

**PARÇALI HÜCRESEL GENETİK ALGORİTMA İLE İNSANSIZ HAVA
ARACI PERFORMANSINA DAYALI YOL PLANLAMA**

Ahmet GEZER

Doktora Tezi

Sivil Havacılık Anabilim Dalı

Danışman: Prof. Dr. Önder TURAN

(İkinci Danışman: Doç. Dr. Tolga BAKLACIOĞLU)

Eskişehir

Eskişehir Teknik Üniversitesi

Lisansüstü Eğitim Enstitüsü

Haziran 2022

16/06/2022

DANIřMAN ONAYI

Daniřmanlıđını yrttđm Doktora đrencisi Ahmet GEZER, PARÇALI HCRESEL GENETİK ALGORİTMA İLE İNSANSIZ HAVA ARACI PERFORMANSINA DAYALI YOL PLANLAMA bařlıklı tez alıřmasını tamamlamıřtır. Hazırlamıř olduđu tez tarafımca incelenmiř ve đrencinin tez savunma sınavına alınması bilimsel ve etik aıdan uygun grlmřtr.

Tez Daniřmanı

Prof. Dr. nder TURAN

ÖZET

PARÇALI HÜCRESEL GENETİK ALGORİTMA İLE İNSANSIZ HAVA ARACI PERFORMANSINA DAYALI YOL PLANLAMA

Ahmet GEZER

Sivil Havacılık Anabilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Haziran 2022

Danışman: Prof. Dr. Önder TURAN

(İkinci Danışman: Doç. Dr. Tolga BAKLACIOĞLU)

İnsansız Hava Araçları (İHA), endüstriyel, askerî ve ticari geniş bir uygulama alanına sahiptir. Değişken amaçlar için tasarlanmış farklı yeteneklere ve boyutlara sahip İHA'ların; planlama, yönetme ve koordinasyonunu sağlayabilmek için hatasız çalışan kapsamlı alt sistemlere ihtiyaç vardır. İHA teknolojik gelişiminin önemli bir parçası, yol planlama alanındaki iyileştirmelerden oluşmaktadır. Yol planlamada operasyonel önceliklere göre farklı tercihler yapılabilir, varış noktasına en hızlı şekilde ulaşılması veya hızdan ödün vererek havada kalma süresinin uzatılması istenebilir. Bir İHA'ya ait uçabildiği her hız ve her irtifa için; seyir, tırmanma ve alçalma fazlarına ait yakıt verileri yol planlama algoritmasında kullanılmıştır. Böylece, bir İHA için özelleştirilmiş kinematik kısıtlara uyumlu performans özellikleri temelinde ekonomik ve havada kalma süresini uzatan yollar üretilebilmiştir. Bu tez çalışmasında, Hücresel (cGA) ve Parçalı Hücresel Genetik Algoritma (scGA) önerilmiştir. Sabit başlangıç popülasyonu ve parçalı kromozom yapısına sahip aşırı korumacı yeni algoritma; optimal çözüme yüksek yakınsama hızı elde etmiş, geleneksel bir genetik algoritmaya (GA) kıyasla ortalama 5,2 kat daha yüksek uygunluk değerine sahip yollar üretebilmiştir. scGA'nın GA'ya kıyasla, başlangıç popülasyonuna göre en iyi çözümü 1,9 kat ve genel popülasyonu 5,8 kat daha iyi geliştirdiği gözlenmiştir.

Anahtar Sözcükler: Yol Planlama, Yörünge Planlama, Genetik Algoritma, Evrimsel Algoritma, İHA, Hava Aracı Performansı

ABSTRACT

PATH PLANNING BASED ON UNMANNED AERIAL VEHICLE PERFORMANCE WITH SEGMENTED CELLULAR GENETIC ALGORITHM

Ahmet GEZER

Department of Civil Aviation

Eskişehir Technical University, Institute of Graduate Programs, June 2022

Supervisor: Prof. Dr. Önder TURAN

(Co-Supervisor: Assoc. Prof. Dr. Tolga BAKLACIOĞLU)

Unmanned Aerial Vehicles (UAV) have a wide range of use on industrial, military and commercial areas. Comprehensive and error-free sub-systems are needed to provide planning, management and coordination of UAVs which are designed for variable purposes with different capabilities and sizes. An important part of UAV technological development consists of improvements in the scope of path planning. Different choices can be made in path planning according to operational priorities, it may be preferred to reach the destination as fast as possible or to increase the airtime by compromising speed. Fuel data of cruise, climb and descent phases are used in the path planning algorithm for every speed and altitude that the UAV can fly. Thus, economical and airtime-maximizing paths could be produced on the basis of performance characteristics compatible with the kinematic constraints customized for the UAV. In this thesis, Cellular (cGA) and Segmented Cellular Genetic Algorithm (scGA) are proposed. The novel overprotective algorithm which has a fixed initial population and segmented chromosome structure achieves a high convergence speed to optimal solution and can generate paths which have 5.2 times higher fitness value on average compared with a conventional Genetic Algorithm (GA). It has been observed that scGA improves the initial population in terms of the best solutions 1.9 times and the general population 5.8 times better compared with GA.

Keywords: Path Planning, Trajectory Planning, Genetic Algorithm, Evolutionary Algorithm, UAV, Aircraft Performance.

TEŞEKKÜR

Bu çalışmanın her aşamasında yardımlarını ve desteklerini esirgemeyen tez danışmanlarım Sayın Prof.Dr. Önder TURAN'a ve Sayın Doç.Dr. Tolga BAKLACIOĞLU'na, değerli katkılarından dolayı jüri üyelerine ayrı ayrı teşekkürlerimi sunarım. Sayın Prof.Dr. Hikmet KARAKOÇ, Sayın Prof.Dr. Muzaffer KAPANOĞLU ve Sayın Dr.Öğr. Üyesi İlkay ORHAN'ın öğrencilik sürecimdeki katkıları için teşekkürü bir borç bilirim. Ayrıca, beni her zaman maddi ve manevi olarak destekleyen aileme duydukları güven ve gösterdikleri sevgi için minnettarlığımı ifade etmek isterim. Motivasyona ihtiyaç duyduğum anlarda yanımda olan arkadaşlarım, iyi ki varsınız.

