranışlarının algoritmadaki karşılığı gösterilmektedir. Ateşböceği algoritmasında, verimli optimum çözümler elde etmek için, verilen bir optimizasyon probleminin amaç fonksiyonu, ateşböceği sürüsüne parlak ve daha çekici yerlere gitmede yardımcı yanıp sönen ışık ya da ışık şiddeti ile ilişkili olmaktadır. Bütün ateş

böcekleri tek cins olarak kabul edilmektedir ve birbirilerini çekmeleri bu algoritmanın temelini oluşturmaktadır. Bir ateş böceği ne kadar parlak olursa diğer ateş böcekleri için o kadar çekici hale gelmektedir. Kendisinden daha parlak bir ateşböceği gördüğünde ona doğru gidecektir (Yang, 2009).

Çizelge 2.1. Ateş böceklerinin davranışlarının algoritmadaki karşılığı (Ateş Böceği Algoritması, 2014).

Esinlenen davranışlar	Algoritmadaki karşılığı
Ateş böceklerinin rastgele konumları belirlenir. Daha sonra parlaklıkları hesaplanır.	Rastgele çözüm kümeleri oluştur. Her çözüm kümesi parametre sayısı kadar eleman içerir. Oluşturulan çözüm kümelerinin uygunluk değerleri bulunur.
Ateş böcekleri daha parlak ateş böceklerine doğru hareket eder.	Tüm çözüm kümelerine sırayla iyileştirme formülü uygulanır.

2.5 Yabani Ot Optimizasyonu Algoritması (Invasive Weed Optimization)

Algorithm-IWO)

Yabani Ot Optimizasyonu (IWO), 2006 yılında Alireza Mehrabian ve Caro Lucas tarafından önerilen yabani otların yayılma stratejisinden ve doğadan esinlenen bir sezgisel algoritmadır (Mehrabian, Lucas, 2006).

IWO algoritmasının işlem akışı aşağıda belirtilmiştir:

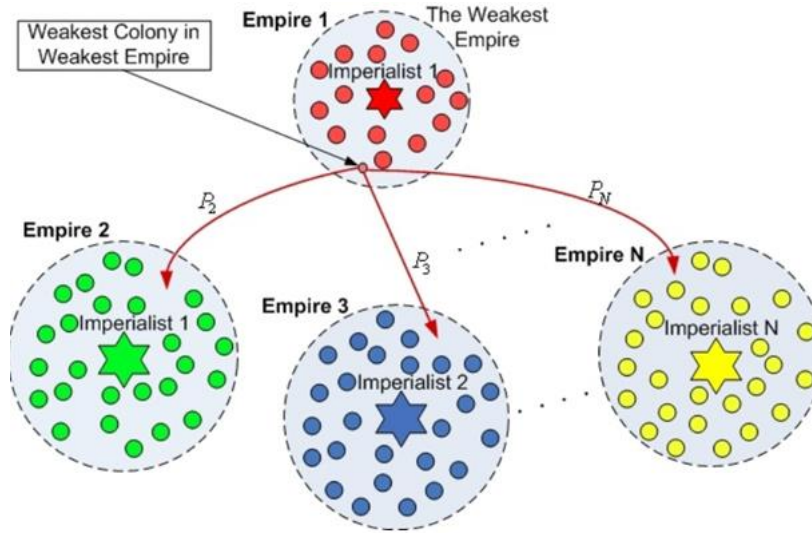
1. İlk olarak başlangıç tohumları rastgele dağıtılır. Sonuç olarak, her tohum n-D çözelti alanında her değişken için rastgele değerler içerir. Burada n, arama uzayında seçilen değişkenlerin sayısıdır.
2. Her bir tohumun uygunluğu optimizasyon problemine göre hesaplanır ve tohumlar yeni üniteler üretebilecek yabani otlara dönüşür.
3. Her birey diğer yabani otlara göre kendi uygunluğuna göre sıralanır. Daha sonra, her yabancı ot popülasyonuna göre yeni tohumlar üretir. Daha fazla kaynak edinmiş olan yabani otların tohum üretme şansı daha yüksektir ve tarlaya daha az adapte olanların çoğalması muhtemel değildir ve böylece daha az tohum

oluştururlar. Yani, her ot tarafından üretilecek tohum sayısı, minimumdan maksimuma doğru değişir.

4. Üretilen tohumlar normal olarak alan üzerinde sıfır ortalama ve tarif edilen değişken standart sapma ile dağıtılır.
5. Her tohumun en uygun değeri (fitness) ebeveynleriyle birlikte hesaplanır ve tüm popülasyon sıralanır. Daha az zindeliğe sahip yabancı otlar rekabet yoluyla elimine edilir ve azami yabancı ot popülasyonuna eşit olan sadece bir miktar yabancı ot kalır.
6. Kullanıcının ulaştığı izin verilen maksimum iterasyon sayısına kadar adım 2 tekrarlanır (Mehrabian, 2006).

2.6 Emperyalist Rekabetçi Algoritma (Imperialist Competitive Algorithm-ICA)

Emperyalist Rekabetçi Algoritma (ICA), ilk olarak sürekli optimizasyon problemleriyle başa çıkmak için 2007 yılında ortaya sürülmüş bir sezgisel algoritmadır. Ancak son zamanlarda farklı optimizasyon problemlerini çözmek için yoğun bir şekilde uygulanmaktadır. ICA diğer algoritmalarından farklı olarak doğadan gelen fenomenlere dayanmayan yeni bir optimizasyon algoritmasıdır. İnsan sosyal evrimi alanından elde edilen ICA, sürü zekası teorisinin bir bileşenidir. ICA, emperyalist rekabetin esinlendiği yeni bir evrimsel optimizasyon yöntemidir. Başlangıcı, emperyalist ülke olarak adlandırılan diğer evrimsel algoritmalar gibi başlar. Emperyalist rekabet, önerilen algoritmanın ana parçasıdır ve kolonilerin minimum maliyet fonksiyonuna yaklaşmasına neden olur. Bu yöntem, farklı optimizasyon görevleriyle uğraşmaya yeni başlamış bir sosyo-politik arama stratejisidir (Gargari and Lucas, 2007; Hosseini and Khaled, 2014). Şekil 2.6'da emperyalist rekabet algoritması gösterilmiştir.

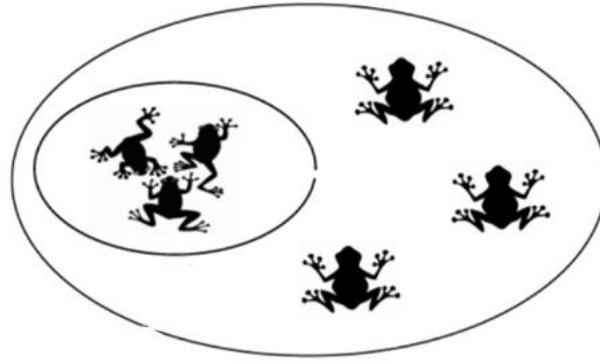


Şekil 2.6. Emperyalist rekabet algoritması (Nazari, vd., 2010).

ICA, emperyalizm ve emperyalist rekabetin sosyal politik sürecini gösterir. Bu algoritma, popülasyondaki her bir bireyin bir ülke olarak adlandırıldığı nüfus tabanlı bir süreçtir. En iyi ülkelerden bazıları emperyalist devletler olarak seçilir ve kalan ülkeler bu emperyalist devletlerin kolonilerini oluştururlar. İlk ülkelerin tümü kolonileri kendi gücüne dayanarak bölüşürler. Her ülkenin gücü, maliyet değeri ile ters orantılıdır (Nazari, vd., 2010).

2.7 Karıştırılmış Kurbağa Sıçraması Algoritması (Shuffled Frog Leaping Algorithm-SFLA)

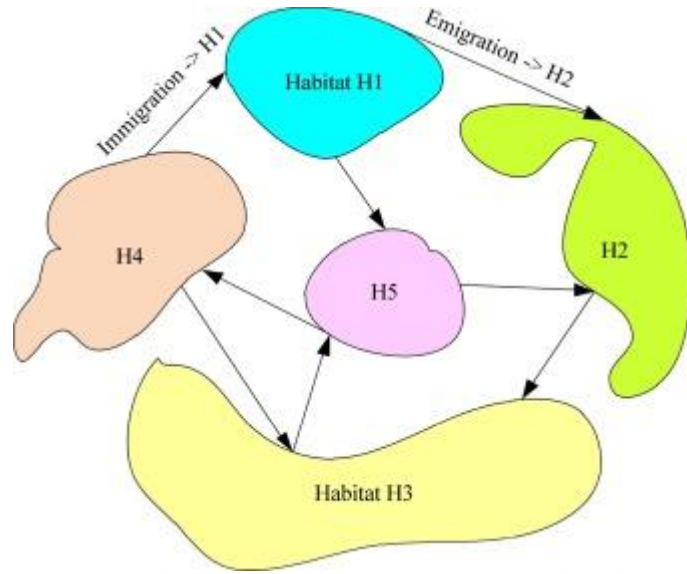
Karıştırılmış Kurbağa Sıçraması Algoritması (SFLA), 2003 yılında Eusuff ve Lansey tarafından ortaya çıkarılmış bir sezgisel algoritmadır. Bir grup kurbağanın, mümkün olan en fazla miktarda gıdaya sahip olan yeri ararken, kurbağaların davranışını gözlemlemeyi, taklit etmeyi ve modellemeyi temel alan popüler bir sezgisel algoritmadır. SFLA'da nüfus bir gruptan ya da alt kümelerle bölünen kurbağalardan oluşur. SFLA işyeri planlaması, gezgin satıcı problemi, çeşitli mühendislik optimizasyon problemleri gibi birçok karmaşık optimizasyon problemlerinin çözümünde başarılı bir şekilde uygulanmıştır. SFLA algoritmasının en belirgin faydası hızlı yakınsamasıdır. SFLA algoritması, popülasyona dayalı bir rastgele arama algoritmasıdır (Alireza & Mirzaei, 20017; Melia, vd., 2016). Şekil 2.7'de SFLA optimizasyon algoritması gösterilmiştir.



Şekil 2.7. SFLA optimizasyon algoritması (Melia, vd., 2016).

2.8 Biyocoğrafya Tabanlı Optimizasyon Algoritması (Biogeography Based Optimization Algorithm-BBO)

Biyocoğrafya Tabanlı Optimizasyon Algoritması (BBO), ilk olarak 2008'de Dan Simon tarafından önerilmiş, biyocoğrafik kavramlardan esinlenen sezgisel bir algoritmadır. Bu algoritma, habitatlar arasındaki türlerin göçünü, uygun olmayan yaşam alanlarından uygun yaşam alanlarına göç şeklinde tanımlayan matematiksel bir modele dayanmaktadır. Bu matematiksel modelde habitat uygunluk indeksi ve tanımı olarak habitatların uygunluğu hesaplanır ve saklanır. Bu algoritma ile ilgili mevcut araştırma ve analizlere göre, sayısal optimizasyon problemlerinde büyük bir başarı elde etmekle kalmamış, aynı zamanda çeşitli uygulamalarda da uygulanmış ve dünya çapında dikkat çekmiştir. İlk olarak, göç ve mutasyon da dahil olmak üzere BBO'nun temel operatörlerine bakılırsa göç operatörü için, adacıklar arasında tür göçünü taklit eder, bu da aday çözümlerinin birbirleriyle etkileşime girmesi için bir rekombinasyon yolu sağlar ve böylece tüm popülasyon geliştirilebilir. Şekil 2.8'de BBO algoritmasının çalışma prensibi gösterilmiştir.



Şekil 2.8. BBO algoritması çalışma prensibi (Rifai, vd., 2018).

2.9 Kovaryans Matris Adaptasyonu Evrimsel Stratejisi (Covariance Matrix Adaptation Evolution Strategy-CMAES)

Kovaryans Matris Adaptasyonu Evrimsel Stratejisi (CMA-ES), 2004 yılında Nikolaus Hansen tarafından sürekli fonksiyonlar için ortaya atılmış bir optimizasyon algoritmasıdır. CMA-ES, bir ortalama vektörü m , bir $n \times n$ kovaryans matrisi C ve bir adım büyüklüğü σ 'ya bağlı olarak çok değişkenli bir normal dağılımdan numune alma çözümleri ile çalışır. Arama, daha iyi çözümler elde etmek için bu parametreleri iteratif olarak uyarlayarak ilerler. Bunu başarmak için üretilen çözümler kalitelerine göre sıralanır. Daha sonra, bu sıralı çözümler, gelecek nesil için dağıtım parametrelerini güncellemek için ağırlıklı denklemlerde kullanılır.

CMA-ES genellikle kısıtlanmamış veya sınırlanmış optimizasyon problemlerine uygulanır. CMA-ES, yinelemeli bir prosedür içinde pozitif bir kesin matrisi tahmin eden ikinci mertebeden bir yaklaşımdır.

2.10 Uyum Arama Algoritması (Harmony Search Algorithm-HSA)

(HSA), 2000 yıllarında Al-Betar, Geem ve arkadaşları tarafından su dağıtım ağı problemini çözmede kullanılmıştır. HSA müzisyenlerin doğaçlama bir uyumu bulmak için doğaçlama sürecini taklit etmeye çalışan bir meta-sezgisel arama algoritmasıdır. Son yıllarda, bazı avantajlar nedeniyle, HSA önemli bir ilgi gördü. HSA'nın uygulanması kolaydır ve en uygun çözüme çabucak ulaşarak makul bir hesaplama

süresinde yeterince iyi bir çözüm bulur. HSA algoritmasının esası, farklı mühendislik alanlarının optimizasyon problemlerine uygulanmasına yol açmıştır. Literatürde HSA'nın zor optimizasyon problemlerini çözmede iyi bir performans gösterdiği ve bu algoritmanın farklı versiyonlarının geliştirildiği gözlenmektedir.

HSA müzik doğaçlama sürecinden esinlenen meta-sezgisel bir algoritmadır. Müzik doğaçlama sürecinde, önceden belirlenmiş sayıda müzisyen, hoş bir uyum (en iyi durum) elde etmek için enstrümanlarının sesini ayarlamaya çalışır. Doğada, bir uyum, farklı frekanslara sahip olan çeşitli ses dalgaları arasındaki özel bir ilişki ile tanımlanır. Estetik kestirimi iyileştirmek ve en iyi uyumu bulmak için müzisyenler pratikten sonra tekrar pratik yaparlar. Müzisyenlerin doğaçlaması ile optimizasyon süreçleri arasında benzerlikler vardır. Bir optimizasyon probleminde, nihai amaç, önceden tanımlanmış bir sayıda karar değişkenini ayarlayarak, söz konusu objektif fonksiyonun en iyi değerini bulmaktır. Daha sonra karar değişkenlerinin değerleri objektif fonksiyona getirilir ve çözüm vektörünün kalitesi hesaplanır. Çözüm vektörü, optimum değer elde edilene kadar yineleme sırasında güncellenir. (Al-Betar, vd., 2013).

2.11 Kültür Optimizasyonu Algoritması (Cultural Optimization Algorithm-COA)

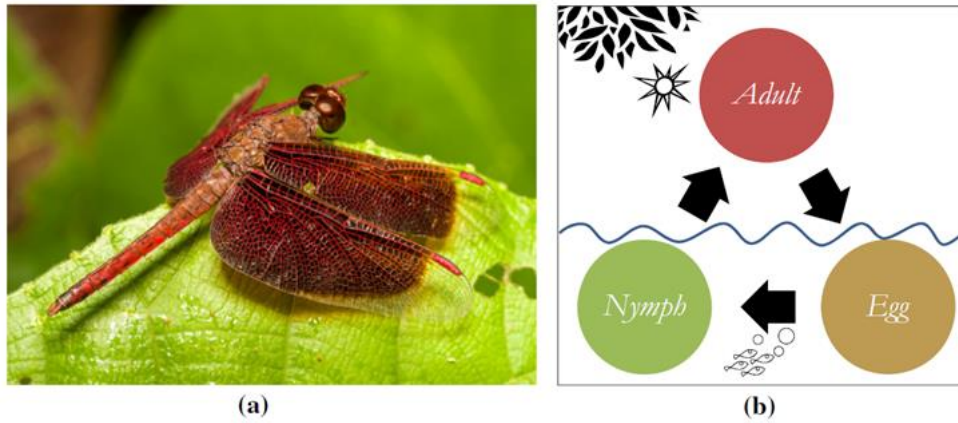
Steward'ın sosyo-kültürel entegrasyon teorisini sistem fikirleri olarak kullanan bir kültürel evrim algoritmasıdır. Reynolds'un kültürel algoritmasından en önemli farkı, COA'nın sosyo-kültürel geçiş olgusuna uymayı amaçlaması, farklı kültürel nüfus evrimi kavramını taklit etmesi ve kültürel türler arasında iletişim, öğrenme gibi davranışları içermesidir. Bu sistemi bilgi arama şemasına taşıyan iki mekanizma vardır. Bunlar, bir popülasyondaki bir sorunu ve nüfusun yenilik modunu temsil eden bir bireyi kodlamanın veya tanımlamanın yoludur. Kültürel türlerin evrim modları grup konsensüsü, bireysel öğrenme, yenilikçi öğrenme ve kendini geliştirme olarak dört tür olarak kategorize edilebilir. Kültürel nüfus grup konsensüsü, kültürel popülasyonun tüm katmanlarının temelinde bulunmaktadır. Düşük rekabet gücüne sahip kültürel türler bu özellikleri öğrenip yeni kültürel türlere dönüşebilir (Kuo, 2013).

2.12 Yusufçuk Optimizasyonu Algoritması (Dragonfly Optimization Algorithm-DOA)

Yusufçuklar süslü böceklerdir. Dünyada bu böceklerin yaklaşık 3000 farklı türü vardır. Şekil 2.9'da görüldüğü gibi, bir yusufçuğun yaşam döngüsü iki ana kilometre taşı

içerir. Yaşam sürelerinin büyük bir kısmını larva içinde geçirirler ve yetişkin olmak için metamorfizme uğrarlar. Yusufçuklar, doğada neredeyse tüm diğer küçük böcekleri avlayan küçük yırtıcı hayvanlardır. Larva dönemi yusufçukları diğer deniz böcekleri ve hatta küçük balıklar için de ön ayaktır. Yusufçuklar hakkında ilginç gerçek onların eşsiz ve nadir karışma davranışlarıdır. Yusufçuklar sadece iki nedenden ötürü avlanırlar. Birincisi statik (besleme) sürü olarak adlandırılır ve ikincisi dinamik (göçmen) sürü olarak adlandırılır. Statik sürüler halinde, yusufçuklar küçük gruplar oluştururlar ve küçük bir alan üzerinde uçarlar ve kelebekler, sivrisinekler gibi diğer uçan avları avlarlar. Yerel hareketler ve yoldaki ani değişimler, statik bir sürünün temel özellikleridir. Bununla birlikte, dinamik sürülerde, çok sayıda yusufçuk, sürüyü uzun mesafeler boyunca bir yönde göç ettirmek için bunu yapar.

DOA algoritmasının temel ilham kaynağı statik ve dinamik kaynaştırma davranışlarıdır. Bu iki kaynaştırma davranışı, meta-sezgiselliği kullanarak keşif ve sömürünün iki ana aşamasına çok benzemektedir. Yusufçuklar, keşif evresinin ana hedefi olan alt sürüleri yaratır ve statik bir sürünün içinde farklı alanlarda uçarlar. Ancak, statik sürüde, yusufçuklar büyük sürüler halinde ve sömürü evresinde uygun olan tek yön boyunca uçarlar (Mirjalili, 2016).

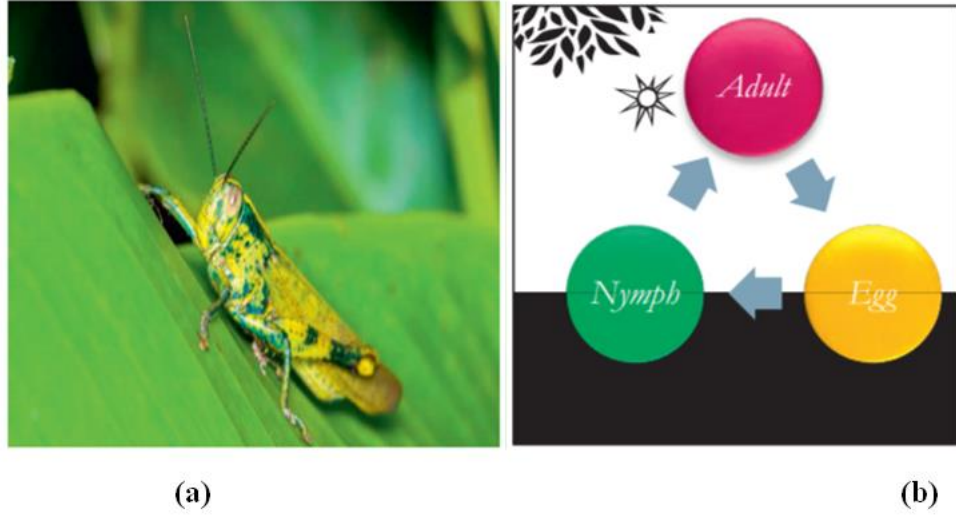


Şekil 2.9. (a) Gerçek yusufçuk, (b) yusufçuğun yaşam döngüsü (Mirjalili, 2016).