Ahmet GEZER

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Eskişehir Teknik Üniversitesi tarafından kullanılan “bilimsel intihal tespit programı”yla tarandığını ve hiçbir şekilde “intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Ahmet GEZER

İÇİNDEKİLER

Sayfa

BAŞLIK SAYFASI	I
JÜRİ VE ENSTİTÜ ONAYI.....	II
DANIŞMAN ONAYI	III
ÖZET	IV
ABSTRACT.....	V
TEŞEKKÜR	VI
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ.....	VII
İÇİNDEKİLER.....	VIII
TABLolar DİZİNİ.....	IX
ŞEKİLLER DİZİNİ.....	X
GÖRSELLER DİZİNİ	XI
SİMGELER VE KISALTMALAR DİZİNİ.....	XII
1. GİRİŞ	1
1.1. Motivasyon.....	4
1.2. Misyon	6
1.3. Problem Tanımı.....	7
1.4. Genel Amaç.....	10
1.4.1. Özel amaçlar	11
1.5. Tezin Özgünlüğü	12
1.6. Tezin Güncelliği.....	12
1.7. Tezin Ekonomik ve Sosyal Katkıları.....	12
1.8. Tez Organizasyonu	13
2. LİTERATÜR ARAŞTIRMASI	15
2.1. İnsansız Hava Araçları	17
2.1.1. Sınıflandırma.....	18
2.2. Yol Planlama Problemi.....	22

2.2.2. Çevresel modelleme ve çalışma alanı.....	29
2.2.2.1. Görünürlük diyagramı	29
2.2.2.2. Voronoi diyagramları	32
2.2.2.3. Hücresel ayrıştırma	34
2.2.2.4. Örnekleme tabanlı yaklaşımlar.....	38
2.2.2.5. Potansiyel Alan Teorisi	40
2.2.3. Yol planlama ve optimizasyon	42
2.2.3.1. Sezgisel (buluşsal) ve metasezgisel algoritmalar	46
2.2.3.2. Genetik Algoritma	54
2.2.3.3. Hücresel Genetik Algoritma	63
2.2.4. İşlem sonrası ve yol yumuşatma teknikleri.....	68
2.2.4.1. Dubins eğrisi.....	69
2.3. Yol Planlamada Hava Aracı Performansı	70
3. MATERYAL VE YÖNTEM.....	73
3.1. İnsansız Hava Aracı Seçimi.....	73
3.2. Çevresel Modelleme	75
3.3. Hücresel Genetik Algoritma	76
3.3.1. Algoritma	76
3.3.2. Mahalle yapısı.....	85
3.3.3. Kromozom şeması	87
3.3.4. Genetik operatörler.....	88
3.3.5. Amaç fonksiyonu	95
3.3.6. Başlangıç popülasyonu	100
3.3.7. Parametreler	102
3.4. Hava Aracı Performansı.....	103
3.4.1. Kinematik kısıtlar	107
3.5. Planlama Sonrası Yol Yumuşatma.....	108
3.6. Gerçek Zamanlı Uygulama	109
3.7. Çoklu İşlem Kullanımı.....	111
3.8. Ortama Ait Haritanın Oluşturulması	112
3.9. Üretilen Yol Planlama Yazılım Paketi	113
3.9.1. Yazılım paketinin grafik kullanıcı arayüzü	114
3.9.2. Yazılım paketinin ana işlevleri.....	117

3.10. Simülasyon Ortamı	118
4. ARAŞTIRMA VE BULGULAR.....	122
4.1. Yol Uzunluğu Optimizasyona Dayalı Yol Planlama	124
4.2. Hava Aracı Performansına Dayalı Yol Planlama	125
5. SONUÇ VE ÖNERİLER.....	159
KAYNAKÇA.....	166
ÖZGEÇMİŞ	



TABLÖLER DİZİNİ

Sayfa

Tablo 2.1. Özerklik düzeyi ölçęęi	17
Tablo 2.2. Aęırlılıęa göre sınıflandırma	18
Tablo 2.3. Menzil ve dayanıklılıęa göre sınıflandırma	19
Tablo 2.4. Azami irtifaya göre sınıflandırma	20
Tablo 2.5. Kanat yüklemesine göre sınıflandırma	20
Tablo 2.6. İHA sınıflandırması	21
Tablo 2.7. GA terimleri	60
Tablo 3.1. Aerosonde İHA özellikleri	74
Tablo 3.2. cGA(R) algoritması	78
Tablo 3.3. scGA(R) algoritması	78
Tablo 3.4. Geleneksel GA algoritması	79
Tablo 3.5. Çok süreçli cGA(R) algoritması	82
Tablo 3.6. Çok süreçli scGA(R) algoritması	83
Tablo 3.7. Çok süreçli geleneksel GA algoritması	84
Tablo 3.8. cGA algoritması seçim ve çaprazlama operatörü	92
Tablo 3.9. scGA algoritması seçim ve çaprazlama operatörü	93
Tablo 3.10. Geleneksel GA algoritması seçim ve çaprazlama operatörü	93
Tablo 3.11. Mutasyon operatörü	94
Tablo 3.12. scGA mutasyon operatörü	94
Tablo 3.13. Hava aracı performans modeline dayalı amaç fonksiyonu	99
Tablo 3.14. GA parametreleri	102
Tablo 3.15. İHA fiziksel kısıtları	103
Tablo 4.1. Test platformu özellikleri	124
Tablo 4.2. 2D ortamda L3 mahalle yapısında yol uzunluęu optimizasyonunu esas alan algoritmaların tek süreçli test sonuçları	124
Tablo 4.3. 3D ortamda L3 mahalle yapısında yol uzunluęu optimizasyonunu esas alan algoritmaların tek süreçli test sonuçları	125
Tablo 4.4. Haritalar	126
Tablo 4.5. Yol uzunluęu optimizasyonunu esas alan algoritmaların Harita-1 üzerindeki test sonuçları	128

Tablo 4.6. Yol uzunluğu optimizasyonunu esas alan algoritmaların Harita-2 üzerindeki test sonuçları	130
Tablo 4.7. Yol uzunluğu optimizasyonunu esas alan algoritmaların Harita-3 üzerindeki test sonuçları	131
Tablo 4.8. Yol uzunluğu optimizasyonunu esas alan algoritmaların Harita-4 üzerindeki test sonuçları	132
Tablo 4.9. Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-1 üzerindeki test sonuçları	134
Tablo 4.10. Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-2 üzerindeki test sonuçları	136
Tablo 4.11. Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-3 üzerindeki test sonuçları	138
Tablo 4.12. Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-4 üzerindeki test sonuçları	140
Tablo 5.1. Tüm harita ve mahalle yapılarının algoritmalara göre ortalamaları	162
Tablo 5.2. Algoritmaların geçerli ve geçersiz sonuç üretmedeki hesaplama süreleri.....	163

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. İnsansız bir hava aracının tipik bileşimi	15
Şekil 2.2. Yol planlama sisteminin İHA şeması üzerinde gösterimi	16
Şekil 2.3. İHA sınıflandırması	18
Şekil 2.4. Farklı referans noktalarına göre yükseklikler	19
Şekil 2.5. Yol planlama sınıflandırılması	26
Şekil 2.6. O (n^3) algoritması tarafından dışbükey engeller ile oluşturulmuş görünürlük haritası	30
Şekil 2.7. Çalışma alanından yapılandırma alanının oluşturulması	31
Şekil 2.8. Voronoi diyagramı	32
Şekil 2.9. Tam hücre ayrıştırma	35
Şekil 2.10. Düzenli ızgara grafikleri	36
Şekil 2.11. Küp uzayı (solda) ve Octree (sağda) algoritmaları	37
Şekil 2.12. Olasılıksal yol haritası (solda), tek tip ızgara yol haritası (sağda)	40
Şekil 2.13. Biyolojik kromozom (solda); hesaplamalı kromozom (sağda)	56
Şekil 2.14. Ebeveynler ve çocuklar	57
Şekil 2.15. Rulet tekerleği seçim yöntemi	58
Şekil 2.16. Panmiktik (a), dağıtılmış (b) ve hücresel (c) GA	66
Şekil 2.17. Dubins eğrisi	70
Şekil 3.1. İHA'nın ızgara hücreleri üzerinde asgari yarıçapta dönüşü	75
Şekil 3.2. scGA(R), cGA(R) ve GA algoritmalarının basitleştirilmiş akış şemaları	77
Şekil 3.3. Tipik olarak kullanılan mahalleler	85
Şekil 3.4. Üst üste binen mahalleler	86
Şekil 3.5. Bazı cGA mahalle yapıları: L3 ve L5	86
Şekil 3.6. L3 mahalle yapısının parçalı kromozomlarda kullanımı	87
Şekil 3.7. scGA'nın kromozom yapısı ve 3D yollar için irtifayı ayarlama	88
Şekil 3.8. Genetik operatörler	89
Şekil 3.9. (a) çift nokta çaprazlama ve (b) tek nokta mutasyon	89
Şekil 3.10. IAS, CAS, TAS ve GS ilişkisi	97
Şekil 3.11. İrtifa ayarlama işlemi öncesi çarpışmalı yol gösterimi	98

Şekil 3.12. cGA ve scGA'nın başlangıç popülasyonu ile scGA'daki kromozom (yol) parçaları.....	101
Şekil 3.13. Aerosonde İHA'ya ait tırmanma fazı yakıt karakteristiği	104
Şekil 3.14. Aerosonde İHA'ya ait alçalma fazı yakıt karakteristiği	105
Şekil 3.15. Aerosonde İHA'ya ait seyir fazı yakıt karakteristiği.....	106
Şekil 3.16. Aerosonde İHA'ya ait MATLAB Simulink diyagramı.....	106
Şekil 3.17. İHA'nın dönüş noktalarındaki hareketi ve dönüş yarıçapı gösterimi.....	107
Şekil 3.18. Yol yumuşatmada silme operatörünün uygulanması	108
Şekil 3.19. İrtifanın düzenlenmesi örneği.....	108
Şekil 3.20. İHA'nın karşılaşılan bir engel durumunda hareket örneği.....	110
Şekil 3.21. Harita ve üretilen yol.....	117
Şekil 3.22. (a) Asgari uygunluk değeri grafiği (b) Ortalama uygunluk değeri grafiği.....	118
Şekil 4.1. Yol uzunluğu optimizasyonunu esas alan algoritmaların uygunluk değerlerine göre karşılaştırması	133
Şekil 4.2. Yol uzunluğu optimizasyonunu esas alan algoritmaların hesaplama sürelerine göre karşılaştırması	133
Şekil 4.3. Hava aracı performansına dayalı algoritmaların ortalama uygunluk değerlerine göre karşılaştırması	142
Şekil 4.4. Hava aracı performansına dayalı algoritmaların geçersiz sonuç yüzdelere göre karşılaştırması	142
Şekil 4.5. Hava aracı performansına dayalı algoritmaların ortalama hesaplama sürelerine göre karşılaştırması	143
Şekil 4.6. Hava aracı performansına dayalı algoritmaların ortalama ilk geçerli yol bulma sürelerine göre karşılaştırması	143
Şekil 4.7. Hava aracı performansına dayalı algoritmaların ortalama ilk geçerli yol bulma yineleme sayısına göre karşılaştırması.....	144
Şekil 4.8. Hava aracı performansına dayalı algoritmaların ortalama en iyi ilk geçerli yol bulma sürelerine göre karşılaştırması	144
Şekil 4.9. Hava aracı performansına dayalı algoritmaların ortalama en kötü ilk geçerli yol bulma sürelerine göre karşılaştırması	145
Şekil 4.10. Hava aracı performansına dayalı algoritmaların çeşitlilik oranlarına göre karşılaştırması	145