2.13 Çekirge Optimizasyonu Algoritması (Grasshopper Optimization Algorithm-GOA)

Çekirgeler bir tür böceklerdir ve zararlı olarak kabul edilirler. Çoğunlukla ekin üretimine ve ayrıca zararlı olduklarından tarıma da zarar verirler. Genellikle çekirgeyi doğada ayrı ayrı görürüz, fakat çoğu zaman doğada tüm canlılar arasında büyük sürüngenlere de katılırlar. Çekirge sürüsü, sürünün büyüklüğü dikkate alındığında

çiftçiler için bir kabus olabilir. Çekirge sürüsü, benzersiz bir karakteristiğe sahiptir ki bu da larvadaki sürü davranışının yanı sıra çekirgede yetişkinliği de gösterir. Larva döneminde çekirge, milyonlarca sayıyla silindirler gibi hareket eder. Neredeyse hareketleri sırasında yollarına çıkan tüm bitki örtüsünü yerler. Onlar larvadan yetişkin olduklarında, havada sürüler oluştururlar ve çok büyük bir mesafeye göç ederler. Şekil 2.10'da gerçek çekirge ve çekirgelerin yaşam döngüsü gösterilmiştir.



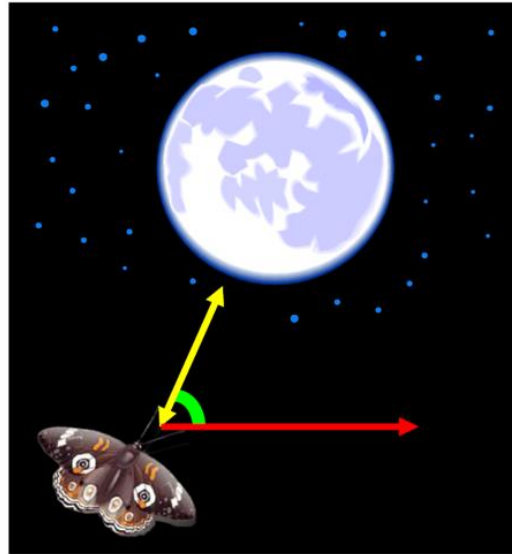
Şekil 2.10. (a) Gerçek çekirge, (b) Çekirgelerin yaşam döngüsü (Saremi, vd., 2017).

Sürüler larva fazında olduklarında genellikle çok yavaş hareket ederler. Çekirgenin adım küçüklüğü, larva fazdaki sürünün temel özelliğidir. Bunun tersine yetişkinlikte sürünün ana özelliği uzun menzilli olması ve sürünün kısa olmasıdır. Çekirge sürüsü, esas olarak gıda kaynağının aranması için oluşturulmaktadır. Çekirgenin yiyecek arayışı, çekirgelerin kaynaşmasının bir başka özelliğidir. Doğadan ilham alan bir optimizasyon algoritması tasarlamak için bu çekirge davranışından bir matematiksel model oluşturulmuştur. Çekirgelerin kaynaştırma davranışını benzetmek için matematiksel modeller kullanılır. Matematiksel modeller, arama alanını keşfedebilir ve kullanabilir. Arama etmenleri için keşif seviyesini ayarlamak için bir mekanizma gereklidir. Doğada, çekirgenin ilk hareketi, kanatları olmadığı larva fazında oldukları yiyecek aramak içindir. Daha geniş bir bölge araştırmasında yetişkinlik dönemlerinde keşfetmek için havada özgürce hareket ederler. Ancak arama alanında, uygun bölgeyi bulma ihtiyacı olduğu için, keşif önce rastsal optimizasyon

algoritmasından gelir. Uygun bölgeyi keşif ile elde ettikten sonra, aramanın optimuma doğru yaklaşımını bulmak için keşif yoluyla taşınır (Saremi, vd., 2017).

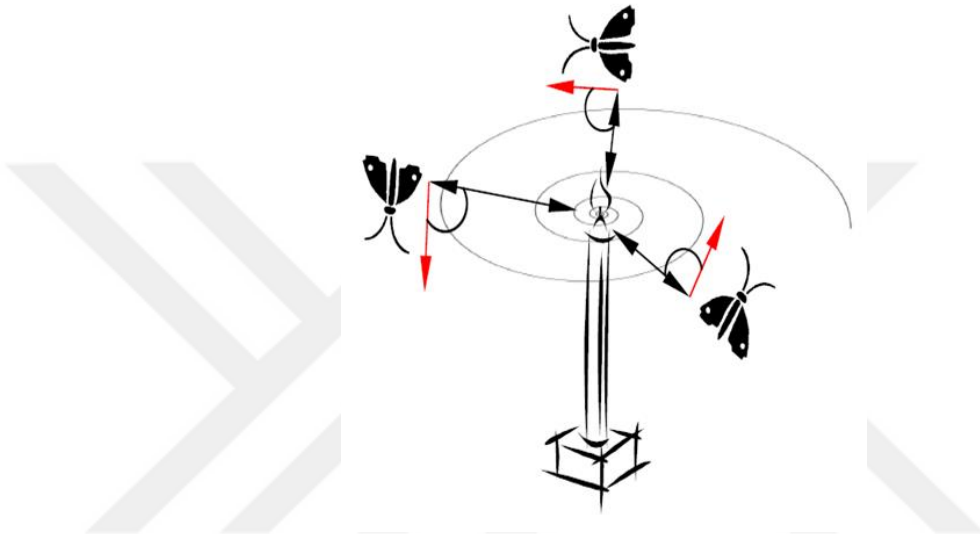
2.14 Güve-Alev Optimizasyonu Algoritması (Moth-Flame Optimization Algorithm (MFO))

Güveler, kelebek ailesine çok benzeyen süslü böceklerdir. Temel olarak, doğada bu böceklerin 160.000'in üzerinde çeşitli türü vardır. Güvelerin yaşamları boyunca larva ve yetişkinlik olmak üzere iki ana kilometre taşı vardır: Larvalar kozalarda güveye dönüştürülür. Güveler hakkında en ilginç gerçek, gece özel navigasyon sistemine sahip olmalarıdır. Ay ışığını kullanarak gece uçmak için evrimleşmişlerdir. Navigasyon için enine yönelimli bir mekanizma kullanmışlardır. Bu yöntemde, bir güve, aya göre sabit bir açıyı koruyarak, düz bir yolda uzun mesafeler için çok etkili bir mekanizma kurarak uçar. Şekil 2.11'de güvenin yöneliminin kavramsal bir modeli gösterilmektedir. Ay uzak olduğu için güve uzağa, bu mekanizmada düz çizgide uçmayı garanti eder. Aynı navigasyon yöntemi insanlar tarafından yapılabilir. Ayın gökyüzünün güney tarafında olduğunu ve bir insanın doğuya gitmek istediğini varsayalım. Yürürken sol tarafını aya tutarsa, doğuya doğru düz bir çizgide ilerleyebilirdi. Enine yönelimin etkinliğine rağmen, genellikle güvelerin ışıkların etrafında spiral şeklinde uçtuğunu görürüz.



Şekil 2.11. Enine yönelim (Mirjalili, 2015).

Güveler insan yapımı yapay bir ışık gördüklerinde, düz bir çizgide uçmak için ışıqla benzer bir açı yapmaya çalışırlar. Böylesi bir ışık ay ile karşılaştırıldığında son derece yakın olduğu için, ışık kaynağına benzer bir açı göstermesi, güveler için yararsız veya ölümcül spiral bir patikaya yol açar. Bu davranışın kavramsal bir modeli Şekil 2.12'de gösterilmiştir. Şekilde, güvenin sonunda ışığa doğru yakınsadığı gözlemlenebilir (Mirjalili, 2015).

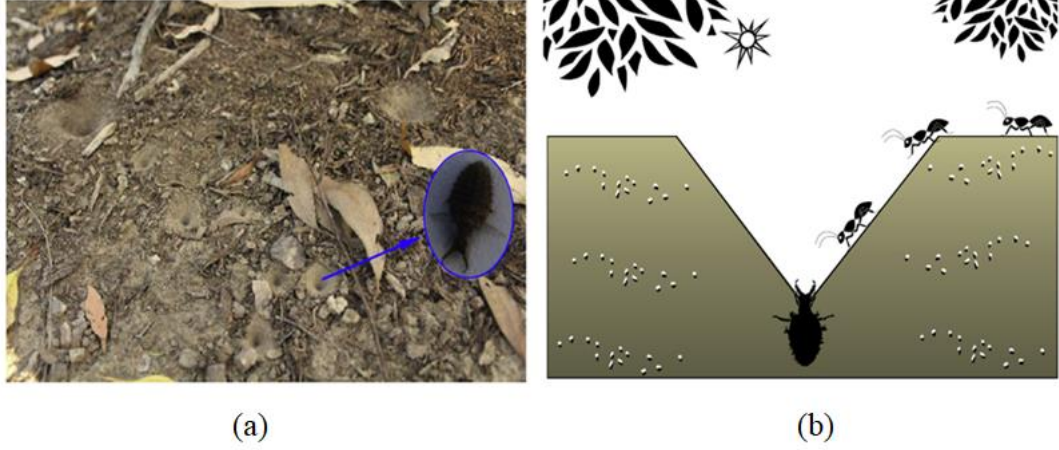


Şekil 2.12. Yakın ışık kaynaklarının etrafında spiral uçan yol (Mirjalili, 2015).

2.15 Karınca Aslanı Optimizasyonu Algoritması (Ant Lion Optimization Algorithm-ALO)

Karınca aslanlarının yaşam döngüsü larva ve yetişkinlik olmak üzere iki ana aşamadan oluşur. Doğal bir yaşam süresi çoğunlukla larvalarda 3 yıla kadar sürebilir. Karınca aslanları koza yetişkin olmak için bir metamorfoza uğrar. Çoğunlukla larva dönemleri avlama, yetişkinlik dönemleri de üreme şeklinde geçer. İsimleri eşsiz avlanma davranışlarından ve onların favori avlarından kaynaklanır. Karınca aslanları larvaları, dairesel bir yol boyunca hareket ederek ve devasa çenesiyle kumları dışarı fırlatarak kumda koni şeklinde bir çukur kazarlar. Şekil 2.13 (a), farklı boyutlarda birkaç koni şeklindeki çukurları göstermektedir. Tuzağı kazdıktan sonra, karınca aslanları koninin tabanının altına oturur ve şekilde gösterildiği gibi çukurda karıncaların ve böceklerin tuzağa gelmesini beklerler. Şekil 2.13 (b)'deki koni şeklinde tuzak, böceklerin ve karıncaların kolayca tuzağın dibine düşmesini sağlar. Karınca aslanları bir avın tuzağa düştüğünü fark ettiğinde, onu yakalamaya çalışır. Ancak, karıncalar ve

böcekler genellikle hemen yakalanmazlar ve tuzaktan kaçmaya çalışırlar. Bu sırada karınca aslanı avına kum fırlatarak avının tuzağın dibine düşmesini sağlar. Tuzağın merkezine gelen av yakalandığında, toprağın altına çekilir ve tüketilir. Avlarını tükettikten sonra, karınca aslanları kalanları çukurun dışına atar ve bir sonraki av için tekrar tuzak kurar (Mirjalili, 2015).

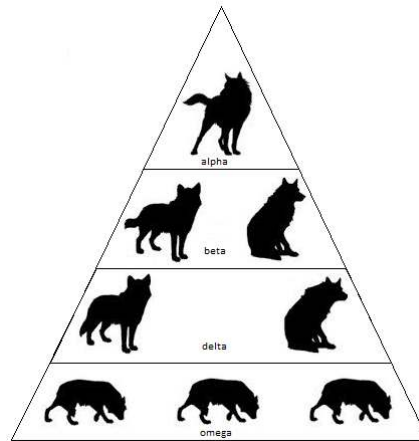


Şekil 2.13. Koni şeklindeki tuzaklar ve karınca aslanlarının avlanma davranışları (Mirjalili, 2015).

3. GRİ KURT OPTİMİZASYONU (GREY WOLF OPTİMİZER-GWO)

Gri kurt optimizasyon algoritması (GWO), 2014 yılında Mirjalili tarafından önerilen gri kurtların avlanma stratejisini ve sosyal liderliğini taklit eden bir optimizasyon algoritmasıdır. Gri kurtlar çoğunlukla grup olarak yaşamayı tercih ederler. Grup büyüklüğü ortalama 5–12 birey arasındadır. Lider kurtlar alfa kurdu olarak adlandırılırlar. Gri kurtların hiyerarşisi, alfa, beta, delta ve omega kurtları olmak üzere dört grup şeklindedir. Lider ya da baskın kurda alfa kurdu denir ve alfa kurt gruptaki diğer kurtları yönetmek açısından en iyi kurttur ve genellikle avlanma, uyuma yeri, uyanma zamanı ve benzeri konularda karar vermekle sorumludur. Kurt grubunun sosyal hiyerarşisindeki ikincisi beta kurttur. Beta birçok etkinlikte lider kurt (alpha) yardımcısı konumundadır. Delta kurdu alfa ve beta kurtlarına uymakla zorunlu üçüncü kurttur ve sadece omega kurtlarına hükmedebilir. Yani omega kurdu en düşük seviyedeki gri kurttur (Mirjalili, 2014). Şekil 3.1’de gri kurt hiyerarşisi gösterilmiştir.

GWO algoritması yeni bir meta-sezgisel algoritma olmasına rağmen, GWO üzerinde iyileştirme ve uygulama çalışmaları literatürde sıkça bulunabilir. GWO ile literatürdeki çalışmalardan bazıları, karmaşık değerli kodlama (Luo, 2015), kaotik harita tabanlı GWO algoritması (Kohli and Arora, 2017), levy uçuş tabanlı GWO algoritması (Amirsadri, vd., 2017), GWO algoritması kullanarak en iyi DC motor kontrolü (Madadi and Motlagh, 2014), çok düzeyli görüntü eşiği için modifiye ayrık gri kurt optimizasyon algoritması (MDGWO) (Li, vd., 2017) şeklinde sayılabilir.



Şekil 3.1. Gri kurt hiyerarşisi (dominantlık yukarıdan aşağıya doğru azalır).

Gri kurtlardaki grup avlanma stratejisi, kurtların başka bir sosyal davranışıdır. Gri kurtlar bu stratejide, öncelikle avın yerini tanırırlar ve alfa kurdunun önderliğinde avı

kuşatırlar. Gri kurtların avlanma stratejisinin matematiksel modelinde alfa, beta ve delta kurtlarının avın yeri hakkında daha iyi bilgi sağladığı varsayılmaktadır. Bundan dolayı, ilk üç en iyi çözüm (alfa, beta, delta), GWO algoritmasında kurtların konumlarını güncellemek için kullanılır. GWO algoritmasında omega kurt bulunmamaktadır. Bu bölümde, sosyal hiyerarşinin izlenmesi, çevrelenmesi, avlanması ve saldıran avların matematiksel modelinden bahsedilecektir (Mirjalili, 2015; Doğan and Yüzgeç, 2018).

3.1 Matematiksel Model

Gri kurt avcılığına ilişkin matematiksel model aşağıda verilmiştir:

$$\vec{D}_\alpha = |\vec{C}_\alpha \cdot \vec{X}_\alpha - \vec{X}_i| \quad (3.1)$$

$$\vec{D}_\beta = |\vec{C}_\beta \cdot \vec{X}_\beta - \vec{X}_i| \quad (3.2)$$

$$\vec{D}_\delta = |\vec{C}_\delta \cdot \vec{X}_\delta - \vec{X}_i| \quad (3.3)$$

$$\vec{U}_\alpha = \vec{X}_\alpha - \vec{A}_\alpha \vec{D}_\alpha \quad (3.4)$$

$$\vec{U}_\beta = \vec{X}_\beta - \vec{A}_\beta \vec{D}_\beta \quad (3.5)$$

$$\vec{U}_\delta = \vec{X}_\delta - \vec{A}_\delta \vec{D}_\delta \quad (3.6)$$

$$\vec{X}_i = (\vec{U}_\alpha + \vec{U}_\beta + \vec{U}_\delta)/3 \quad (3.7)$$

Burada $\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$, av ve kurt (alfa, beta, delta), arasındaki vektör mesafesini, $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$, alfa, beta ve delta kurtları için avın pozisyon vektörünü, $\vec{X}_i$, gri kurtun i'ninci iterasyonda pozisyon vektörünü, $\vec{C}_\alpha, \vec{C}_\beta, \vec{C}_\delta, \vec{A}_\alpha, \vec{A}_\beta, \vec{A}_\delta$, alfa, beta ve delta kurtları için katsayı vektörlerini, $\vec{U}_\alpha, \vec{U}_\beta, \vec{U}_\delta$, alfa, beta ve delta kurtları için deneme vektörlerini göstermektedir. Alfa, beta ve delta kurtları için katsayı vektörleri aşağıdaki gibi hesaplanır:

$$\vec{A}_\alpha = 2\vec{a}\vec{r}_{\alpha 1} - \vec{a} \quad (3.8)$$

$$\vec{C}_\alpha = 2\vec{r}_{\alpha 2} \quad (3.9)$$

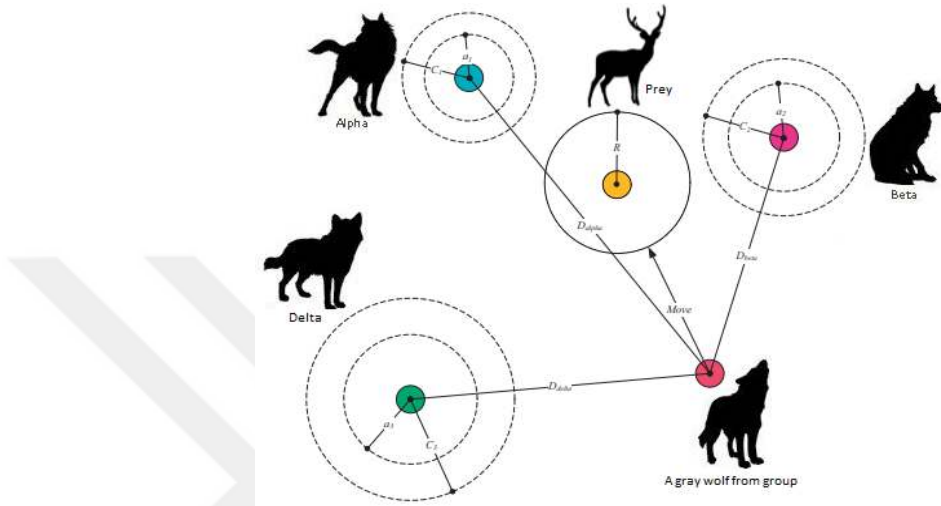
$$\vec{A}_\beta = 2\vec{a}\vec{r}_{\beta 1} - \vec{a} \quad (3.10)$$

$$\vec{C}_\beta = 2\vec{r}_{\beta 2} \quad (3.11)$$

$$\vec{A}_\delta = 2\vec{a}\vec{r}_{\delta 1} - \vec{a} \quad (3.12)$$

$$\vec{C}_\delta = 2\vec{r}_{\delta 2} \quad (3.13)$$

Burada $\vec{a}$, vektörün optimizasyon sırasında doğrusal olarak 2'den 0'a düştüğünü gösterir, $\vec{r}_{\alpha 1}, \vec{r}_{\beta 1}, \vec{r}_{\delta 1}$ ifadeleri [0, 1] de ilk rastgele vektörleri, $\vec{r}_{\alpha 2}, \vec{r}_{\beta 2}, \vec{r}_{\delta 2}$ ifadeleri de [0, 1] 'de ikinci rastgele vektörleri gösterir. Gri kurt grubunun avlanma mekanizması Şekil 3.2'de gösterilmiştir. Gri kurt grubunun üyeleri alfa, beta, delta kurtları ve avlarına göre pozisyonlarını güncelleyerek avlarını yakalar.



Şekil 3.2. Gri kurtların avlama stratejisi (Mirjalili, 2016; Doğan and Yüzgeç, 2018).

GWO algoritmasının kaba kodu aşağıdaki gibidir.