Şekil 4.11. Hava aracı performansına dayalı algoritmaların son popülasyon ortalama uygunluk değerlerine göre karşılaştırması	146
Şekil 4.12. Yinelemelere göre ortalama en iyi çözüm uygunluk değerinin değişimi...	146
Şekil 4.13. Yinelemelere göre ortalama popülasyon uygunluk değerinin değişimi	147
Şekil 4.14. 30 yineleme sonunda başlangıç popülasyonuna göre iyileşme oranları.....	147
Şekil 4.15. Görece zor bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda scGA tarafından üretilen yolların gösterimi	148
Şekil 4.16. Görece zor bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda scGAR tarafından üretilen yolların gösterimi	149
Şekil 4.17. Görece zor bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda cGA tarafından üretilen yolların gösterimi	150
Şekil 4.18. Görece zor bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda cGAR tarafından üretilen yolların gösterimi	151
Şekil 4.19. Görece zor bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda GA tarafından üretilen yolların gösterimi	152
Şekil 4.20. Görece kolay bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda scGA tarafından üretilen yolların gösterimi	153
Şekil 4.21. Görece kolay bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda scGAR tarafından üretilen yolların gösterimi	154
Şekil 4.22. Görece kolay bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda cGA tarafından üretilen yolların gösterimi	155
Şekil 4.23. Görece kolay bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda cGAR tarafından üretilen yolların gösterimi	156

Şekil 4.24. Görece kolay bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda GA tarafından üretilen yolların gösterimi 157



GÖRSELLER DİZİNİ

Sayfa

Görsel 3.1. Aerosonde "Laima" İHA (solda), Aerosonde MK 4.7 İHA (sağda)	73
Görsel 3.2. Yol planlama arayüzü	114
Görsel 3.3. Yol planlama yazılımı harita seçme ekranı	115
Görsel 3.4. Yol planlama yazılımı algoritma ve çoklu işlem seçenekleri.....	116
Görsel 3.5. Yol planlama yazılımı algoritma parametreleri.....	116
Görsel 3.6. Yol planlama yazılımı algoritma parametreleri.....	116
Görsel 3.7. FlightGear ve GoogleEarth eşzamanlı görüntüsü.....	120
Görsel 3.8. Planlanan yolun (kırmızı) ve İHA hareket izinin (sarı) görüntülediği GoogleEarth ekran görüntüsü	120
Görsel 3.9. FlightGear Route Manager	120
Görsel 4.1. Harita-1	126
Görsel 4.2. Harita-2	127
Görsel 4.3. Harita-3	127
Görsel 4.4. Harita-4.....	128

SİMGELER VE KISALTMALAR DİZİNİ

İHA	: İnsansız Hava Aracı
SİHA	: Silahlı İnsansız Hava Aracı
İHS	: İnsansız Hava Sistemi
EA	: Evrimsel Algoritma (Evolutionary Algorithm)
GA	: Genetik Algoritma (Genetic Algorithm)
cGA	: Hücresel Genetik Algoritma (Cellular Genetic Algorithm)
scGA	: Parçalı Hücresel Genetik Algoritma (Segmented Cellular Genetic Algorithm)
OPRO	: Aşırı Korumacı Değiştirme Operatörü (Overprotective Replacement Operator)
PGA	: Paralel Genetik Algoritma (Parallel Genetic Algorithm)
3B	: 3 Boyutlu
2B	: 2 Boyutlu
GUI	: Grafik Kullanıcı Arayüzü (Graphical User Interface)
ICAO	: Uluslararası Sivil Havacılık Örgütü (International Civil Aviation Organization)
EASA	: Avrupa Havacılık Emniyeti Ajansı (European Union Aviation Safety Agency)
ATC	: Hava Trafik Kontrol (Air Traffic Control)
VTOL	: Dikey Kalkış ve İniş (Vertical Take-off And Landing)
AMSL	: Ortalama Deniz Seviyesinden Yükseklik (Above Mean Sea Level)
ASL	: Deniz Seviyesinden Yükseklik (Above Sea Level)
AGL	: Yer Seviyesinden Yükseklik (Above Ground Level)
MTOW	: Azami Kalkış Ağırlığı (Maximum Takeoff Weight)
RRT	: Hızlı-Keşfedilen Rastgele Ağaçlar (Rapidly-exploring Random Tree)
RDT	: Hızlı-Keşfedilen Sık Ağaçlar (Rapidly-exploring Dense Tree)
PRM	: Olasılıksal Yol Haritası (Probabilistic Roadmap)
APF	: Yapay Potansiyel Alan Teorisi (Artificial Potential Field Algorithm)
MILP	: Karma Tamsayılı Doğrusal Programlama (Mixed-Integer Linear Programming)
LP	: Doğrusal Programlama (Linear Programming)
ACO	: Karınca Kolonisi Optimizasyonu (Ant Colony Optimization)
ABC	: Yapay Arı Kolonisi Yaklaşımı (Artificial Bee Colony)
GOA	: Çekirge Optimizasyonu Yaklaşımı (Grasshopper Optimisation Algorithm)
DEA	: Diferansiyel Evrimsel Algoritma (Differential Evolution Algorithm)
PSO	: Particle Swarm Optimization (Parçacık Sürü Algoritması)
ANN	: Yapay Sinir Ağı (Artificial Neural Network)

- SPP : En Kısa Yol Problemi (Shortest Path Problem)
TSP : Gezgin Satıcı Problemi (Traveling Salesman Problem)
OSP : Optimal Arama Yolu (Optimal Search Path)
CPU : Merkezi İşlem Birimi (Central Processing Unit)
GPU : Grafik İşlemci Birimi (Graphics Processor Unit)



1. GİRİŞ

Sivil Havacılık Örgütü (ICAO) tarafından Uluslararası Sivil Havacılık Sözleşmesi'nin 8. Maddesinde “pilotsuz hava aracı” olarak tanımlanan İnsansız Hava Araçları (İHA) veya bir başka yaygın deyişle dronlar; uzaktan kontrol edilebilir ya da tamamen özerk bir yapıda olabilir [1].