Gri kurtların pozisyonlarını başlat

Gri kurtların maliyet değerlerini hesapla

En iyi gri kurdu alfa kurt olarak kaydet

İkinci en iyi gri kurdu beta kurt olarak kaydet

Üçüncü en iyi gri kurdu delta kurt olarak kaydet

while(iterasyon sayısı < maksimum iterasyon sayısı)

$\vec{a}$ yı azalt

for (her gri kurt için)

Alfa, beta, delta katsayı vektörlerini oluştur

Mesafe vektörlerini hesapla (denklem. 1-3)

Deneme vektörlerini hesapla (denklem. 4-6)

Gri kurtun konumunu güncelle (denklem. 7)

end for

Güncellenen gri kurtların maliyet değerlerini hesaplayın

```

for (her gri kurt için)
    if (gri kurt < alfa kurt)
        alfa kurtu güncelle
    else if (gri kurt < beta kurt)
        beta kurtu güncelle
    else if (gri kurt < delta kurt)
        delta kurtu güncelle
    end if
end for
Seçilenlerin durumunu güncelle
yinelemeyi bir artır

```

end while

alfa kurdunu döndür

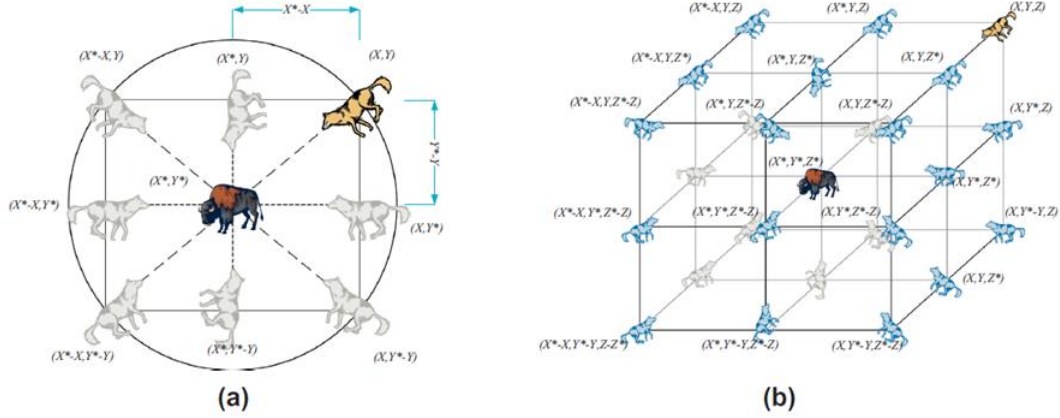
3.2 Sosyal hiyerarşi

GWO tasarlarken kurtların sosyal hiyerarşisini matematiksel olarak modellemek için en uygun çözüm alfa (α) olarak kabul edilir. Sonuç olarak, ikinci ve üçüncü en iyi çözümler sırasıyla beta (β) ve delta (δ) olarak adlandırılır. Diğer aday çözümlerin ise omega (ω) olduğu varsayılmaktadır. GWO algoritmasında av (optimizasyon) α , β ve δ tarafından yönlendirilir. Diğer kurtlar bu üç kurdu takip eder.

3.3 Avın Kuşatılması

Önceki bölümlerde belirtildiği gibi, gri kurtlar av boyunca avını kuşatırlar. 3.1'de verilen denklemlerin etkilerini görmek için iki boyutlu bir pozisyon vektörü ve olası komşuların bazıları Şekil 3.3 (a)'da gösterilmektedir. Şekilde görüldüğü gibi (X,Y) pozisyonundaki gri bir kurt, avının konumuna göre konumunu değiştirebilir (X^*, Y^*). En iyi etkenler etrafında farklı yerler $\vec{A}$ ve $\vec{C}$ değerini ayarlayarak mevcut konuma göre ulaşılabilir. 3 boyutlu uzayda gri bir kurtun güncellenmiş pozisyonları Şekil 3.3 (b) 'de gösterilmiştir. r_1 ve r_2 rastgele vektörlerinin, kurtların Şekil 3.3'te gösterilen noktalar arasında herhangi bir pozisyona ulaşmasına izin verir. Böylece gri bir kurt, bu bölümde bahsedilen eşitlikleri kullanarak herhangi bir rastgele yerde avın çevresindeki boşluğun içindeki yerini güncelleyebilir. Aynı kavram n boyutlarıyla bir arama alanına

geniştirilebilir ve gri kurtlar şimdiye kadar elde edilen en iyi çözüm etrafında küp veya küreler içinde hareket edebilir.



Şekil 3.3. 2D ve 3D uzayında pozisyon vektörleri ve muhtemel bir sonraki konumları (Mirjalili, 2014).

3.4 Avlanma

Gri kurtlar avının yerini tanıma ve onu kuşatma yeteneğine sahiptirler. Av genellikle alfa kurdu tarafından yönlendirilir. Beta ve delta da bazen avlanmaya katılabilirler. Ancak, soyut bir arama alanında avın optimum yeri hakkında hiçbir fikirleri yoktur. Gri kurtların avlanma davranışlarını matematiksel olarak elde etmek için, alfa, beta ve delta kurtlarının avın potansiyel yeri hakkında daha iyi bilgiye sahip olduğu varsayılır. Bu nedenle, elde edilen ilk üç en iyi çözüm saklanır ve diğer kurtların (omegalar dahil) konumlarının en iyi arama faktörleri konumuna göre güncellenmeye zorlanır. Şekil 3.4’de bir arama alanında alfa, beta ve delta kurtlarına göre grup içindeki bir kurdun konumunun nasıl güncellendiği gösterilmektedir. Son konumun, arama alanındaki alfa, beta ve delta pozisyonları tarafından tanımlanan bir daire içinde rastgele bir yerde olacağı gözlemlenebilir. Başka bir deyişle, alfa, beta ve delta avın konumunu tahmin eder ve diğer kurtlar avlarını rastgele sırayla güncellerler.

4. YOL PLANLAMA PROBLEMİ

Robot yol planlama problemi otonom robot teknolojisinde önemli bir yere sahip bir optimizasyon problemidir. Literatürde kullanılan yol bulma yöntemlerinden bazıları Dijkstra Algoritması (Dijkstra, 1959), Bellman-Ford Algoritması (Bellman, 1958) ve A* Algoritmasıdır (Hart, vd., 1968). Bu algoritmalar en kısa yolu bulmayı garanti eder ancak haritanın çok büyük olması durumunda gereken işlem zamanı ve ihtiyaç duyulan hafıza dezavantaj yaratabilmektedir (Boğar, 2016). Bu problemi çözmenin temel amacı, herhangi bir engelle dokunmadan mobil robotun başlangıç noktasından hedef pozisyona en kısa yoldan ulaşmasıdır. Bu yol başlangıç ve hedef pozisyonları, engellerin büyüklüğü, engellerin şekli, engellerin sayısı, bölgenin sınırlarından oluşur (Karakuzu and Babuska, 2014). Yol planlama probleminde kullanılan maliyet fonksiyonu aşağıda verilmiştir:

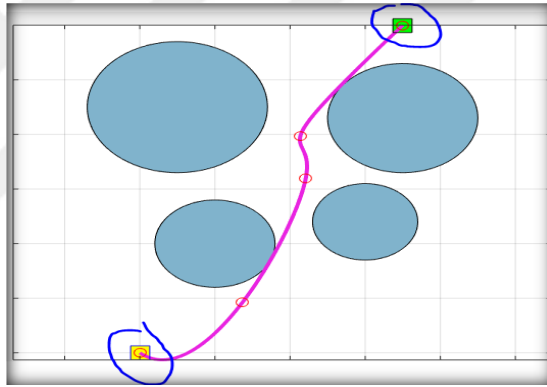
$$J = \min_{x,y} Q(1 + \beta V) \quad (4.1)$$

Burada β ihlal katsayısı (çalışmada 100 olarak alınmıştır), V ihlal maliyetini, Q başlangıç ve hedef nokta arasındaki toplam mesafeyi gösterir.

Yol planlaması robotik ve navigasyon dahil olmak üzere birçok alanda araştırmaya konu olmuştur. Bu çalışmada kullanılan yol probleminde, dairesel tehlike bölgelerinden kaçınarak hedefe giden yol planlamasından bahsedilecektir. Yol planlama probleminin çözümünde kullanılan birçok yöntem vardır. Bu problemlerin ve bunları çözmek için geliştirilen yöntemlerin kapsamlı bir sınıflandırması, 1992 yılında Hwang ve Ahuja tarafından yapılan bir araştırmada bulunabilir. Yol planlama problemini bir örnekle açıklamak gerekirse; Bir robot kol bir dizi engelin arasına yerleştirilir. Robot kolunun bir başlangıç ve bir son konumu düşünüldüğünde, amaç robotun iki pozisyon arasında engellerle çarpışmadan hareket etmesini sağlayacak bir dizi hareket bulmaktır. Bu şekilde robotu engellerin arasında hedefe sürmek için ilk yöntemler 1983 yılında Brooks (Brooks, 1983) tarafından robotun 3 boyutlu modellerini ve çözüm bulmak için engelleri kullandılar. Yani 3 boyutlu bir alan kullandılar. Bu alanda yol planlama problemi, karmaşık bir 3 boyutlu uzayda karmaşık bir 3 boyutlu yapının (robot) hareketlerini bulmayı amaçlar. Diğer başka önemli bir gelişme de 1986 yılında Lozano-Pérez (Lozano-Pérez, 1986) tarafından ileri sürülen bir robot için hareket alanının belirlenmesidir. Bu alanda, bir robotun koordinat n bağımsız

parametreye sahip tek bir nokta ile belirlenir. Fiziksel olarak uygun olmayan pozisyonlar (çarpışma) engel olarak temsil edilir. 3D uzayında, yol planlama problemi, tek bir geometrik nokta için bir yolu temsil eden sürekli bir eğri bulmaktan oluşur ve robotun ilk ve son 3D uzaydaki yerini temsil eden noktaları birleştirir ve herhangi bir engelin varlığında da kesişmez. Bu yöntem, yol planlama probleminin daha yüksek boyutlu bir arama alanına ve daha karmaşık engellere karşı basitleştirilmesini gerektirir. Fiziksel engeller çok karmaşık engeller ile sonuçlanabilir (Alexopoulos and Griffin, 1992; Sariff and Buniyamin, 2006; Hofner and Schmidt, 1995).

Bu tez çalışması kapsamında robot yol planlama problemi için bazı benzetim senaryoları tasarlandı ve bu benzetim senaryoları üzerinden GWO ve diğer sezgisel algoritmaların performansları değerlendirildi Şekil 4.1'de bir robot için yol planlaması senaryosu gösterilmektedir.

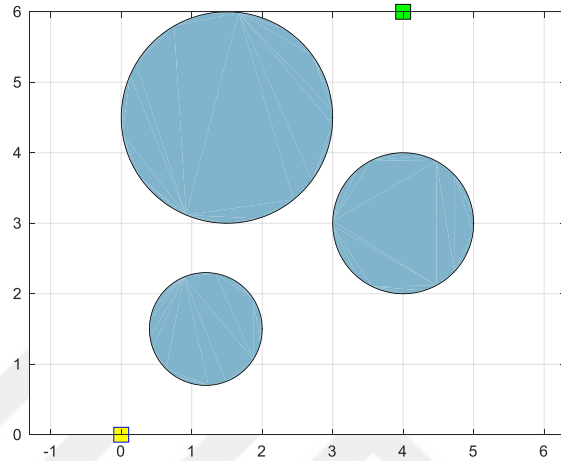


Şekil 4.1. Bir robot için yol planlaması senaryosu.

Şekilde görüldüğü gibi yol planlaması bir başlangıç noktasıyla başlatılıp engellere çarpmadan bitiş noktasıyla sonlanacak şekilde planlanmıştır. Burada dairesel engelleri aşıp hedefe en kısa yoldan ulaşan bir yol planlaması yapmak amaçlanmıştır.

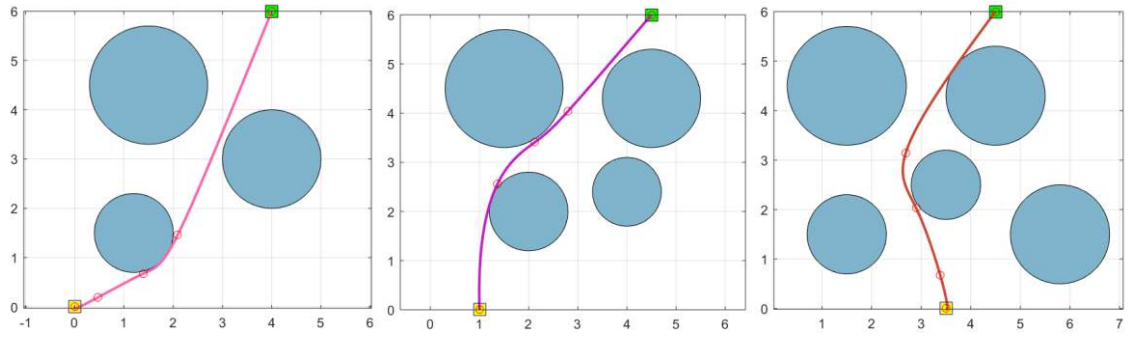
5. YAPILAN DENEYSEL ÇALIŞMALAR

Yol planlama problemi için Gri Kurt Optimizasyonu (GWO) algoritmasının performansını göstermek için www.yarpiz.com web sitesinden bir örnek senaryo alınmıştır. Şekil 5.1’de, bu örnek senaryo gösterilmiştir. 6x6 bölgesinde farklı yarıçapa sahip üç daire şeklinde engeller vardır.



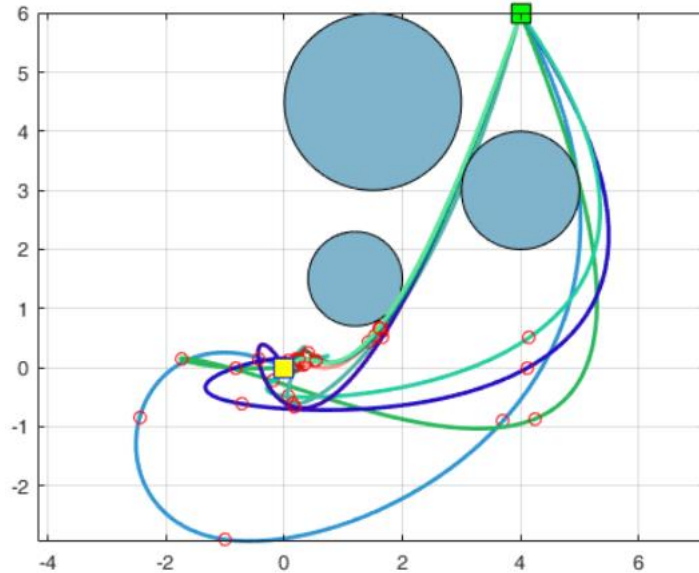
Şekil 5.1. Çalışmada temel alınan robot yol problemi (Yarpiz, 2015).

Sarı kare, mobil robotun başlangıç noktasını, yeşil kare ise hedef noktasını göstermektedir. Bu problem, GWO algoritması yol planlama problemine uyarlanarak çözülmeye çalışılmıştır. Ayrıca GWO algoritması, farksal gelişim, parçacık sürü optimizasyonu, yapay arı kolonisi algoritması, ateş böceği algoritması, yabani ot optimizasyonu algoritması, emperyal rekabetçi algoritma, karıştırılmış sıçramalı kurbağa algoritması, biyocoğrafyaya dayalı optimizasyon algoritması, kovaryans matris adaptasyonu evrimsel stratejisi, uyum arama algoritması, yusufçuk optimizasyonu algoritması, çekirge optimizasyonu algoritması, güve-alev optimizasyonu algoritması ve karınca aslanı optimizasyonu gibi iyi bilinen meta-sezgisel algoritmalar ile karşılaştırılmıştır. Bu çalışmada üç dairesel engel dışında dört ve beş dairesel engelli problemlerde ele alınmıştır. Bu problemler Şekil 5.2’ de gösterilmiştir.

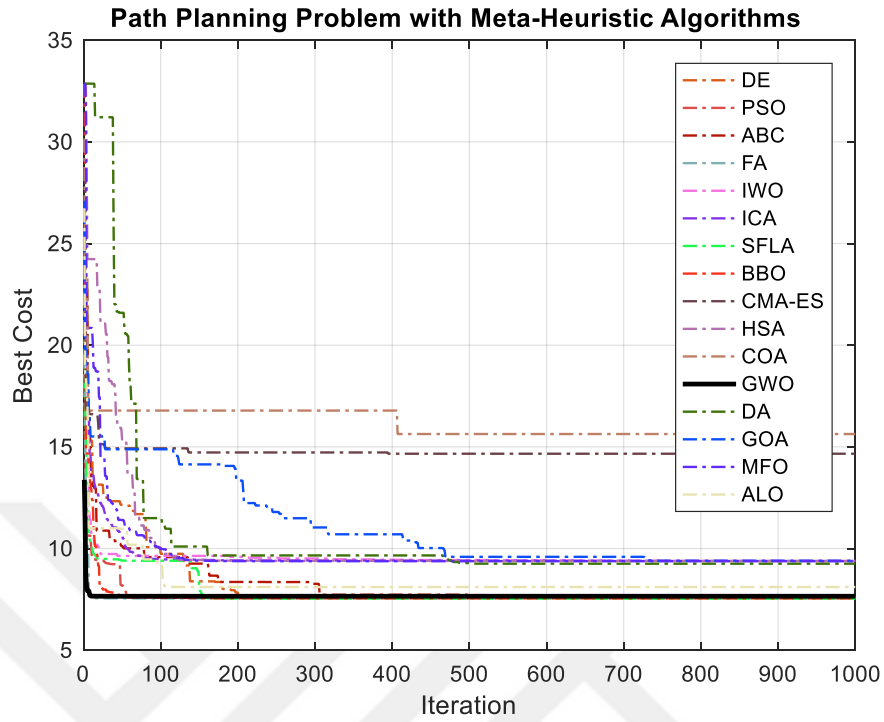


Şekil 5.2. Yol planlama problemi için kullanılan örnek senaryolar.

GWO algoritması ile elde edilen yol planlama problemi (1) için bir çalıştırma boyunca elde edilen tüm çözümler Şekil 5.3’de gösterilmiştir. Bu şekle göre, GWO algoritması optimizasyon sırasında başlangıç ve hedef konumlar arasındaki minimum mesafeyle uygun yolu bulmaya çalışmaktadır. Yani sistem en kısa rotanın engellere takılmayacak şekilde hesaplanabilmesini kısaltma amacıyla başlangıç ve hedef noktalar arasına doğrusal bir hat çekmekte, başlangıçta belirtilen çözüm noktası sayısı kadar bu hat etrafında engellere takılmayacak şekilde random çözüm noktaları atmaktadır. Çözüm noktası engel bölgesinde ise, maliyet fonksiyonuna ihlal değeri eklenir. Ayrıca, her bir yinelemede en iyi çözümün bulunduğu yollar çok az ihlale sahiptir. GWO'nun yakınsama eğrileri ve diğer algoritmaların maliyet değerleri Şekil 5.4’de gösterilmiştir.



Şekil 5.3. GWO algoritması ile elde edilen yol planlama problemi (1) için elde edilen tüm çözümler.



Şekil 5.4. Sezgisel algoritmaların robot yol planlama problemi için en iyi maliyet değerlerinin karşılaştırılması.

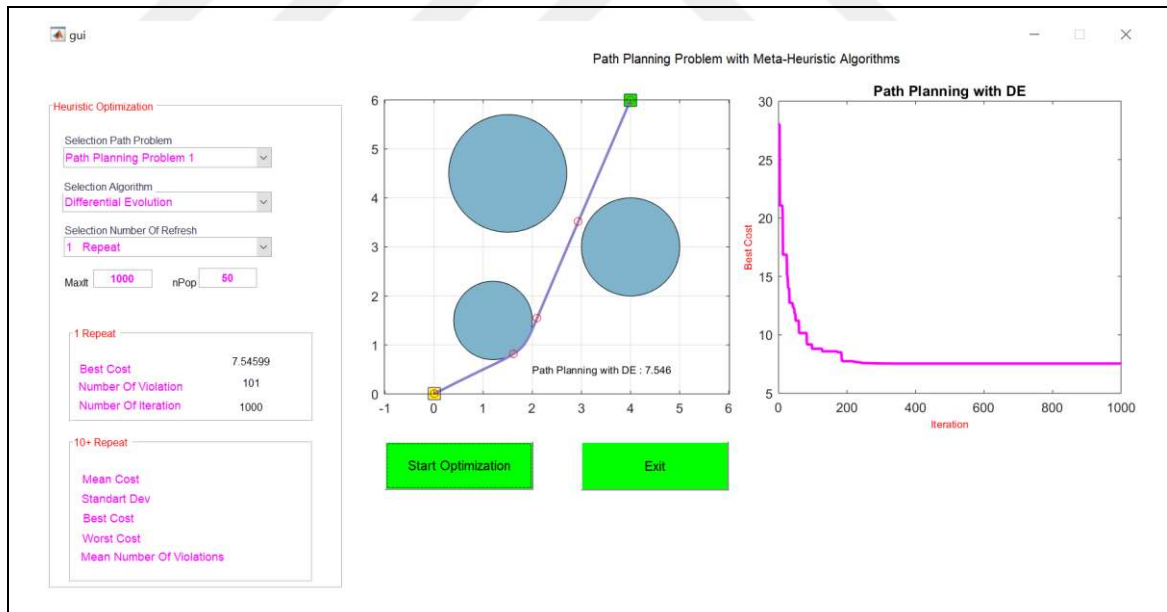
GWO ve diğer meta-sezgisel algoritma kodları, Intel (R) Core (TM) i5-7200U CPU@2.60GHz RAM / 8 ile PC'de çalıştırılmıştır. Nüfus büyüklüğü 50, maksimum iterasyon sayısı 1000'dir. Bu bölümde geliştirilen grafiksel arayüz çalışmasından, bir tekrarlı ve çok tekrarlı koşmalar sonunda elde edilen en iyi yol planlama problemi çözümlerinden bahsedilmektedir. Tablo 5.1'de meta-sezgisel algoritma parametreleri verilmektedir.

Çizelge 5.1. Meta-sezgisel algoritma parametreleri.

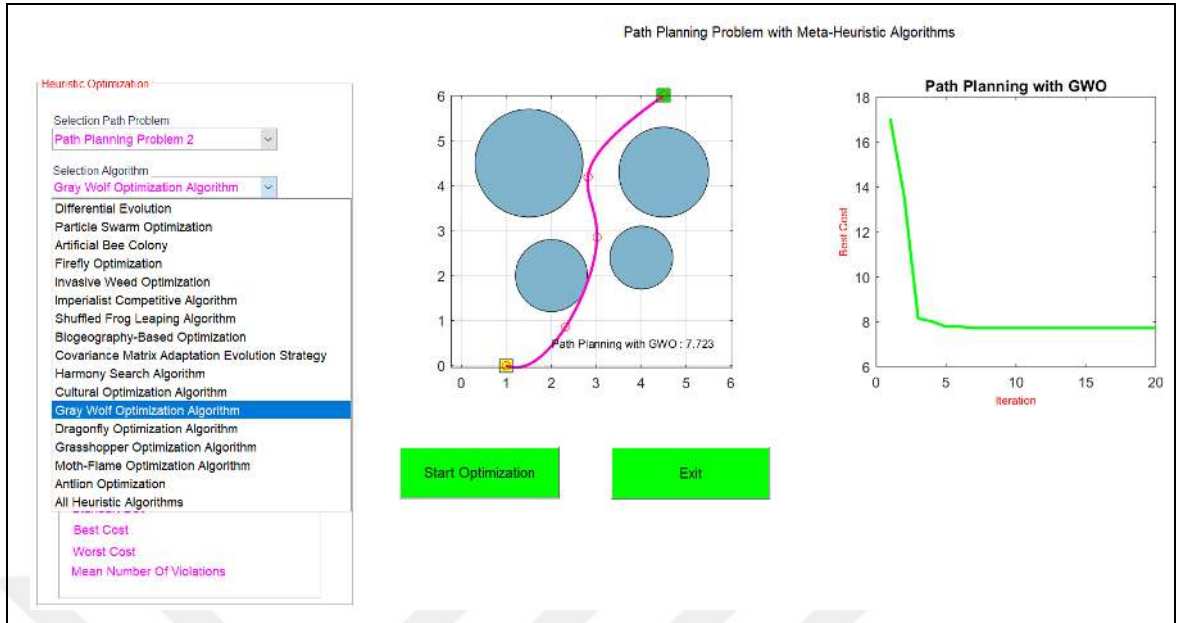
Algoritma	Parametreler
DE	ölçekleme faktörünün alt sınırı : 0.5 ölçekleme faktörünün üst sınırı: 1.0 olasılık değişimi : 0.7 strateji: rand2bin popülasyon sayısı : 50
PSO	atalet ağırlığı: 1.0 atalet ağırlık sönüm oranı: 0.99 kişisel öğrenme katsayısı: 1.5 küresel öğrenme katsayısı: 1.5 popülasyon sayısı : 50
ABC	popülasyon sayısı : 50
FA	ışık emme katsayısı: 1.0 ilk çekim katsayısı: 2.0 mutasyon katsayısı: 0.2 mutasyon katsayısı sönüm oranı : 0.98 popülasyon sayısı : 50
IWO	minimum tohum sayısı = 0; maksimum tohum sayısı = 5; varyans azaltma bileşeni = 2; standart sapma başlangıç değeri = 1; standart sapma son değeri = 0.001;
ICA	devrim olasılığı =0.5; devrim oranı =0.1; koloniler ort. maliyet katsayısı =0.1 popülasyon sayısı : 50
SFLA	popülasyon sayısı : 50
BBO	saklama oranı=0.2; alfa=0.9; mutasyon =0.1; popülasyon sayısı : 50
CMAES	popülasyon sayısı : 50
HSA	yeni harmonilerin sayısı =20; uyum hafıza değerlendirme oranı =0.9; adım ayarlama oranı =0.25 popülasyon sayısı : 50
COA	kabul oranı =0.35; alfa=0.3; beta=0.5; popülasyon sayısı : 50
GWO	popülasyon sayısı : 50
DOA	popülasyon sayısı : 50
GOA	maksimum katsayı=1; minimum katsayı=0.00004; popülasyon sayısı : 50
MFO	popülasyon sayısı : 50
ALO	popülasyon sayısı : 50

5.1 Geliştirilen Grafiksel Arayüzü

GWO ve diğer sezgisel algoritmaların sonuçlarının karşılaştırılması ve parametre değişikliklerinin testler esnasında kolayca değiştirebilmesi amacıyla bir grafiksel arayüzü (GUI) geliştirildi. Bu arayüz yardımıyla algoritma için gerekli tüm parametrelerin değiştirilerek daha hızlı bir şekilde performans değerlendirmesi yapması, sonuçların toplu bir şekilde gösterilmesi amaçlanmıştır. Şekil 5.5'de geliştirilen arayüz görüntüsü gösterilmektedir. Bu arayüzde sol menüde ilk olarak hangi yol probleminin çözüleceği seçilmektedir. Daha sonra bu yol problemini çözecek algoritma, iterasyon sayısı ve popülasyon sayısı girilerek optimizasyon başlatılmaktadır. Grafik arayüzün sağ tarafında seçilen yol problemi ve onun da sağ tarafında maliyet fonksiyonunu gösteren grafikler bulunmaktadır. Arayüzde optimizasyonu başlat ve çıkış olarak iki seçim butonu vardır. Optimizasyonu başlat butonunun tıklanması ile beraber sol tarafta yol planlama problemini çözen yol planı ve sağ tarafta bu yol planlaması sonunda maliyet hesabının iterasyon boyunca durumu grafiklendirilmektedir. Şekil 5.6'da matlab grafik arayüzünde algoritma seçimi gösterilmiştir.



Şekil 5.5. Matlab grafik arayüzü.



Şekil 5.6. Matlab grafik arayüzünde algoritma seçimi.

5.2 Bir Tekrarlı Robot Yol Planlama Sonuçları

Bu bölümde üç farklı robot yol planlaması için GWO ile birlikte 16 sezgisel algoritmanın bir tekrarlı ve 1000 iterasyonluk koşmanın sonuçları grafiksel olarak sunulmuştur. Grafikler üzerinde sezgisel algoritmaların çözdüğü yol problemi ve en iyi maliyet değeri gösterilmiştir. Ek 1’de sezgisel algoritmaların yol planlama problemi-1 için, Ek 2’de yol planlaması problemi-2 için, Ek 3’de yol planlama problemi-3 için elde edilen sonuçlar gösterilmiştir.

5.3 Çok Tekrarlı Robot Yol Planlama Sonuçları

Bu bölümde üç farklı robot yol planlaması için GWO ile birlikte 16 sezgisel algoritmanın çok tekrarlı ve 1000 iterasyonluk koşmanın sonuçları karşılaştırılarak gösterilmiştir. Tablo 5.2, Tablo 5.3 ve Tablo 5.4’de sezgisel algoritmaların yol problemleri için bulduğu ortalama maliyet değeri, standart sapma, en iyi maliyet değeri, en kötü maliyet değeri ve yapılan ihlal sayıları verilmiştir.

Çizelge 5.2. Yol planlama problemi-1 için sezgisel algoritmaların başarımlarını karşılaştırması.

Algoritma	Ortalama Maliyet	Standart Sapma	En iyi maliyet	En kötü maliyet	İhlal
DE	8.08078	0.795201	7.39396	9.20993	84
PSO	7.7316	0.636715	7.40106	8.9392	116
ABC	7.73167	0.0968746	7.59388	7.87042	153
FA	7.547	0.46575	7.55534	7.85746	67
IWO	7.5795	0.478569	7.39418	8.92988	108
ICA	8.18265	0.689499	7.55145	8.99554	74
SFLA	7.46826	0.0742649	7.39563	7.56183	116
BBO	8.65157	0.798151	7.41585	9.40433	44
CMAES	11.3746	0.1.3457	9.74852	13.5873	106
HSA	7.95054	0.747971	7.41641	9.21439	123
COA	15.3939	2.25236	11.9823	19.074	146
GWO	7.8697	0.456704	7.55571	9.07913	48
DOA	8.60678	0.716284	7.54621	9.23957	140
GOA	7.57233	9.362e-16	7.57233	7.57233	203
MFO	9.53455	2.09785	7.3947	15.2974	51
ALO	8.79437	1.76549	7.67733	13.2856	126

Yol Planlama Problemi-1’de sezgisel algoritmaların üç dairesel engelli ortamda çok tekrarlı, 1000 iterasyonlu koşma sonucunda başarımları karşılaştırılmıştır.

Çizelge 5.3. Yol planlama problemi-2 için sezgisel algoritmaların başarımlarını karşılaştırması.

Algoritma	Ortalama Maliyet	Standart Sapma	En iyi maliyet	En kötü maliyet	İhlal
DE	7.67783	0.879965	7.16936	9.48024	95
PSO	7.5042	0.628217	7.17781	8.71194	158
ABC	7.63367	0.187468	7.38629	7.93549	422
FA	7.8574	0.374987	7.43794	8.23562	79
IWO	7.61704	0.718878	7.16965	8.65946	102
ICA	7.34791	0.131997	7.17712	7.61446	168
SFLA	7.1895	0.00941091	7.17065	7.19856	126
BBO	8.80189	0.637928	7.26722	9.62626	62
CMAES	11.86489	0.947247	10.8563	12.84731	114
HSA	7.98909	0.913452	7.24326	9.64002	170
COA	15.6811	1.16401	13.5005	17.8939	135
GWO	6.68643	0.832957	7.47181	9.4844	183
DOA	8.55155	0.647187	7.19665	9.58229	88
GOA	8.75955	0	8.75955	8.75955	77
MFO	8.83585	0.25067	8.6736	9.49905	179
ALO	8.74417	0.272485	8.0073	9.0038	117

Yol Planlama Problemi-2’de sezgisel algoritmaların dört dairesel engelli ortamda çok tekrarlı, 1000 iterasyonlu koşma sonucunda başarımları karşılaştırılmıştır.

Çizelge 5.4. Yol planlama problemi-3 için sezgisel algoritmaların başarımlarını karşılaştırması.

Algoritma	Ortalama Maliyet	Standart Sapma	En iyi maliyet	En kötü maliyet	İhlal
DE	7.45063	1.10272	6.76532	9.04906	114
PSO	8.20554	1.78311	6.5654	11.3861	81
ABC	7.62787	0.468062	7.10248	8.74804	190
FA	6.93386	0.750727	6.56104	9.05301	155
IWO	7.07604	1.22176	7.54057	10.54	77
ICA	6.58955	0.680678	6.54874	6.78144	100
SFLA	6.57531	0.0708747	6.54292	6.77495	142
BBO	11.2325	0.774812	9.05398	11.6733	47
CMAES	11.7584	0.784567	9.64733	12.1485	86
HSA	9.6431	1.46785	7.19452	11.8758	186
COA	15.037	1.27931	13.054	17.4319	184
GWO	8.64657	2.12244	6.75297	12.8113	178
DOA	6.7664	0.998959	6.56901	6.89259	143
GOA	6.83444	9.362e-16	6.83444	6.83444	70
MFO	9.06067	4.17155	6.55015	18.7276	110
ALO	7.19512	0.682531	6.56589	8.54264	417

Yol Planlama Problemi-3'te sezgisel algoritmaların beş dairesel engelli ortamda çok tekrarlı, 1000 iterasyonlu koşma sonucunda başarımları karşılaştırılmıştır.

6. SONUÇ

Bu çalışmada robot yolu planlama problemi tartışılmış ve bu problemin çözümü için GWO algoritması önerilmiştir. Algoritmanın yol planlama problemini çözme performansının değerlendirilmesi için iyi bilinen ve güncel 16 meta-sezgisel algoritmalar kullanılmıştır. Karşılaştırma sonuçları, önerilen GWO algoritmasının çok rekabetçi sonuçlar sağlayabildiğini göstermektedir. Deneysel çalışmalarda GWO algoritması ile elde edilen yol planlama probleminin tüm çözümleri değerlendirildi. Buna göre, GWO algoritması optimizasyon sırasında başlangıç ve hedef konumlar arasında her zaman minimum mesafeyle uygun yolu bulmaya çalışmaktadır. Ayrıca, her bir yinelemede en iyi çözümün bulunduğu yollar çok az ihlale sahiptir. Bu GWO'nun performansının rekabetçi sonuç verdiğini ve yol planlaması için alternatif bir algoritma olabileceğini göstermektedir.

GWO ve diğer sezgisel algoritmaların sonuçlarının daha rahat karşılaştırılması ve parametre değişikliklerinin testler esnasında kolayca değiştirilmesini sağlayan bir grafiksel arayüzü (GUI) geliştirildi. Bu arayüz yardımıyla algoritma için gerekli tüm parametrelerin değiştirilerek daha hızlı bir şekilde performans değerlendirmesi yapması, sonuçların toplu bir şekilde gösterilmesi amaçlanmıştır.

Yol planlama probleminde; bir tekrarlı ve çok tekrarlı tablo sonuçlarına ve diğer deneysel sonuçlarına bakıldığında GWO' nun diğer algoritmalarla rekabet edebildiğini ve yol planlamasında iyi bir performans sergilediği görülmektedir.

KAYNAKLAR

- Canayaz, M., “Cırcır Bözceği Algoritması: Yeni Bir Meta-Sezgisel Yaklaşım ve Uygulamaları”, Doktora Tezi, *İnönü Üniversitesi Fen Bilimleri Enstitüsü*, Malatya (2005).
- Behesti, Z. and Shamsuddin, S. M. H, “A review of population-based meta-heuristic algorithm”, *International Journal of Advanced in Soft Computing and its Applications*, 5(1): 1-35 (2013).
- Blum, C., Puchinger, J., Raidl, G. R. and Roli, A., “A brief survey on hybrid metaheuristics”, in *Bioinspired Optimization Methods and their Applications – Proceeding of the 4th International Conference on Bionspired Optimization Methods and their Applications*, BIOMA 2010, 3-16 (2010).
- Nowicki, E. And Smutnicki, C., “A fast tabu search algorithm for the permutation flow-shop problem”, *Eur. J. Oper. Res.*, 91(1): 160-175 (1996).
- Gendreau, M., Lori, M., Laporte, G. And Martello, S., “A Tabu Search Algorithm for a Routing and Container Loading Problem”, *Transp. Sci.*, 40(3): 342-350 (2006).
- Kirkpatrick, S., Gelat, C. D. and Vecchi, M. P., “Optimization by Simulated Annealing”, *Science* (80-.), 220(4598): 671-680 (1983).
- Mavridou, T. D. and Pardalos, P.M., “Simulated Annealing and Genetic Algorithms for the Facility Layout Problem: A Survey”, *Comput. Optim. Appl.*, 7: 111-126 (1997).
- Goldberg, D. E., “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-We (1989).
- Noraini, M. And Geraghty, J., “Genetic algorithm performance with different selection strategies in solving TSP”, *World Congr. Eng.*, 2(978-988-19251-4-5): 4-9 (2011).
- Storn, R. and Price, K., “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces”, *J. Glob. Optim.*, 11(4): 41-359 (1997).
- Price, K. V., Storn, R. M. and Lampinen, J. A., *Differential Evolution: A Practical Approach to Global Optimization*, 28 (2005).
- Yüzgeç, U., “Performance comparison of differential evolution techniques on optimization of feeding profile for an industrial scale baker’s yeast fermentation process”, *ISA Trans.*, 49(1): 167-176 (2010).

KAYNAKLAR (Devam Ediyor)

- Kennedy, J. and Eberhart, R., "Particle swarm optimization," Neural Networks, 1995. Proceedings., *IEEE Int. Conf.*, 4: 1942–1948 (1995).
- Clerc, M. and Kennedy, J., "The particle swarm – explosion, stability, and convergence in a multidimensional complex space", *IEEE Trans. Evol. Comput.*, 6(1): 58-73 (2002).
- Karaboga, D., "An idea based on Honey Bee Swarm for Numerical Optimization", *Tech. Rep. TR06*, Erciyes Univ., TR06: 10 (2005).
- Karaboga, D. and Akay, B., "A comparative study of Artificial Bee Colony algorithm", *Appl. Math. Comput.*, 214(1): 108-132 (2009).
- Dorigo, M., Maniezzo, V. and Coloni, A., "Ant system: Optimization by a colony of cooperating agent", *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, 6(1): 29-41 (1996).
- Dorigo, M., Birattari, M. and Stutzle, T., "Ant colony optimization", *IEEE Comput. Intell. Mag.*, 1(4): 28-39 (2006).
- Li, X., Shao, Z. and Qian, J., "An optimization method based on autonomous animats: fish-swarm algorithm", *Syst. Eng. Theory Pract.*, 22(11): 32-38 (2002).
- Neshat, M., Sepidnam, G., Sargolzaei, M. and Toosi, A. N., "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications", *Artif. Intell. Rev.*, 42(4): 965-997 (2014).
- Hofmeyr, S. A. and Forrest, S., "Architecture for an Artificial Immune System", *Evol. Comput.*, 8(4): 443-473 (2000).
- Timmis, J., Andrews, P. and Hart, E., "On artificial immune system and swarm intelligence", *Swarm Intell.*, 4(4): 247-273 (2010).
- Das, S., Biswas, A., Dasgupta, S. And Abraham, A., "Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis and Applications", *Found. Comput. Intell.* 3(3): 23-55 (2009).
- Passino, K. M., "Biomimicry of bacterial foraging for distributed optimizaiton and control", *Control Syst. IEEE*, 22(3): 52-67 (2002).
- Passino, K. M., "Bacterial Foraging Optimization", *Int. J. Swarm Intell Res.*, 1(1): 1-16 (2010).
- Yang, X. S. and Deb, S., "Cuckoo search: Recent advances and applications", *Neural Computing and Applications*, 24(1): 169-174 (2014).

KAYNAKLAR (Devam Ediyor)

- Yang, X.-S., "Cuckoo Search via Levy flights", in **2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)**, 210-214 (2009).
- Yang, X. S., "Firefly Algorithm", *Nature-Inspired Metaheuristic Algorithm*, 79-90, (2007).
- Yang, X. S., "Firefly algorithms for multimodal optimization", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5792: 169-178 (2009).
- Isçan, H. and Gunduz, M., "Parameter Analysis on Fruit Fly Optimization Algorithm", **J. Comput. Commun.**, 2(March): 137-141 (2014).
- Rashedi, E., Nezamabadi-pour, H. and Saryazdi, S., "GSA: A Gravitational Search Algorithm", **Inf. Sci. (Ny)**, 179(13): 2232-2248 (2009).
- Sabri, M., Puteh, M. and Mahmood, M. R., "A review of gravitational search algorithm", **Int. J. Adv. Soft Comput. its Appl.**, 5(3): (2013).
- Atashpaz-Gargari, E. and Lucas, C., "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition", in **2007 IEEE Congress on Evolutionary Computation, CEC 2007**, 4661-4667 (2007).
- Hosseini, S. and Al Khaled, A., "A survey on the Imperialist Competitive Algorithm metaheuristic: Implementation in engineering domain and directions for future research", **Applied Soft Computing Journal**, 24: 1078-1094 (2014).
- Mirjalili, S., "The ant lion optimizer", **Adv. Eng. Softw.**, 83: 80-98 (2015).
- Kiliç, H. and Yüzgeç, U., "Improved antlion optimization algorithm", in **2nd International Conference on Computer Science and engineering, UBMK 2017**, 84-88 (2017).
- Mirjalili, S., "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete and multi-objective problems", **Neural Comput. App.**, 27(4): 1053-1073 (2016).
- Sree Ranjini, S. R. And Murugan, S., "Memory based Hybrid Dragonfly Algorithm for numerical optimization problems", **Expert Syst. Appl.**, 83: 63-78 (2017).
- Mirjalili, S. and Lewis, A., "The Whale Optimization Algorithm", **Adv. Eng. Softw.**, 95: 51-67 (2016).
- Mirjalili, S., Mirjalili, S. M. and Lewis, A., "Grey Wolf Optimizer", **Adv. Eng. Softw.**, 69: 46-61 (2014).