İHA'lar, havacılık alanındaki ilerlemelere paralel olarak ortaya çıkmıştır ve gelişimini sürdürmektedir. Havacılık tarihinin önemli yapıtaşlarından olan Wright Kardeşlerin 17 Aralık 1903'te Kuzey Carolina'daki ilk uçuş denemesinden yaklaşık 14 yıl gibi kısa bir süre sonra, (1917) ilk İHA olarak kabul edilen “Ruston Proctor Aerial Target”, Archibald Low tarafından basınçlı hava kullanılarak fırlatmıştır.

İHA alanında gelişim süreci 1950'li yıllardan itibaren hızlanmıştır. Soğuk savaşın etkisiyle birçok ülke pilotsuz hareket kabiliyeti olan hava araçları üretmek için projeler oluşturmuştur. Düşük maliyet ve risk ile cazip hale gelen İHA'lara AR-GE için ayrılan kaynak her geçen gün artmıştır. Araştırma programlarının çoğalması ile ortaya çıkan teknolojik gelişim, beklentileri karşılayabilecek İHA sistemlerinin ortaya çıkmasına olanak sağlamıştır. Yeteneklerin gelişim hızı değerlendirildiğinde yakın gelecekte, hâlihazırda insanlı araçlar ile gerçekleştirilen neredeyse tüm operasyonların İHA'lar tarafından özerk yapılabileceği öngörülebilir.

Askerî alanda mücadele yöntemleri değişmektedir. Küçük gruplar halinde sürekli hareket halinde olan terör örgütleri ile mücadelede teknolojik araçlar önem kazanmıştır. Terör ile mücadelede anlık ve güvenilir bilgiye gereksinim duyulmaktadır. İHA'ların keşif yeteneği bu alanda ön plana çıkmaktadır. SİHA'lar ile askerî operasyon yetenekleri büyük oranda gelişmiştir.

Askerî alandaki gelişim, özellikle; tarım, ormancılık, kamu güvenliği ve çevresel izleme alanındaki sivil uygulamalar için de İHA sistemlerinin geliştirilmesini kolaylaştırmıştır. İHA sistemleri haritalama, hava tahmini, gözetleme, kargo dağıtımı, trafik, arama ve kurtarma operasyonları, acil durumlarda bilgi toplama, telekomünikasyon, boru hattı kontrolü, meteorolojik ölçümler, çevre izleme ve sınır kontrolü gibi çok çeşitli alanlarda kullanılmaktadır.

İHA maliyetleri düşerken verimlilik ve görev başarı oranları büyük oranda artmıştır. Son on yılda sivil ve askerî her alanda İHA kullanımı, yeri doldurulamaz bir seviyeye ulaşmıştır. İHA'lara olan ilgi artarak devam etmektedir.

Pilotsuz uçuş avantajı ile olumsuz bir senaryoda insan kaybının önlenmesi, İHA'ların en önemli avantajıdır. Daha hafif ve küçük tasarım imkânı operasyonelliği artırır ve maliyeti düşürür.

İHA'lar genel olarak üç tip ortamda kullanılmaktadır ("üç D" olarak da ifade edilir): kirli (dirty), tehlikeli (dangerous) ve sıkıcı (dull) [2]. Kirli ortamlar; biyolojik, nükleer ve özellikle kimyasal partiküllerle kirlenmiş olma ihtimali bulunan alanları ifade etmek için kullanılmaktadır. Uzun süren operasyon görevlerine sıkıcı ortam tanımlaması yapılmaktadır. Tehlikeli ortamlar ise uçaklar ve pilotlar için operasyonel riskler barındıran ortamlar şeklinde sınıflandırılmaktadır. İHA'lar, kirli ve tehlikeli görevlerde ve yinelenen faaliyetlerde insanlı uçaklara kıyasla daha uygun bir araç olarak ön plana çıkmaktadır.

Hava görevlerinde pilotun yapabileceklerinin sınırlı olması İHA'ların tercih edilmesinin önemli nedenlerinden birisidir [3]. Bir insan en fazla 9g yerçekimi kuvveti altında hayatta kalabilirken pilotsuz bir İHA, 50g gibi çok yüksek bir yerçekimi kuvveti altında görev yapabilir [1]. İHA; artan manevra kabiliyeti, daha yüksek hız, daha küçük dönüş yarıçapı ve daha büyük yatış açısı yeteneği ile daha yüksek bir performans elde edilebilir.

Genel havacılıktaki standartlara göre İHA'larda mekanik arıza ihtimalinin kabul edilebilir değerden az veya buna eşit olduğu saptanmıştır [2]. Araç içinde yüksek eğitimli bir pilot barındırma zorunluluğu olmayan İHA'lar, pilotların yetiştirilmesi ve uçuş ihtiyaçlarının karşılanması için gerekli olan kokpit, fırlatma sistemi ekipmanları, uçuş simülatörleri ve gelişmiş tesislerin olmaması gibi birçok konuda maliyet avantajı sağlamaktadır.

2019 yılında İHA pazarı 19,3 milyar dolar iken 2025 yılına kadar 45,8 milyar dolarlık bir hacme ulaşması tahmin edilmektedir ve %15,5 orana sahip bileşik yıllık büyüme oranının son yıllarda sürekli arttığı görülmüştür [3].

İHA'lar, sipariş teslimatında dünya genelinde bir kullanıma sahiptir. 30 dakika gibi kısa bir süre içinde hızlı ve güvenilir teslimat hizmeti sunan uygulamalar bulunmaktadır

[4]. Bu sektöre yönelik olarak drone pazarının 2023 yılına kadar 2,4 milyon gönderim sayısına ve 2025 yılına kadar 63,6 milyar dolar işlem hacmine ulaşması; bununla birlikte bileşik yıllık büyüme oranının %66,8 olması beklenmektedir [5].