KAYNAKLAR (Devam Ediyor)

- Luo, Q., Zhang, S., Li, Z. and Zhou, Y., “A Novel Complex-Valued Encoding Grey Wolf Optimization Algorithm”, *Algorithms*, 9(1): 4, (2015).
- Kohli, M. and Arora, S., “Chaotic grey wolf optimization algorithm for constrained optimization problems”, *J. Comput. Des. Eng.*, 2017.
- Amirsadri, S., Mousavirad, S. J. and Ebrahimpour-Komleh, H., “A Levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training”, *Neural Computing and Applications*, 1-14 (2017).
- Madadi, A. and Motlagh, M. M., “Optimal Control of DC motor using Grey Wolf Optimizer Algorithm”, *Tech. J. Eng. Appl.*, 4(4): 373-379 (2014).
- Mohanty, S., Subudhi, B. and Ray, P.K., “A Grey Wolf-Assisted Perturb & Observe MPPT Algorithm for a PV System”, *IEEE Trans. Energy Convers.*, 32(1): 340-347 (2017).
- Li, L., Sun, L., Guo, J., Qi, J., Xu, B. and Li, S., “Modified Discrete Grey Wolf Optimizer Algorithm for Multilevel Image Thresholding”, *Comput. Intell. Neurosci.*, vol. 2017 (2017).
- Alexopoulos, C. and Griffin, P. M., “Path planning for a mobile robot”, *IEEE Trans. Syst. Man. Cybern.*, 22(2): 1132-1139 (1992).
- Bellman, R., “On a routing problem. *Quarterly Applied Mathematics*”, XVI(1): 87-90 (1958).
- Sariff, N. and Buniyamin, N., “An overview of autonomous mobile robot path planning algorithms”, in Research and Development, 2006. *SCORED 2006. 4th Student Conference on, 2006*, Selangor, Malaysia 183-188 (2006).
- Hofner, C. and Schmidt, G., “Path planning and guidance techniques for an autonomous mobile cleaning robot”, *Rob. Auton. Syst.*, 14(2-3): 199-212 (1995).
- Dijkstra, E.W., “A Note on Two Problems in Connexion with Graphs”, *Numerische Mathematic 1*, 1(1): 269-271 (1959).
- Brooks, R. A., “Find-Path For A Puma-Class Robot”, *MIT Artificial Intelligence Laboratory 545 Technology Square Cambridge, Massachusest, 02139, USA.*, <https://pdfs.semanticscholar.org/dc22/b0f019f19abdb92936679a2ef58.pdf> (1983).
- Lozano-Perez, T., “A Simple Motion Planning Algorithm For General Robot Manipulators”, *MIT Artificial Intelligence Laboratory 545 Technology Square Combridge, Mass. 02139 USA*, 3(3): 224-238 (1986).
- Kavian, Y. S., Rashedi, A., Mahani, A., Ghassemlooy, Z., “Routing and wavelength assignment in optical networks using Artificial Bee Colony algorithm”, *Optik – International Journal for Light and Electron Optics*, 124: 243-249 (2013).

KAYNAKLAR (Devam Ediyor)

- Mehrabian, A. R., Lucas, C., “A novel numerical optimization algorithm inspired from weed colonization”, *Ecological Informatics*, 1(4): 355-366 (2006).
- Nazari-Shirkouhi, S., Eivazy, H., Ghodsi, R., Rezaie, K., Atashpaz-Gargari, E., “Solving the integrated product mix-outsourcing problem using the Imperialist Competitive Algorithm”, *Expert System with Applications*, 37(12): 7615-7626 (2010).
- Yarpiz, Optimal Robot Path Planning using PSO. <http://yarpiz.com/403/ypap115-path-planning>, (Accessed 28 March 2018).
- Nevşehir Bilim ve Teknoloji Dergisi, “Güncel Sürü Zekası Optimizasyon Algoritmaları”, Int: <http://dergipark.ulakbim.gov.tr/nevbiltek/article/view/1035000078/1035000079>, (Ziyaret edilme tarihi 22.04.2018).
- Boğar, E., “Tek ve Çok Amaçlı Robot Yol Planlama Problemi için Hibrit Bir Optimizasyon Yöntemi”, Yüksek Lisans Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Denizli (2016).
- Hart, P., Nilsson, N. and Raphael, B., “A formal basis for the heuristic determination of minimum cost paths”, *IEEE trans. Sys. Sci. and Cyb.* 4: 100-107 (1968).
- Kenny, V., Nathal, M. and Saldana, S., “Heuristic Algorithms”, Int: https://optimization.mccormick.northwestern.edu/index.php/Heuristic_algorithms, (Ziyaret edilme tarihi 10.05.2018).
- Farksal Gelişim, “Farksal Gelişim (Differential Evolution) Algorithm Nasıl Çalışır”, Int: <http://ahmetcevahircinar.com.tr/farksal-gelisim-differential-evolution-algoritmasi-nasil-calisir>, (Ziyaret edilme tarihi 12.03.2018).
- Mnemstudio, “Particle Swarm Introduction”, Int: <http://mnemstudio.org/particle-swarm-introduction.htm>, (Ziyaret edilme tarihi).
- PSO algorithm, “PSO algorithm”, Int: <http://www.turingfinance.com/about/psa-algorithm/#prettyPhoto>, (Ziyaret edilme tarihi 13.03.2018).
- Karaboğa, “Artificial Bee Colony Algorithm”, Int: http://www.scholarpedia.org/article/Artificial_bee_colony_algorithm, (Ziyaret edilme tarihi 16.03.2018).
- SlideShare, yapay arı kolonisi algoritması“, Int: https://www.slideshare.net/Orhan_Eripek/yapay-ar-kolonisi-algoritmasi, (Ziyaret edilme tarihi 17.03.2018).
- Saim, B., “Yapay arı kolonisi algoritması”, Int: <https://www.bilalsaim.com/yapay-ari-kolonisi-algoritmasi-article-bee-colony-algorithm-h1630>, (Ziyaret edilme tarihi 22.04.2018).

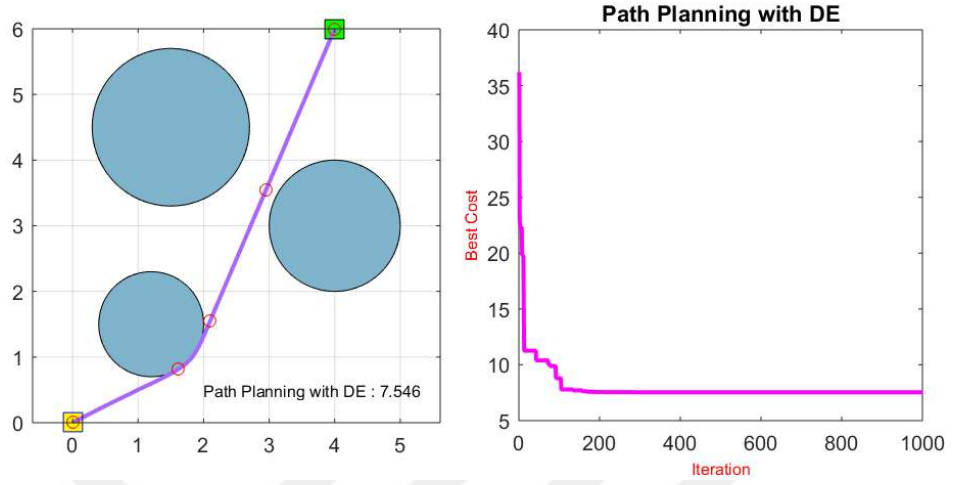
KAYNAKLAR (Devam Ediyor)

- Saim, B., “Ateş Böceği Algoritması”, Int: <https://www.bilalsaim.com/ates-bocegi-algoritmasi-fafirefly-algorithm-h1635>, (Ziyaret edilme tarihi 02.05.2018).
- Imperialist Competitive Algorithm, “Imperialist Competitive Algorithm for Engineering Design Problems”, Int: <https://pdfs.semanticscholar.org/1f23/0cc5e1264599f7ed6525505d2f5406f16da1.pdf>, (Ziyaret edilme tarihi 05.05.2018).
- Yarpiz, Biogeography-Based-Optimization, Int: <http://yarpiz.com/239/ypea113-biogeography-based-optimization>, (Ziyaret edilme tarihi 07.05.2018).
- Biogeography Based Optimization, “Biogeography-Based Optimization: A 10-Year Review”, Int: <https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/45804/versions/2/screenshot.jpg> (MLP), (Ziyaret edilme tarihi 10.05.2018).
- Ma, H., Simon, D., Senior Member, IEEE, Patrick Siarry, Zhile Yang and Minrui Fei, “Biogeography-Based optimization: A 10-Year Review”, *Ieee Transactions On Emerging Topics In Computational Intelligence*, 1(5): 391-407 (2007).
- Al-Betar, M. A., Khader, A. T., Geen, Z. W., Doush, I. A. & Awadallah, M. A., An analysis of selection methods in memory consideration for harmony search. *Applied Mathematics and Computation*, 219(22): 10753-10767 (2013).
- Kuo, H. C. and Lin, C. H., “Cultural Evolution Algorithm for Global Optimizations and its Applications”, *Journal of Applied Research and Technology*, 11: 510-522 (2013).
- Yan, Wu, Zhang, An Efficient Function Optimization Algorithm based on Culture Evolution, Int: <https://pdfs.semanticscholar.org/a0b4/2966bbe76fb723d634f27f0cfcc300189b35.pdf>, (Ziyaret edilme tarihi 12.05.2018).
- Mirjalili, Like Moths to A Flame, Int: <http://www.alimirjalili.com/MFO.html>, (Ziyaret edilme tarihi 10.05.2018).
- Doğan, L., Yüzgeç, U., “Robot Path Planning using Grey Wolf Optimizer”, *International Conference on Advanced Technologies, Computer Engineering and Science, ICATCES 2018 Safranbolu*, 69-74 (2018).
- Yüzgeç, U., Doğan, L., “Solving Path Planning Problem in Mobile Robots using Meta-Heuristic Algorithms”, *International Congress on Engineering and Life Sciences, ICELIS 2018 Kastamonu*, 611(2018).

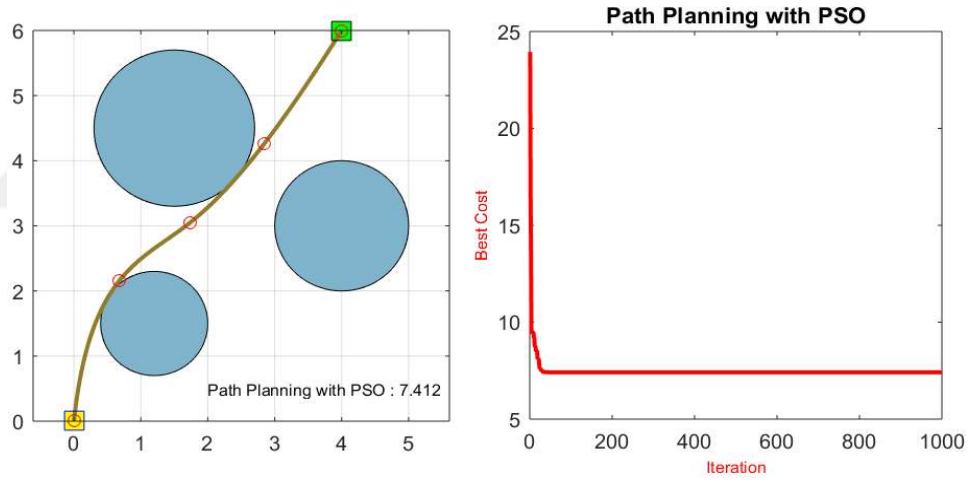
KAYNAKLAR (Devam Ediyor)

- Karakuzu, C., Babuska, R., “On-line Path Planning for Mobile Robots Based on Basic Heuristic Guidance”, *Conference: 15. EU/ME Workshop Metaheuristic and Engineering, At İstanbul/ Turkey* (2014).
- Saremi, Mirjalili, Lewis, Grasshopper Optimization Algorithm: Theory and application, Int: <https://www.sciencedirect.com/science/article/pii/S0965997816305646>, (Ziyaret edilme tarihi 11.05.2018).
- Mirjalili, Ant Lion Optimizer (ALO), Int: <http://www.alimirjalili.com/ALO.html>, (Ziyaret edilme tarihi 12.05.2018).
- Mirjalili, Dragonfly Algorithm (DA), Int: <http://www.alimirjalili.com/DA.html>, (Ziyaret edilme tarihi 12.05.2018).
- Alireza, R. V. and Mirzaei, A. H, “A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem”, *uters and Industrial Engineering* 53(4): November 2007: 642-666 (2007).
- Rifai, A. P., Huu-ThoNguyen, Aoyama, H., Dawal, S. Z. M., Masruroh, N. A., “Non-dominated sorting biogeography-based optimization for bi-objective reentrant flexible manufacturing system scheduling”, *Applied Soft Computing*, 62: 187-202 (2018).

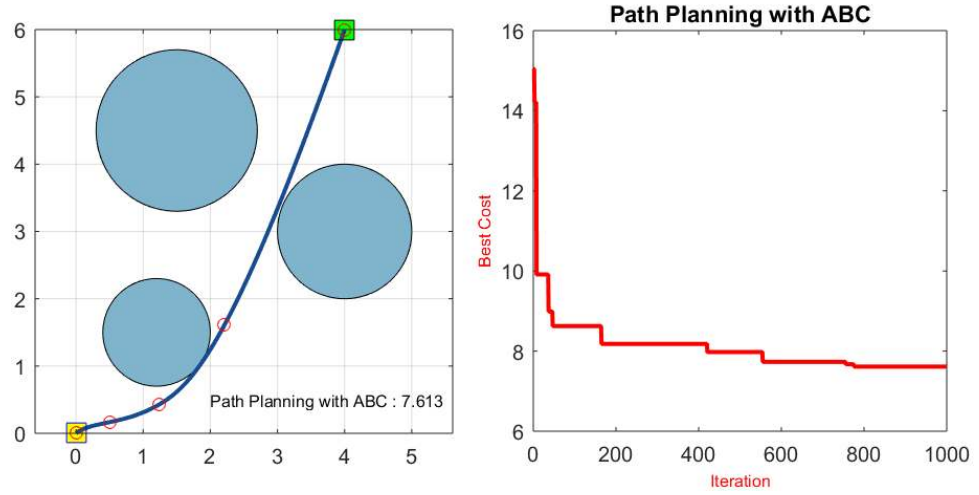
EK 1: Sezgisel algoritmaların yol planlama problemi-1 için başarımlarının karşılaştırılması (A-P).



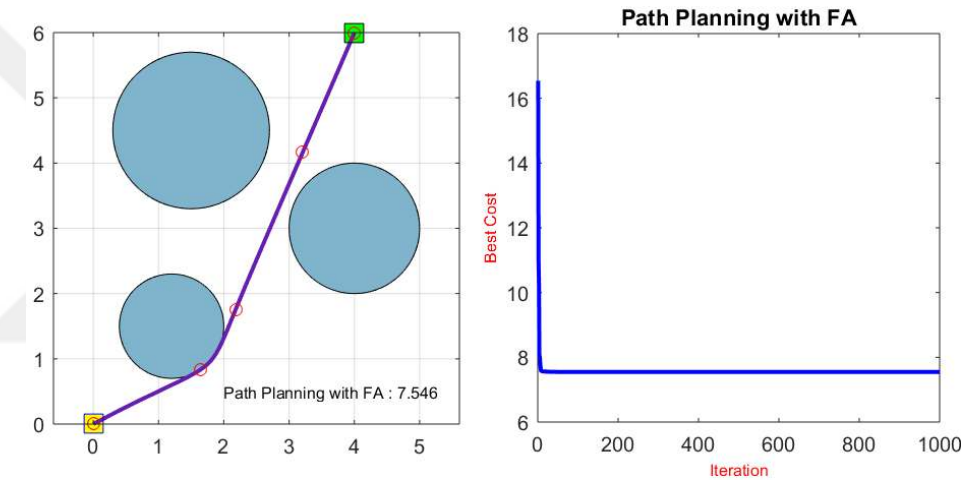
(A)



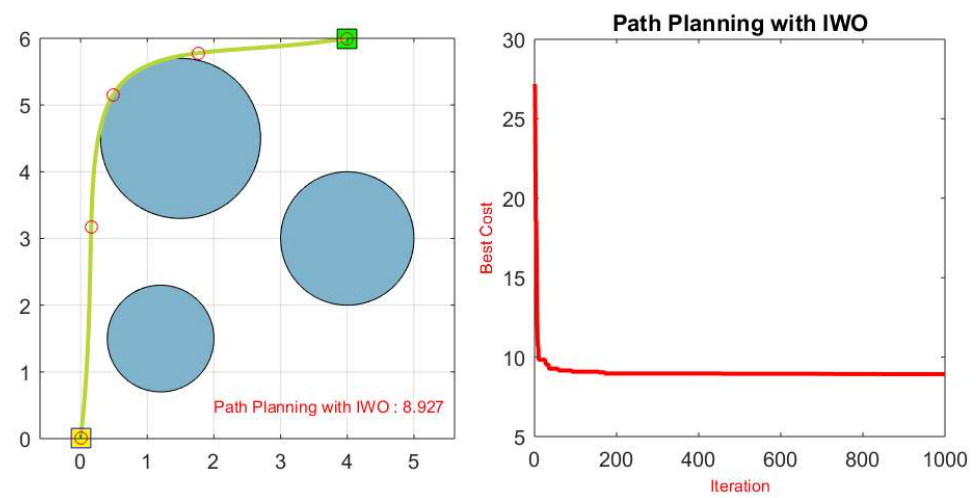
(B)



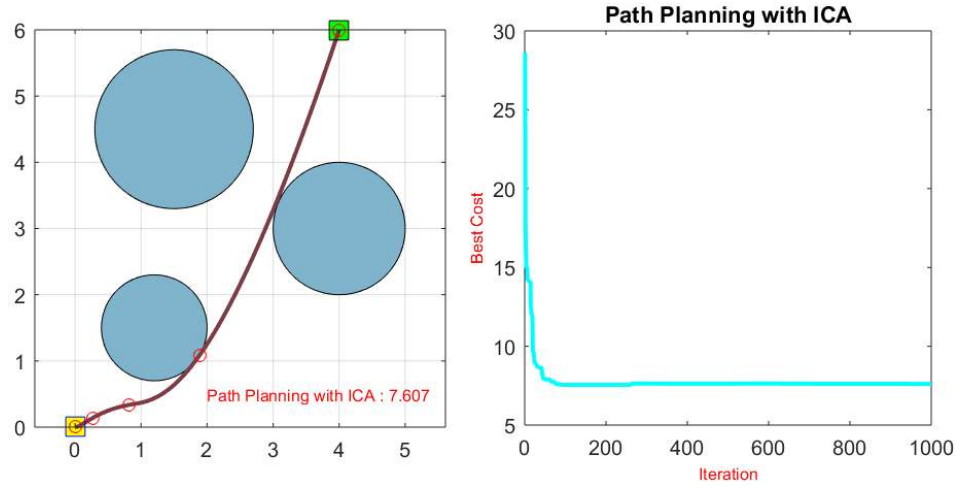
(C)



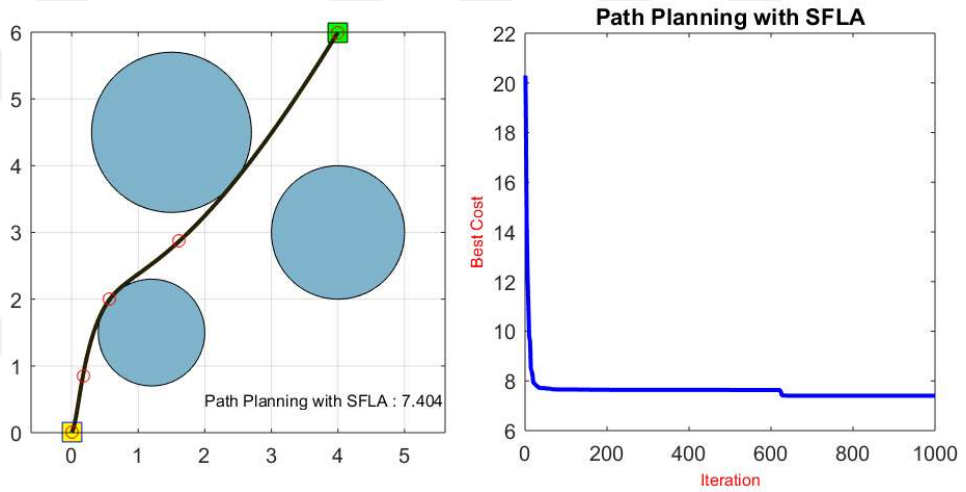
(D)



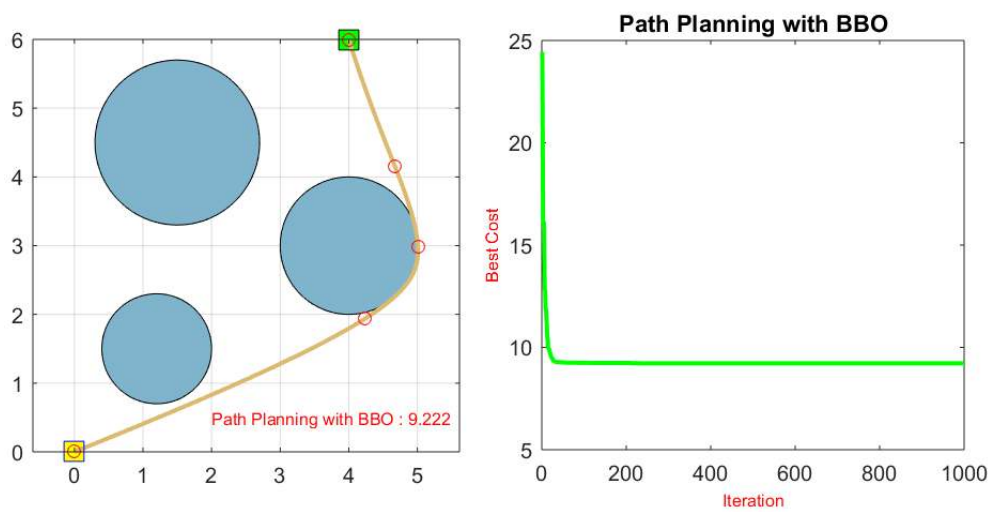
(E)