1980'li yılların başlarında yol planlama, teknolojik yetersizlikler nedeniyle teknisyenlerce manuel yapılmıştır. Teknolojik gelişmeler ile birlikte kontrol sistemlerindeki ilerlemeler, İHA yol planlanmasına katkı sağlamıştır. İletişim teknolojisindeki yenilikler, uçuş ortamına ait bilgilerin aktarılmasını kolaylaştırmıştır.

İHA teknolojik gelişiminin önemli bir parçası, yol planlama kapsamındaki iyileştirmelerden oluşmaktadır. Her ne kadar görev esnasında pilot değiştirme olanağı bulunsa da görevi sürdürürken İHA'yı tehlikeden uzak tutmak, süreklilik arz eden karmaşık bir görevdir. Bu nedenle, İHA'ların operasyon süresince en azından birtakım görevleri özerk yerine getirmesi önem arz etmektedir.

Akıllı robotik kavramı açısından bakıldığında İHA'ların araştırma alanı, havacılık robotiği şeklinde isimlendirilebilir [6]. İHA yol planlaması, robotlar için yapılan yol planlamalar ile büyük ölçüde benzerdir. İHA'lar ve robotlar arasındaki en önemli fark, hareket yetenekleri ile ilgilidir. Sabit kanatlı bir İHA için azami dönüş, tırmanma ve alçalma açısı, azami ve asgari uçuş hızı ve uçuş irtifası, uçuş süresi, iletişim menzili ve görüntü alma mesafesi dikkate alınan kısıtlardır. Döner kanatlı İHA'lar daha az kısıta sahiptir.

Günümüzde İHA'lar, merkezi olmayan özerk stratejiler ile çok çeşitli görevleri gerçekleştirebilir. Özerkliğin gerektirdiği merkezi olmayan planlama kapasitesi, İHA üretiminde çeşitli zorluklara sebep olmaktadır. İHA'larda uçuş güvenliğinin garanti edilebilmesi ancak yeterli donanım ve yazılım ile sağlanan üst düzey planlama ve navigasyon yeteneği ile mümkün olabilir.

Uçuş hazırlık sürecinin bir parçası olan yol planlama genel olarak uçuş öncesi tamamlanır ancak şartlar gerektirdiğinde uçuş esnasında yeniden yapılır. Görev sürecinde meydana gelecek değişimler gerçek zamanlı ve çevrim içi yol planlamayı da bir zaruret haline getirmektedir. Yerleşik sistemler kullanılarak 3B gerçek zamanlı yol planlama yapılabileceğine yönelik çalışmalar bulunmaktadır [7]. Ancak bir yer istasyonu kadar işlem yeteneğine sahip olması mümkün olmayan İHA'larda yol planlama becerileri, en kritik sorunlardan biri olmaktadır.

İHA'ların özerk bir yapıya sahip olmalarıyla birlikte, otonom yeteneklerde zaman ve enerji tasarrufu gibi birçok konuda verimliliğin artırılması için çalışılmaktadır. Yüksek hızla hareket eden İHA'ların seyir esnasında yeniden rotalandırılmasında kaybedilen saniyeler, aracın istenmeyen bir yönde kilometrelerce ilerlemesine neden olabilir. Örneğin kullandığı en hızlı İHA'lardan biri olan GlobalHawk, 635 km/s seyir hızında ilerlerken 20 s gibi bir hesaplama süresinde 3,5 km yol alacaktır [8, 9]. Havada kalma süresi kısıtlı olan İHA'lar için en azından yakıt kaybı ile sonuçlanan bu süre, operasyon yapısına göre özellikle askerî kullanımda İHA'nın kaybedilmesi ile dahi sonuçlanabilir. Bu tez çalışmasının amacı yol planlamayı hızlandırırken yakıt optimizasyonu ile havada kalma süresini uzatan yollar üreten bir algoritma geliştirmektir. Çözüm hızı, yol planlama algoritmalarının önceliklerindendir.

Bu tezde; İHA'ların bir noktadan başka bir noktaya gitme sürecini, hava aracı performans modeli temelinde en verimli şekilde planlama ile ilgili çalışmalar yapılmıştır. Birçok alan ve disiplinle bağlantılı olan bu konu, tez kapsamında üç aşamada çalışılmıştır:

- Arazinin tüm kısıtlar ile birlikte modellenmesi ve engellerin tanımlanması.
- Optimal çözüme hızla yakınsayan yollar üretecek bir algoritma tasarlanması.
- Geliştirilen algoritmanın, hava aracının performans modeli temelinde uçuş verilerini kullanacak şekilde yeniden tasarlanması.
- Yeryüzü yükseltisi ve engellerin yoğun olduğu riskli ortamları da içeren farklı operasyon senaryolarında testler yapılarak yöntem başarısının araştırılması.
- Simülasyon ortamında uygulanması.

1.1. Motivasyon

Çağımızdaki hızlı teknolojik gelişim, hayatı kolaylaştırdığı ölçüde her alanında etkisini hissettirmektedir. İHA'lar, bunun en belirgin örneklerindendir. Modern insansız sistemler üzerine yapılan araştırmalar ile çok sayıda tez ve makale kaleme alınmıştır. Tam veya yarı otonom bir insansız sistem tasarımı ancak disiplinler arası bir çalışma ile mümkün olmaktadır. Yol planlama, havacılık teknolojileri ile bilişim teknolojilerinin kesiştiği bir araştırma alanıdır.

İHA sisteminin otonom bileşenleri, tanımlanan görevleri nihayete erdirmek için süreç içinde stratejik kararlar alır. İHA'ların, insan operatörler yönetiminde daha doğru kararlar ile daha iyi bir performans ortaya koyduğu görülmektedir. Bu durum tam otonom

sistemlerin istenen performansı henüz gösterememesi nedeniyledir. Teknolojik gelişim süreci yeterli seviyeye ulaştığında tam otonom uygulamalar ideal bir tercih olacaktır.