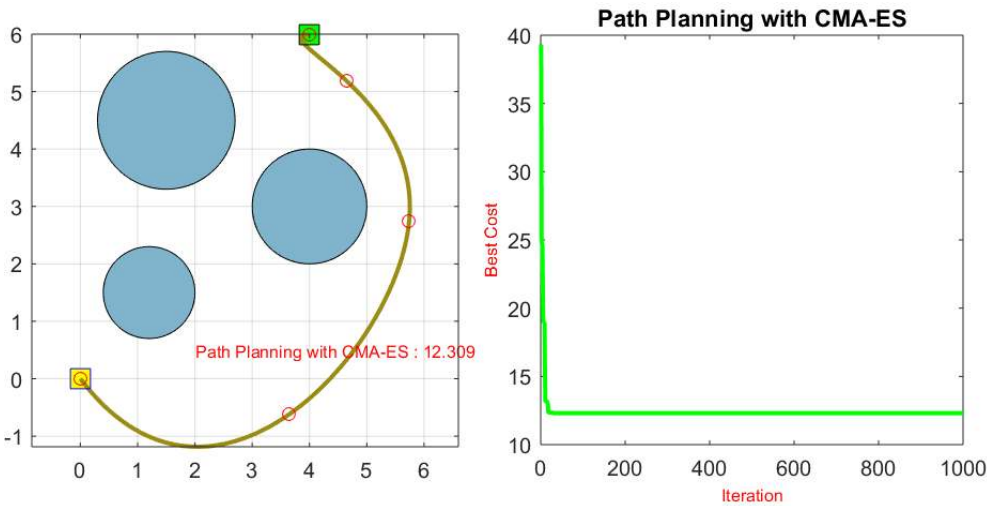
(F)



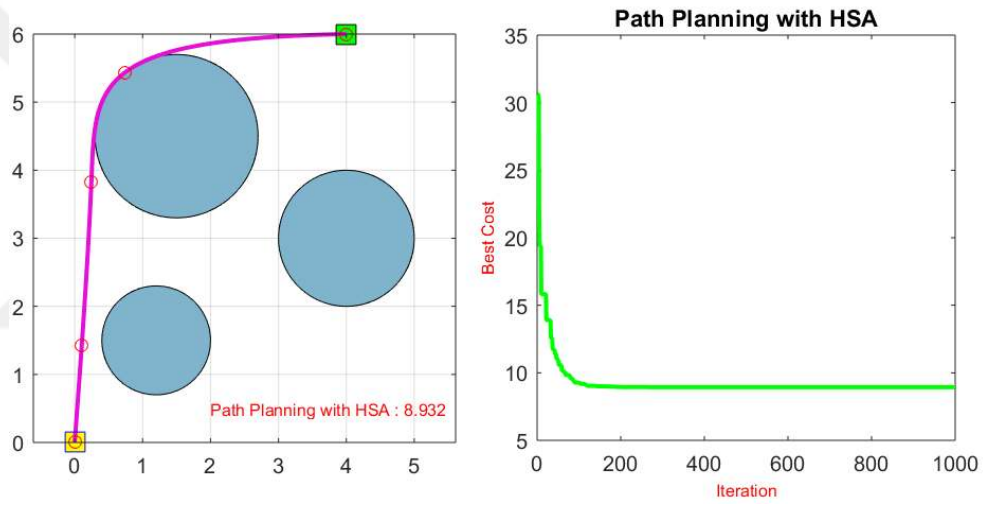
(G)



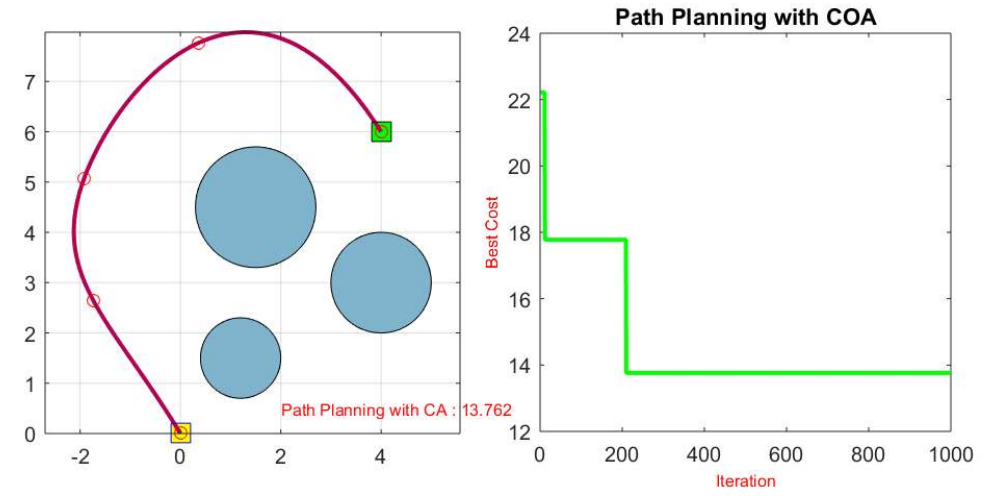
(H)



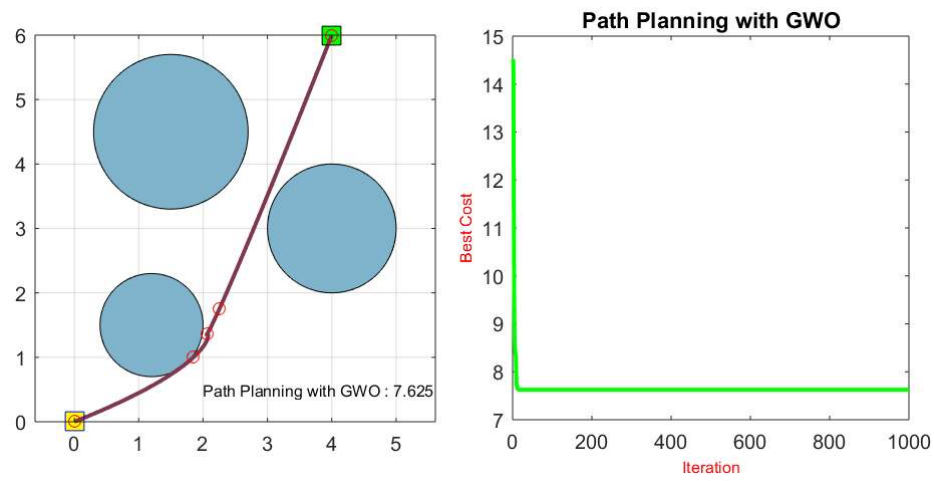
(I)



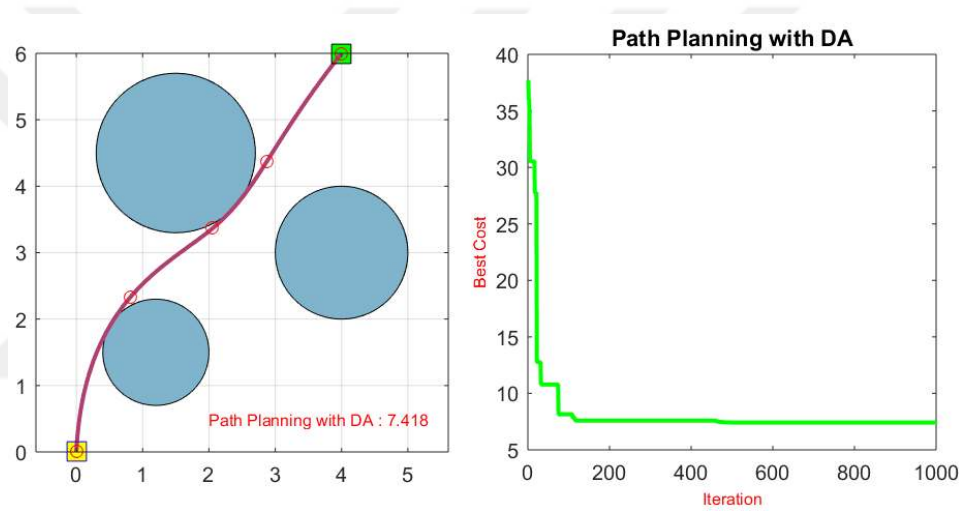
(J)



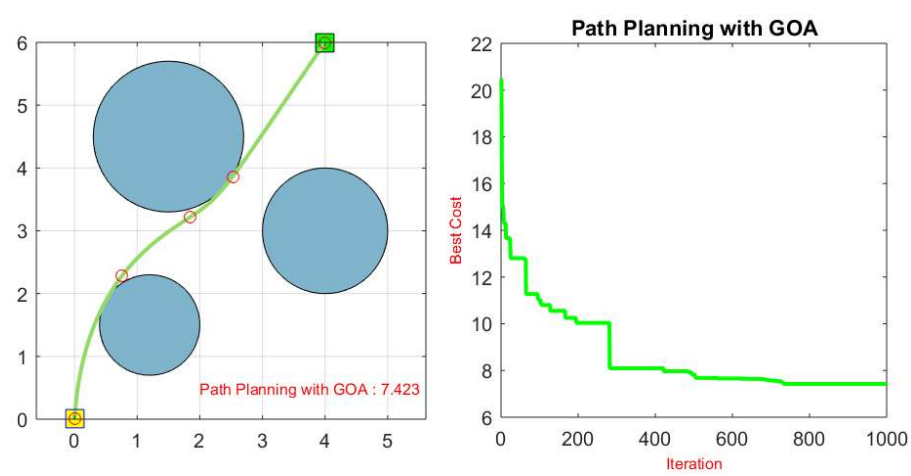
(K)



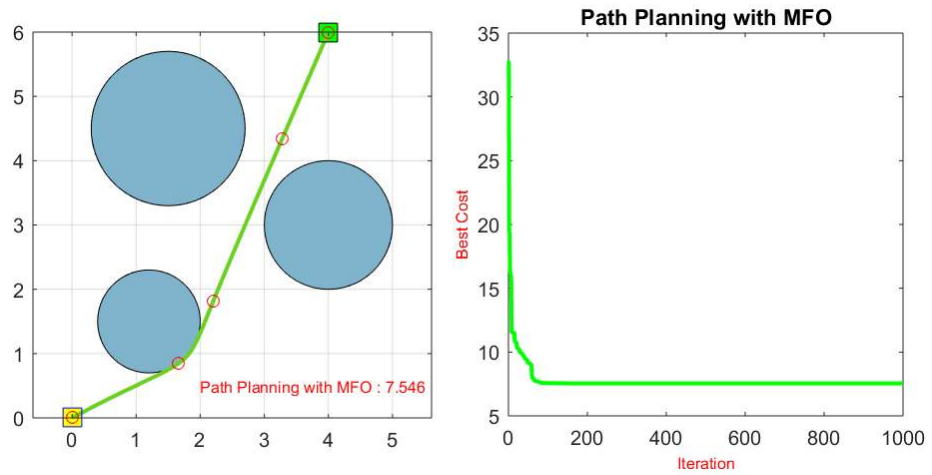
(L)



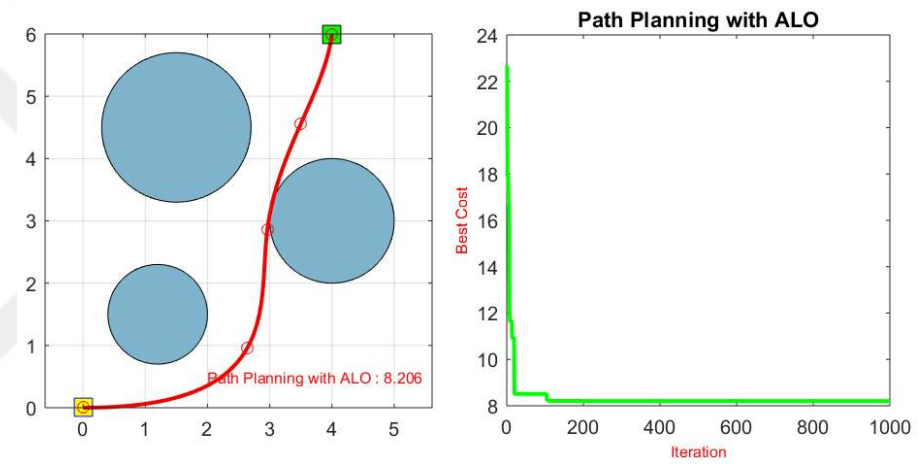
(M)



(N)

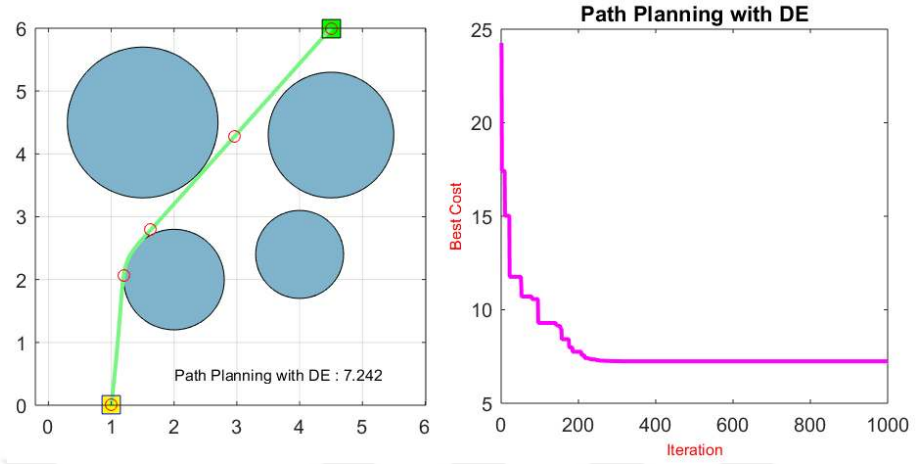


(O)

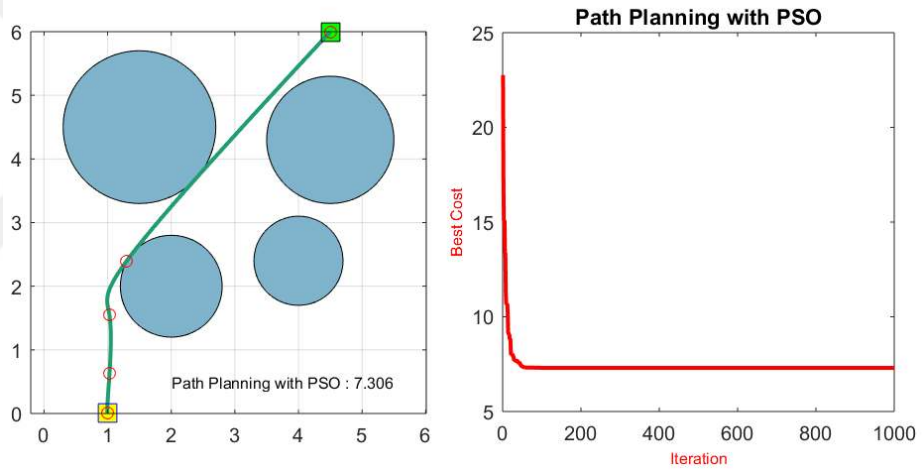


(P)

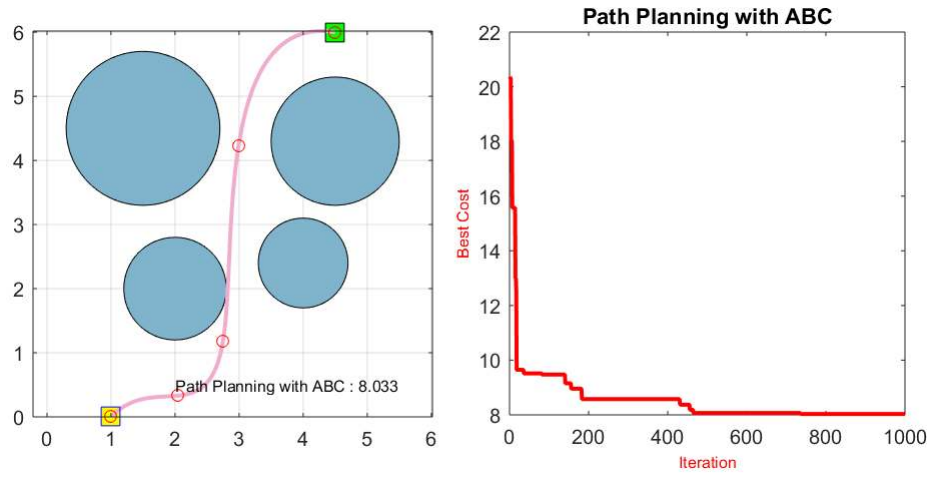
EK 2: Sezgisel algoritmaların yol planlama problemi-2 için başarımlarının karşılaştırılması (A-P).



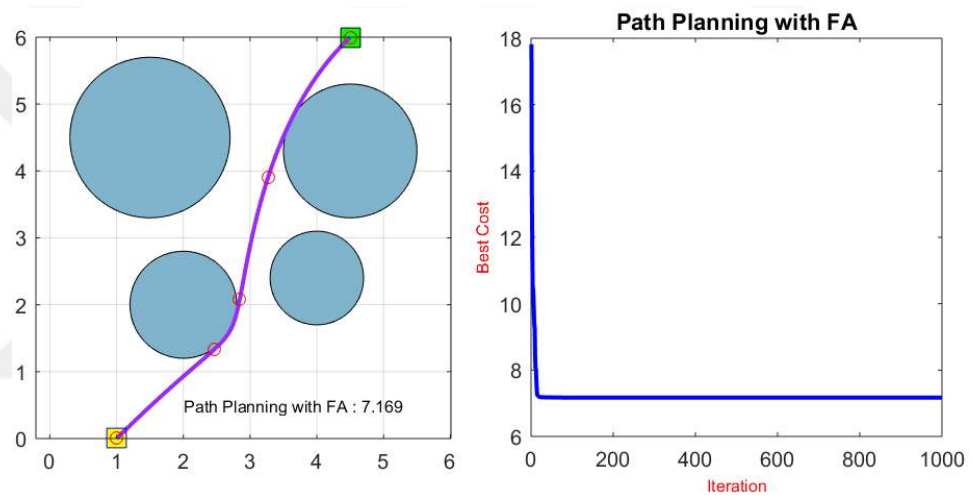
(A)



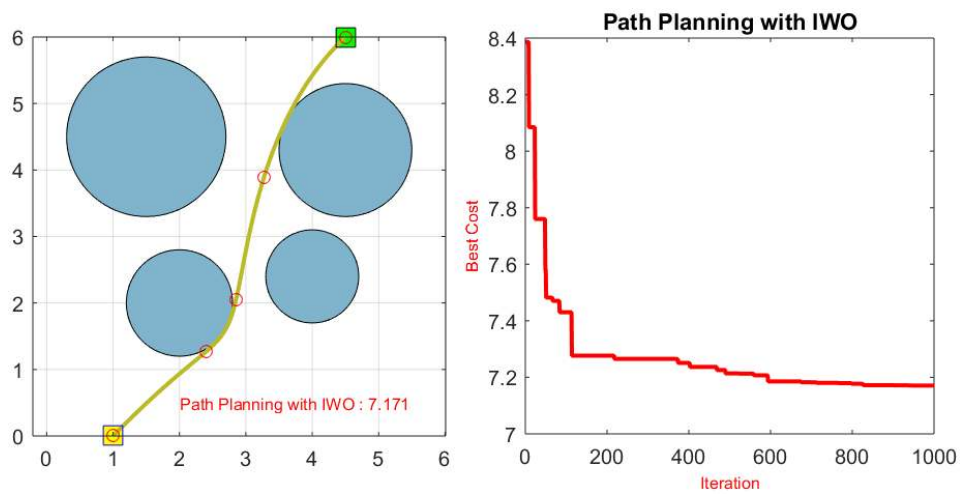
(B)