İHA sistemleri birçok yetenekle donatılmıştır. Birbiri ile uyumlu entegre sistemler üzerinden insan operatörü, yer istasyonu ve iletişim bağlantılarını içeren bir ağa ihtiyaç duyulmaktadır. Örneğin Global Hawk ve Predator üç operatörden meydana gelen bir uçuş ekibiyle çalışmaktadır. Global Hawk'ın fırlatma ve kurtarma bölümü için bir sensör ve bir pilot operatörü, görev kontrol bölümü için de bir pilot operatörü bulunmaktadır [10, 11].

Değişken koşullar altında hedef belirleme ve yol planlama görevlerini gerçekleştirirken eksik özerklik, operatörün iş yükünü oldukça artırmaktadır. İnsan etkileşimini en aza indirebilmek için bu sistemlere özerklik kazandırılması verimliliğin artmasında önemlidir.

Savunma sanayisinin tüm dünyada gelişmesiyle birlikte İHA'lar, askerî görevleri yüksek başarıyla gerçekleştirebilmeleri nedeniyle birçok ülkede ilgi odağı olmuştur. Türk Silahlı Kuvvetleri; sınır içi ve dışı operasyonlar, arama kurtarma faaliyetleri ve terörle mücadele kapsamında İHA'ları aktif bir şekilde kullanmaktadır. İçişleri Bakanlığının da çeşitli birimleri aracılığıyla çeşitli operasyonlarda İHA'ları aktif saha görevlerinde kullandığı bilinmektedir. Ülkemizde İHA alanındaki çalışmalar ile her geçen gün bu alandaki uzmanlık düzeyi artmaktadır. Savunma sanayiinde yerli üretimin artmasıyla önemli bir ithalat kalemi silinmiş, ithal İHA'ların yerini yerli ve milli imkânlarla üretilen ve ithal muadillerine göre daha başarılı Bayraktar TB2, Akıncı, Anka ve Aksungur gibi İHA sistemleri almıştır. Yurt dışına ihraç edilen İHA sayısı her geçen gün artmaktadır.

İHA'lar, ülkemiz için askerî alanlardaki kullanımının dışında sivil alanlarda da büyük bir potansiyele sahiptir. Ülkemizin, geniş orman arazileri nedeniyle yangın riskiyle ve uzun kıyı şeridi nedeniyle kaçakçılık ve düzensiz göçmen geçişleriyle mücadele edilmesi gibi zorlu görev alanlarında İHA'lar, düşük maliyetli ve başarılı çözümler sunmaktadır.

Sabit sensörleri aracılığıyla İHA'lar; orman yangınları, tarım arazileri, ülke ve sınır güvenliği, jeolojik araştırma, bilimsel veriler ve askerî gözetleme gibi düşük irtifa ve yüksek çözünürlüklü uygulamalarda yaygın bir kullanıma sahiptir. Sivil ve askerî alanlardaki İHA uygulamaları; bilgi işlem, algılama, görüntüleme ve pil teknolojisindeki

gelişmelere motivasyon kaynağı olmuştur. Bu kapsamda yol planlama yazılımı; İHA'nın, tüm operasyonlarda başvuru ve operasyon başarısına doğrudan etki eden bir bölümdür.

1.2. Misyon

Otonom sistemlerin yürüttüğü görevlerin çoğunluğunu keşif, gözetleme (ISR) ve istihbarat faaliyetleri oluşturmaktadır. Bir hedef bölgeyi arama, izleme, gözetleme gibi görevler, İHA'lar tarafından icra edilmektedir. Sabit (statik) veya hareketli engeller içeren (dinamik) bir ortamda İHA'ların uçuş güvenliği sağlanmalıdır. Bir İHA açısından güvenlik, yeryüzü yükseltilerine ve hareket halindeki engellere çarpmadan, kinematik kısıtların dışına çıkmadan sabit bir rotada olduğu durum olarak ifade edilebilir [12]. Güvenlik; mümkün olabilecek tüm çevresel değişikliklerde hem engellerden kaçınma hem de planlama ve kontrol sistemlerinin yeterli performansı göstermesi şeklinde de tanımlanabilir.

Sivil harekâtlarda, hava koşullarıyla beraber hedef arazinin coğrafyası da önem arz etmektedir. Askerî harekâtlarda düşman radar pozisyonları ve hava unsurları gibi yüz yüze gelinebilecek tehlikeler de dikkate alınmalıdır. İHA'lar, görev esnasında düşman saldırısına, engele veya diğer hava araçlarıyla çarpışma riskine maruz kalabilir. Risklerin artması veya bir operasyon değişkeninin farklılaşması nedeniyle yol planlamasının yeniden yapılması önem arz etmektedir. Yeni bir yolun hesaplama süresi ve hesaplanan yolların kullanım maliyeti asgari düzeyde tutulmalıdır.

İHA'lar; sınırlı bir enerjiye sahiptir. Bu enerji hareket organları, kontrol sistemleri, veri işleme ve diğer görev odaklı araçlar için dikkatli kullanılmalıdır. Görev süresi; görev planlaması ve yol planlaması kapsamında hesaplama, algılama ve hareket bölümlerinin güç tüketiminden etkilenmektedir. Operasyon sürelerini uzatmak için yerleşik güç kapasitesinin artırılması önerilebilir ancak bu durum İHA'ların üretim ve kullanım maliyetini önemli ölçüde artıracaktır. Bu nedenle İHA'ların operasyon süresini artırabilmek için mevcut gücün en verimli şekilde kullanılması amacıyla yakıt maliyeti etkin algoritmalar geliştirmek önem arz etmektedir.

Huang vd., (2005) İHA özerklik düzeyini dört seviyede açıklamışlardır [13]:

Uzaktan kontrol: Bir insan operatörün, İHA'yı sürekli olarak uzaktan kontrol etmesi şeklinde tanımlanmaktadır. İHA yönetilir ve sorumluluk tamamen insan operatöründedir.