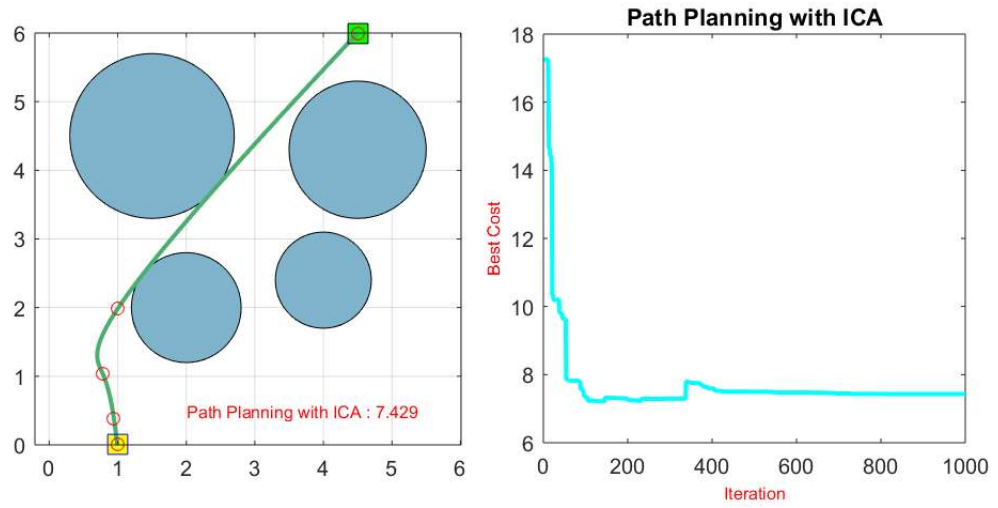
(C)



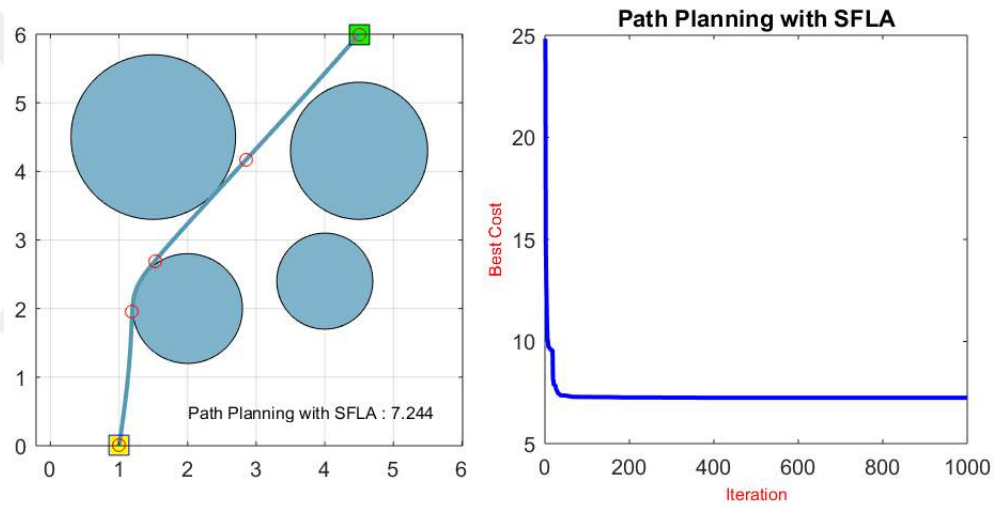
(D)



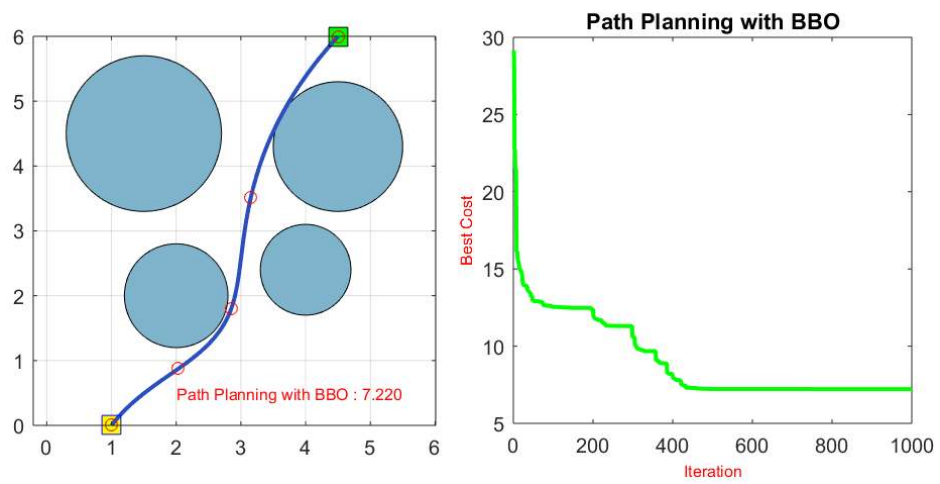
(E)



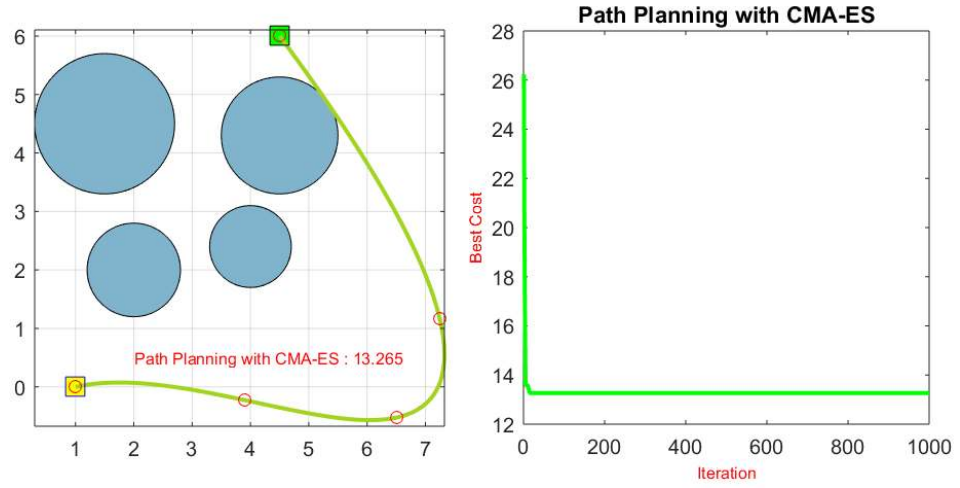
(F)



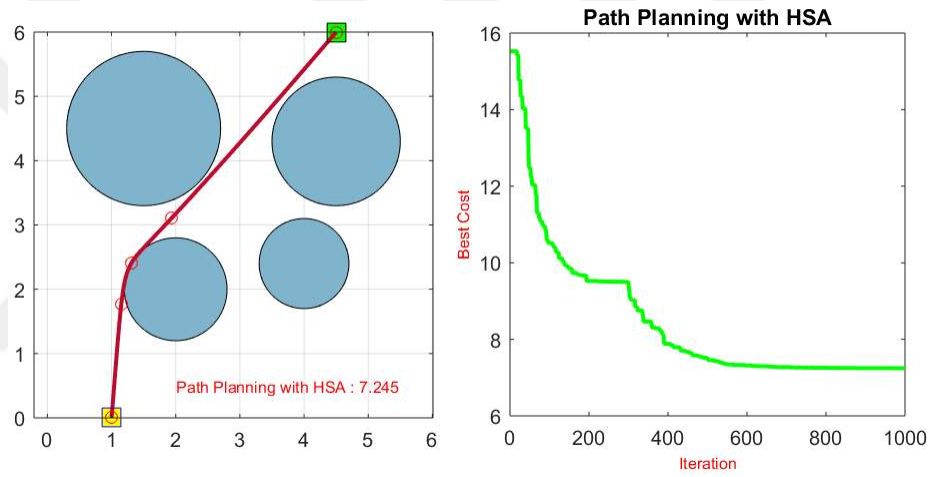
(G)



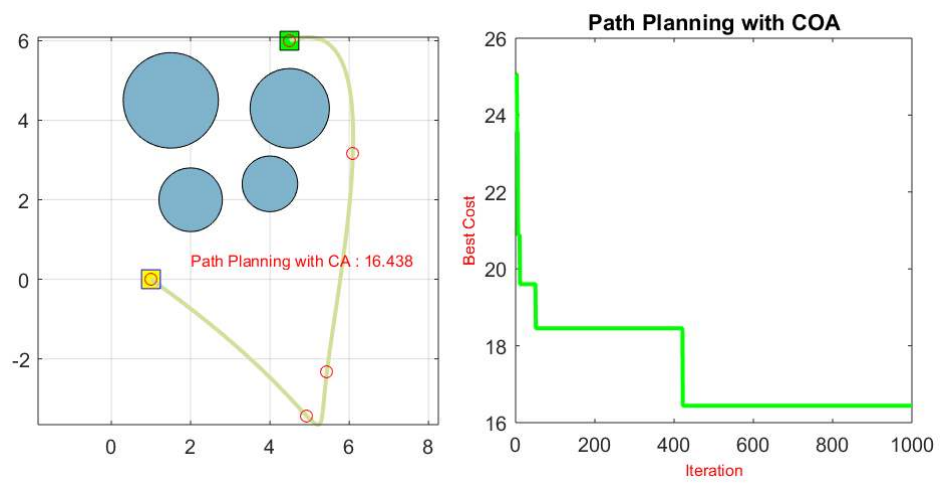
(H)



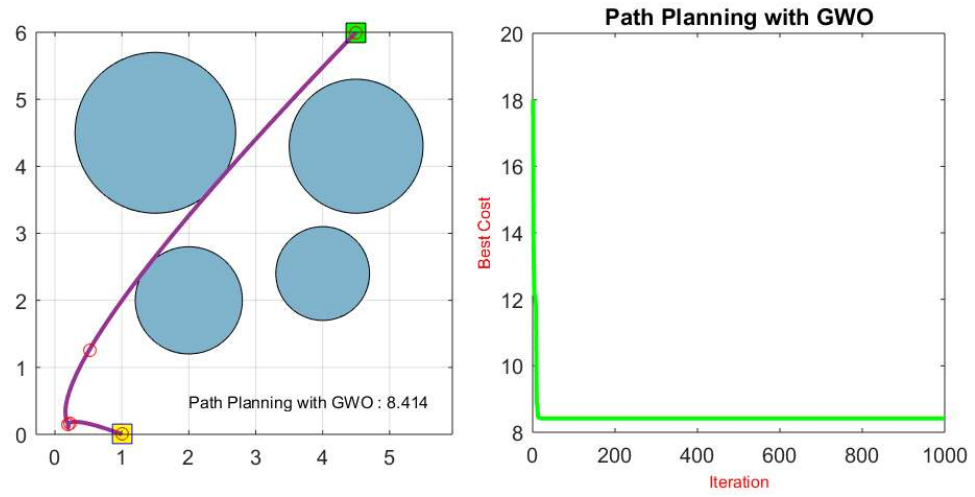
(I)



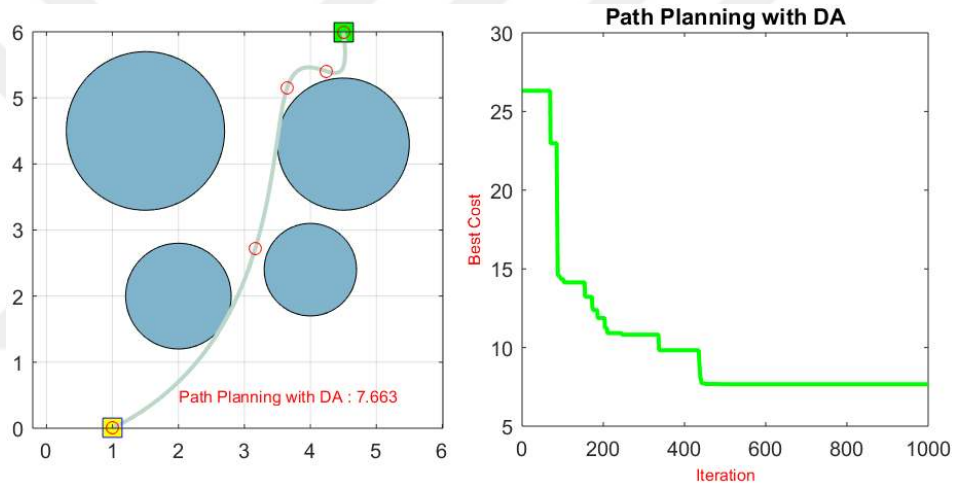
(J)



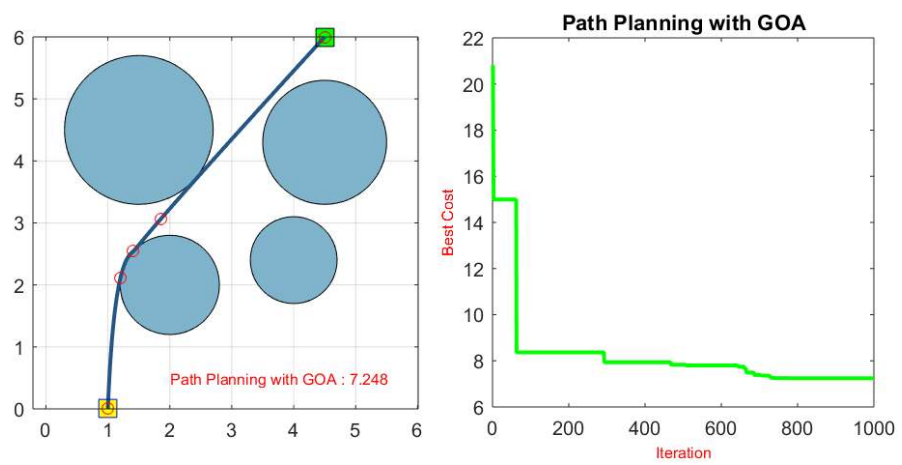
(K)



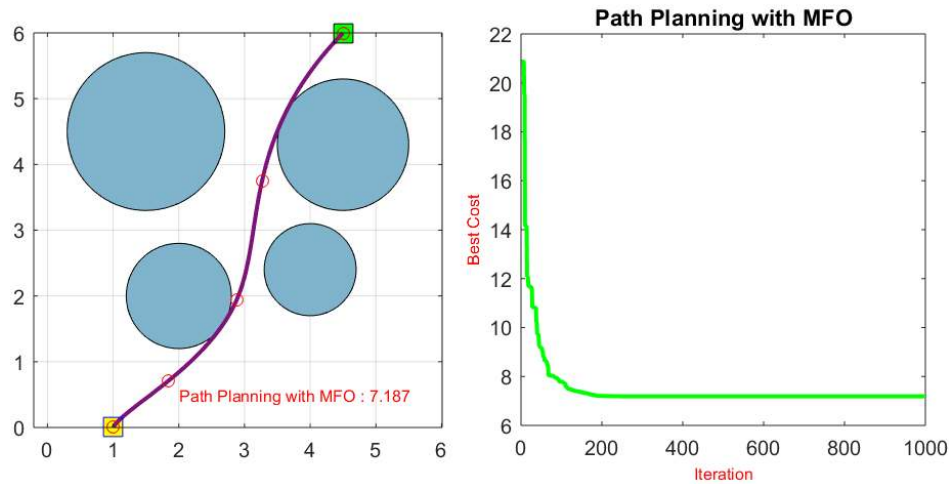
(L)



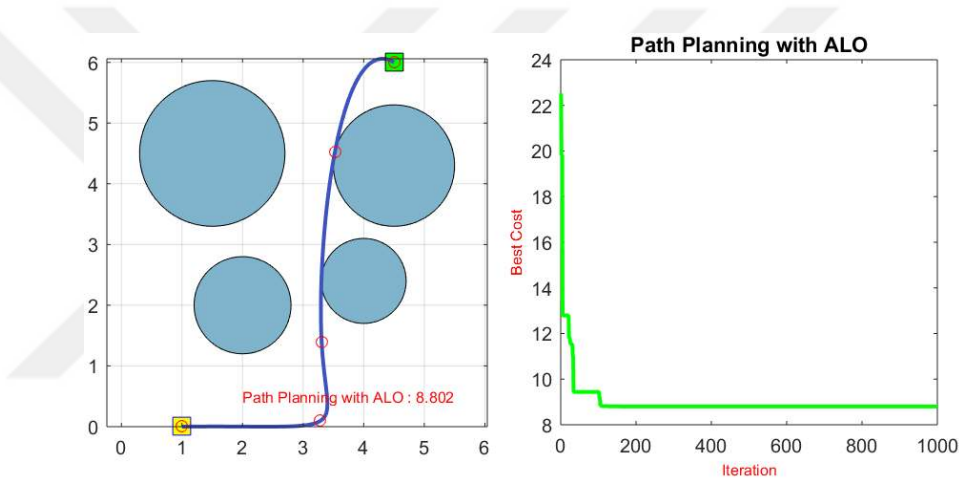
(M)



(N)

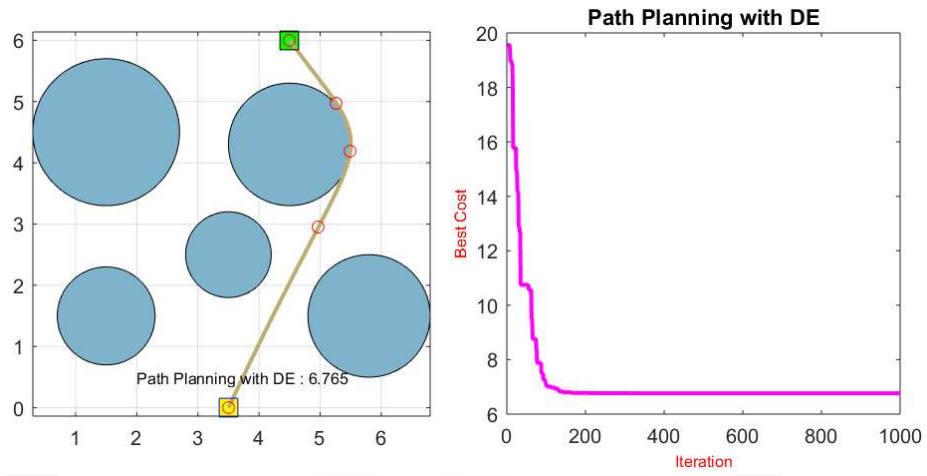


(O)

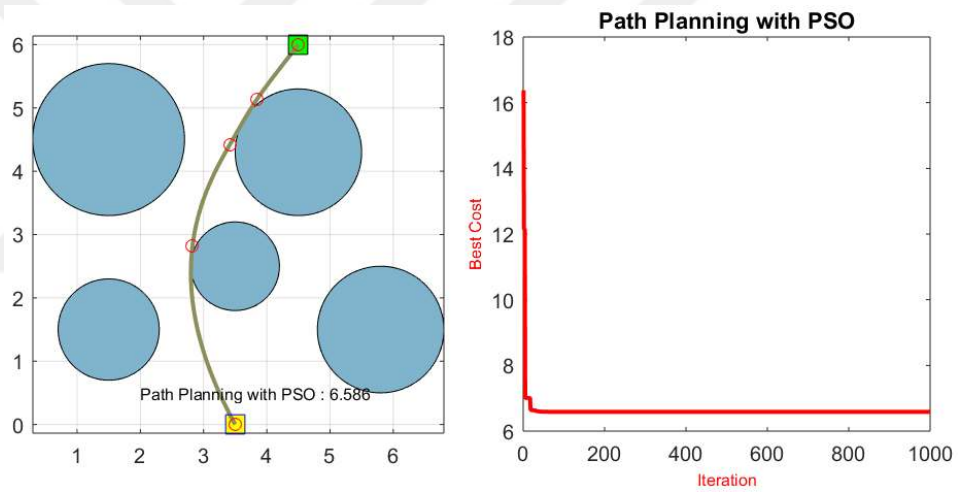


(P)

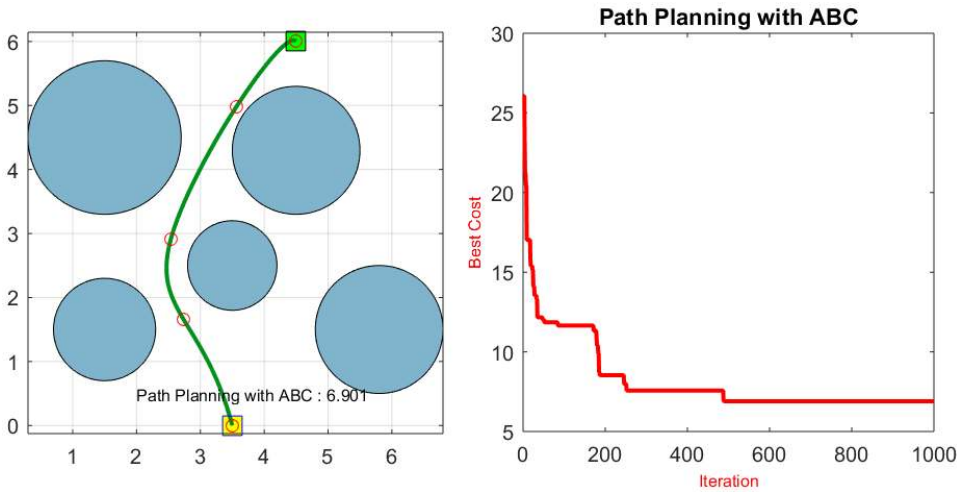
EK 3. Sezgisel algoritmaların yol planlama problemi-3 için başarımlarının karşılaştırılması (A-P).



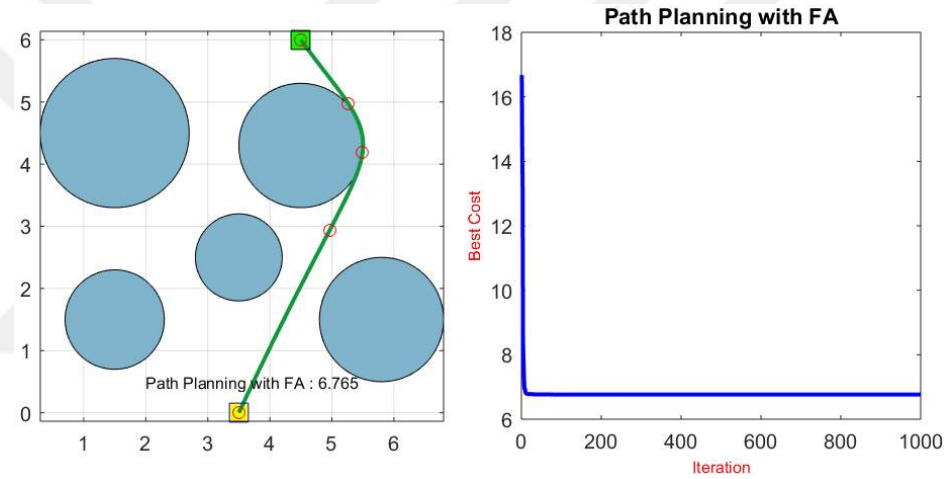
(A)



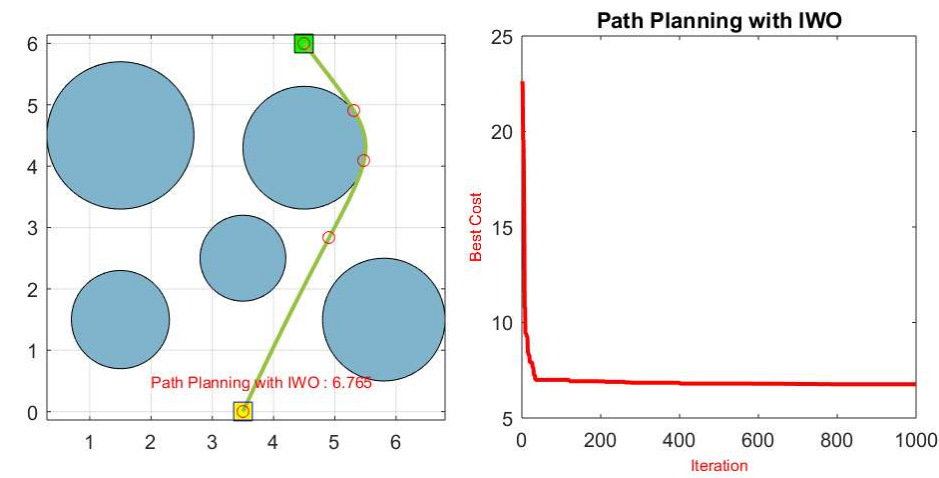
(B)



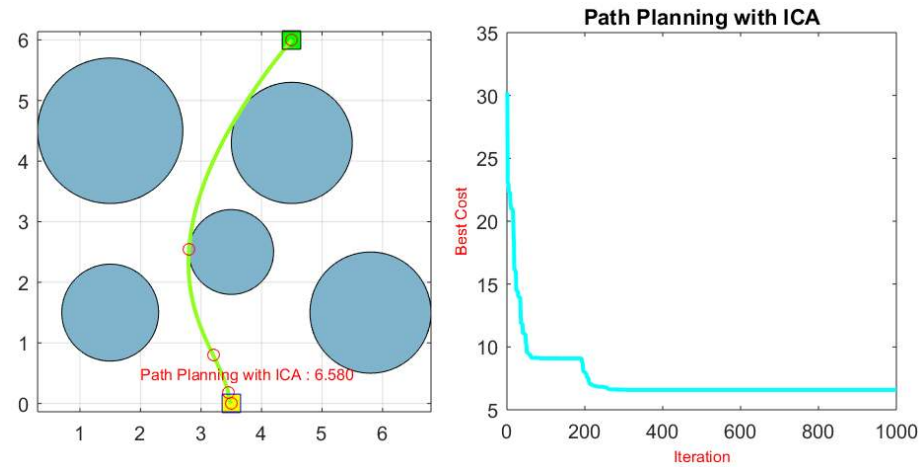
(C)



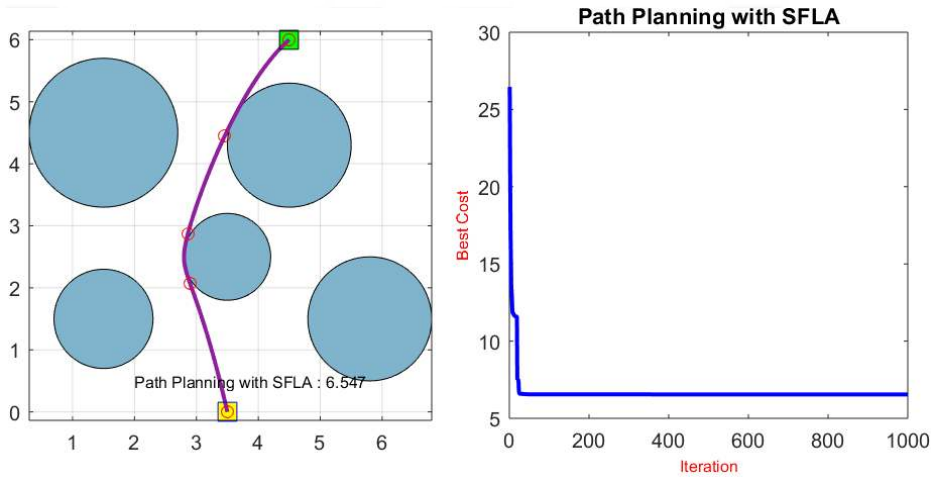
(D)



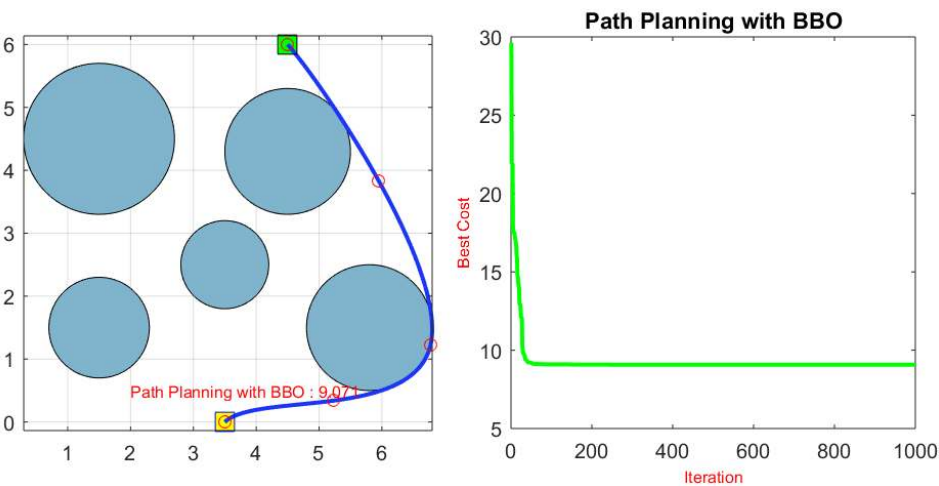
(E)



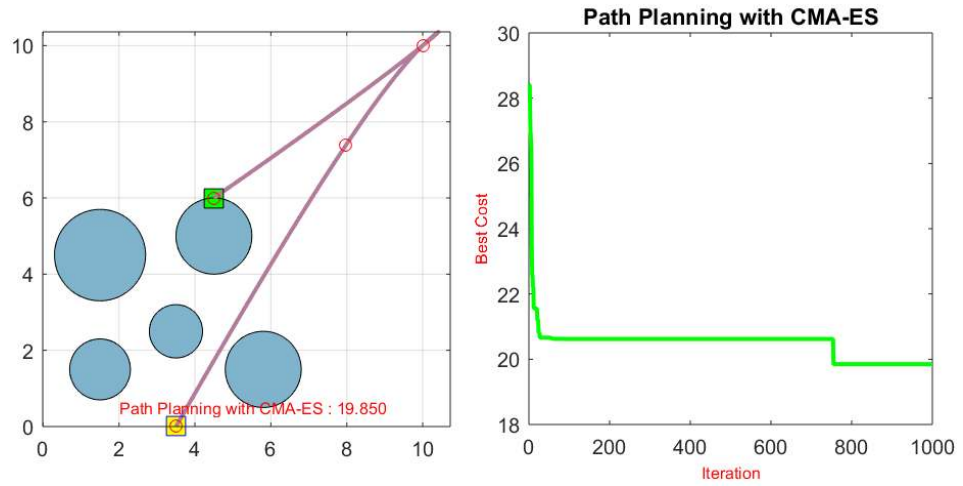
(F)



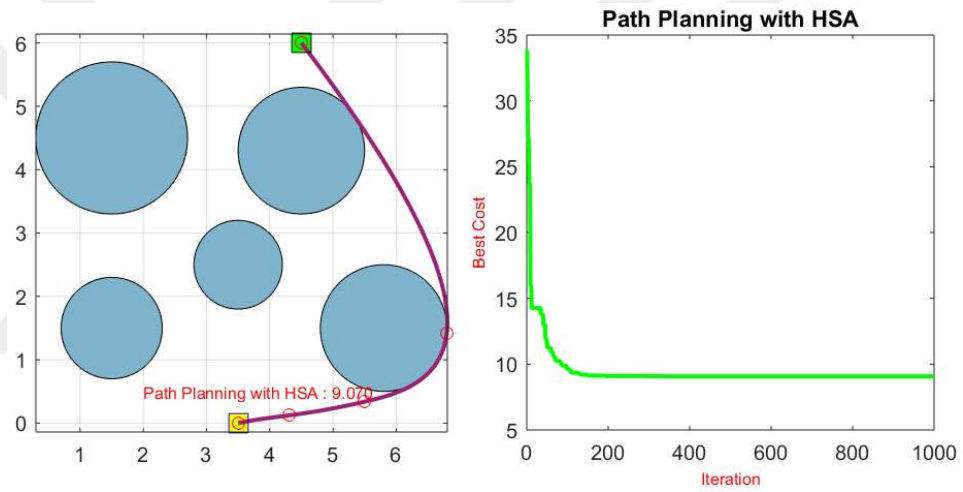
(G)



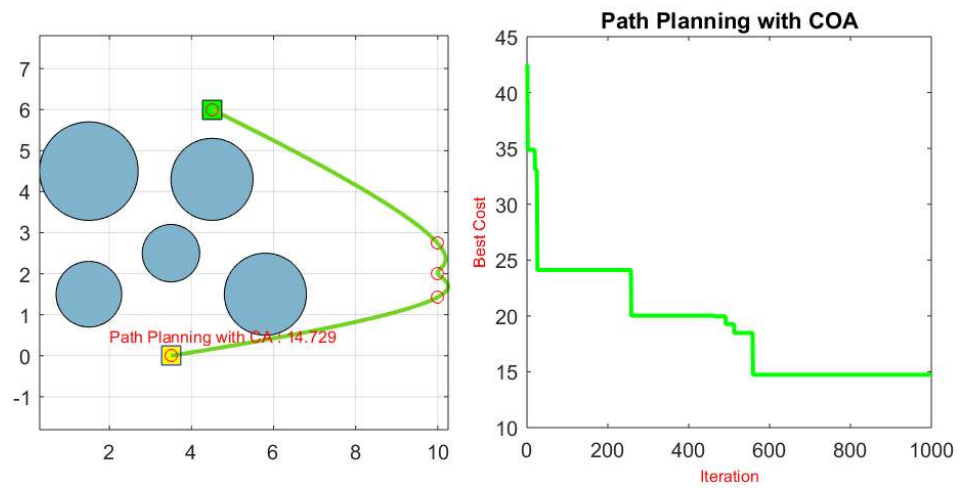
(H)



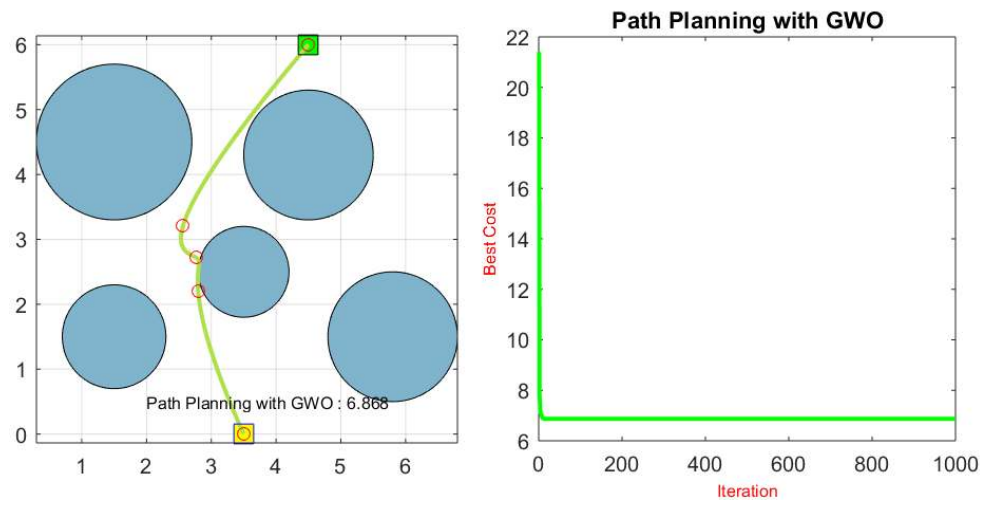
(I)



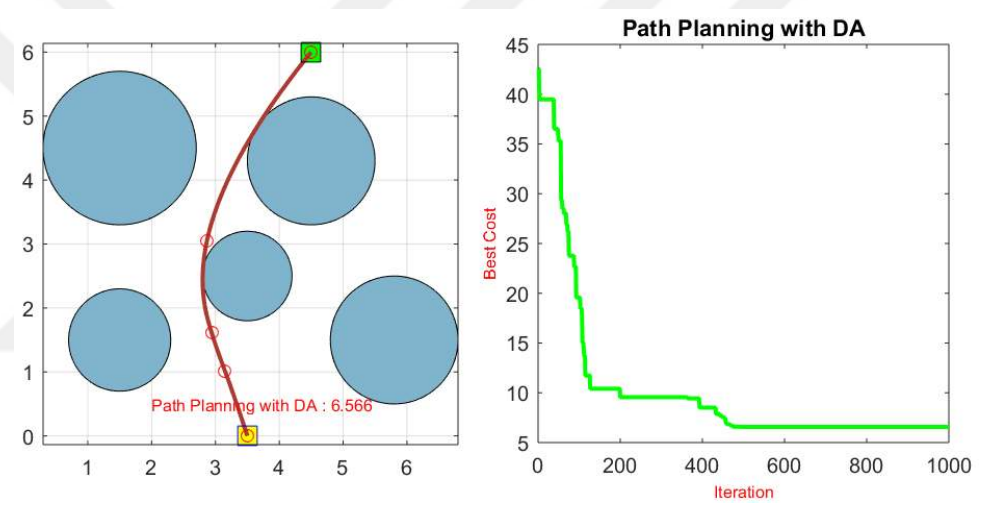
(J)



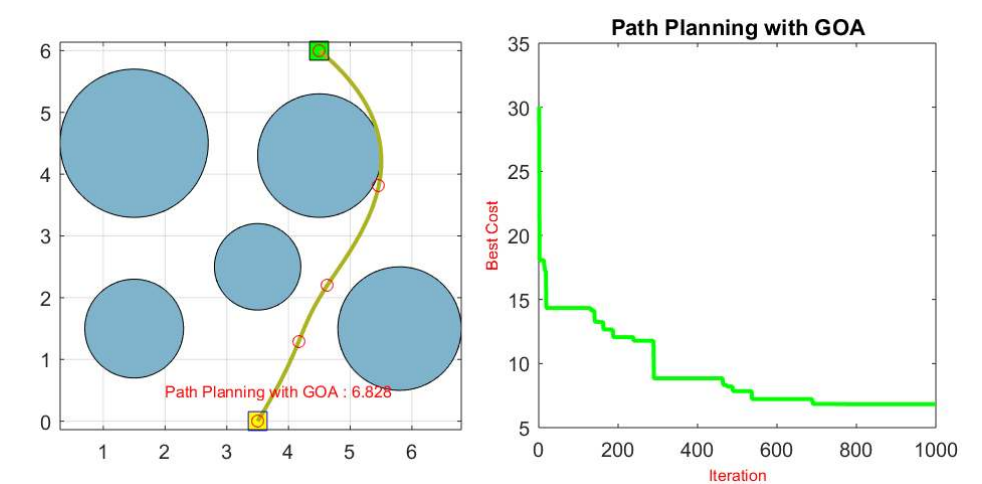
(K)



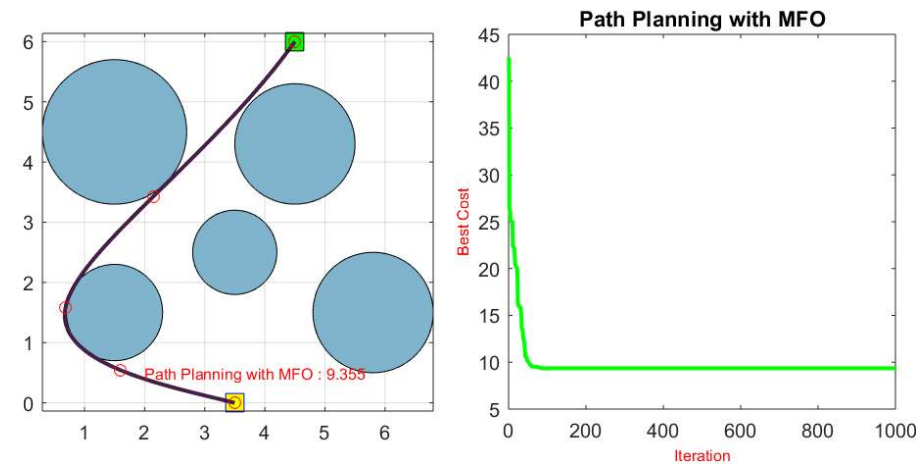
(L)



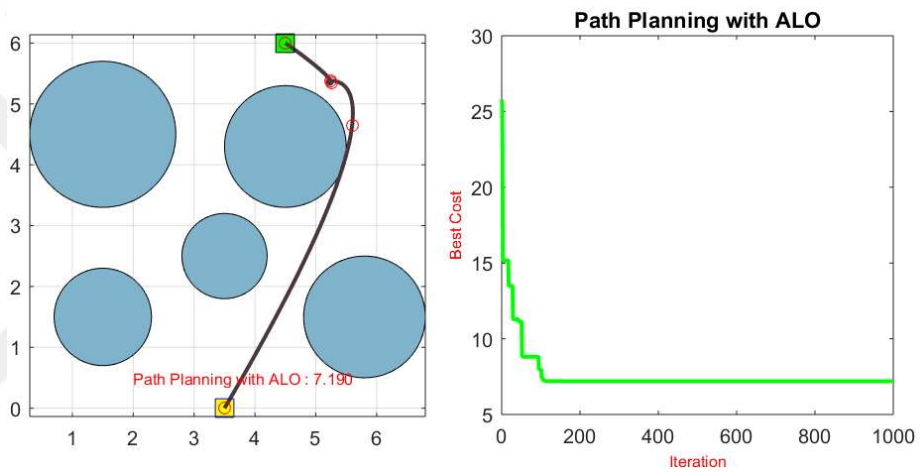
(M)



(N)



(O)



(P)

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Lokman DOĞAN
Doğum Yeri ve Tarihi : İdil, 06.10.1988



Eğitim Durumu

Lisans Öğrenimi : Bilecik Şeyh Edebali Üniversitesi, Mühendislik
Fakültesi, Bilgisayar Mühendisliği(2014)
Bildiği Yabancı Diller : İngilizce
Bilimsel Faaliyetleri :

İş Deneyimi

Stajlar : Türk Telekom(Bilecik/Türkiye)(1. Staj)
Gençlik ve Spor Bakanlığı(Ankara/Türkiye)(2. Staj)
Özel Sektör : Simternet(İstanbul/Türkiye)
T-HOS(İstanbul/Türkiye)
Projeler :
Çalıştığı Kurumlar :

İletişim

Adres : Yeni Mah. 138609. Sok. No: 12 İç Kapı No: 1
Kulu/ KONYA
Tel : 0 544 858 55 20
E-Posta Adresi : lokmandogan34@gmail.com

Akademik Çalışmaları

- Doğan L., Yüzgeç U., "Robot Path Planning using Gray Wolf Optimizer", International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES'18), pp. 69-74, Safranbolu, May 11-13, 2018.
- Yüzgeç U., Doğan L., "Solving Path Planning Problem in Mobile Robots using Meta-Heuristic Algorithms", International Congress on Engineering and Life Sciences (ICELIS'18), pp. 611, Kastamonu, April 26-29, 2018.

Yabancı Dil Bilgisi

Yabancı Dil Sınavı : 55.00

Tarih: 01/07/2018