Teleoperasyon: Bir insan operatör, sabit sensörlerden iletilen bilgileri kullanarak İHA'yı sürekli takip eder, doğrudan cihazın aktüatörlerinin kontrollerini sağlayarak veya sıralı hedefleri ayarlayarak kontrol etmektedir.

Kısmi özerklik (yarı otonom): İHA'nın bazı işlemleri, insan müdahalesi olmadan özerk olarak yapılabilir.

Özerk sistem (tam otonom): İHA'nın, verilen görevleri insan müdahalesi olmadan değişken koşullara ve çevreye uyum sağlayarak tamamlamasıdır.

Kısmi özerk ve özerk sistemlerde yol planlama algoritmasının başarısının genel sistem başarısına etkisi üst seviyededir.

İHA, sabit bir irtifada aynı hızla yol aldığı anda enerji sarfiyatının uçuş mesafesiyle orantılı olduğu gösterilebilir. Bu nedenle enerji sarfiyatını en aza indirmek için uçuş mesafesinin kısaltılması önerilir. 3D yol planlamada ise yakıt tüketimi, aracın alçalma ve yükselme manevraları ile süreden bağımsız etmenlerden büyük oranda etkilenir.

Yol planlamada operasyonel öncelikli farklı tercihler olabilir, varış noktasına en hızlı şekilde ulaşılması ya da hızdan ödün vererek havada kalma süresinin uzatılması istenebilir. Bu tez çalışmasında, optimal yolun havada kalma süresi öncelikli hesaplanmasına odaklanılmıştır.

İyi bir yol planlama yalnızca yakıttan veya zamandan tasarruf etmeyi sağlamaz aynı zamanda İHA'nın kullanıma bağlı aşınmasını ve bakım masraflarını da azaltır. Çalışmamızda önerilen algoritma ile yakıt optimizasyonu ve İHA havada kalma süresini uzatan bir yol planlama hedeflenmektedir.

1.3. Problem Tanımı

İHA'ların belirlenen hedeflere en az düzeyde insan müdahalesi ile varması istenir. Bir tehdit algısında operatör müdahalesine ihtiyaç duyulur [2]. Operatör müdahalesi olmadan akıllı karar verme yeteneğine sahip otonom İHA'larda özerk karar alma işlevlerini gerçekleştiren katmanlar dört gruba ayrılabilir: hata toleranslı yedeklilik yönetimi, yörünge oluşturma, yol planlaması ve karar verme [14]. Bu tez çalışmasının odak noktası bu işlev katmanlarından yol planlamadır.

İHA’larda uçuş yolu, uçuшта takip ettiğı noktalar dizisi olarak tanımlanabilir. Noktalar sürekli. Her nokta, yola bir defa eklenir. Bu noktalar için dönüş noktaları [15], ara noktalar [15] ve ilgi noktaları [16] gibi isimler de kullanılmıştır.

Yol planlama problemleri aslında optimizasyon problemlerinin bir çeşidi olarak düşünülebilir. Her çeşit İHA'nın yol planlamaya ihtiyacı bulunmaktadır. Algoritmalar benzer olabilir buna karşın kinematik kısıtlar dikkate alınmalıdır. Teknik özellik kaynaklı kısıtlar varsa da İHA’larda ortak kısıt, enerji kaynağı sınırlılığıdır.

Bir operasyonda uçuş yolları, ilk hareket anından operasyonun tamamlandığı ana kadar iyi hesaplanmalıdır. İHA'ların sahip olduğu sınırlı kaynaklar, engellerin aşılması ve operasyonun belirlenen sürede tamamlanabilmesi için en uygun şekilde kullanılmalıdır. Operasyonun başarısı buna bağlıdır. Hangi görev icra edilmekte olursa olsun İHA yol planlama için kullanılacak yol planlama algoritması, hızlı ve maliyet etkin olmalıdır.

İHA'ların, yol planlama ile statik tehditler olarak;

- yeryüzü yükseltilerinden,
- uçuşa yasak alanlardan,
- meteorolojik nedenlerle uçuşa elverişli olmayan alanlardan,
- askerî operasyonlarda düşman radarlarından,

sakınması beklenmektedir.

Mobil tehditler, (diğer uçaklar, İHA’lar, füzeler, balonlar, kuş sürüleri vb.) yol planlamanın bir boyutudur. Çoğunlukla ilk planlamada mobil tehditlerin pozisyonu öngörülemez. Uçuş esnasında yapılan ve yeniden planlama olarak da ifade edilen yol planlamalarda dikkate alınmaktadır.

Üretilen yollar, İHA’ya özgü manevra kabiliyetine uyumlu olmalıdır. Tez çalışması kapsamında önerilen yol planlama algoritmasının başarısı, aşağıdaki kriterlere göre değerlendirilir:

- İHA’nın kinematik kısıtlarının dikkate alınması.
- İki nokta arasında yol planlamanın yerel optimuma takılmadan yapılması.
- İHA’ya özgü uçuş performansına göre yakıt sarfiyatını en aza indiren havada kalma süresini uzatma öncelikli yolun seçilmesi.
- Hesaplama süresinin kısa olması.

Sabit kanatlı İHA'lar, itki sonucu dolaylı olarak üretilen kaldırma kuvveti ile uçmaktadır. Döner kanat İHA'larda ise kanatların dönmesiyle bir kaldırma kuvveti oluşturulur [17]. Döner kanatlı İHA'ların yaygın örneği quadrotor, dört rotor tarafından hareket ettirilen bir İHA'dır. Farklı rotorların hızının değişmesi, quadrotor'u 3D ekseninde hareket ettirir [18]. Hareket esnekliği sayesinde sıradan bir araçla ulaşılması zor alanlarda rahatlıkla kullanılabilir. Quadrotor; dikey kalkış kabiliyeti, yüksek manevra kabiliyeti ve uygulama kolaylığı avantajına sahiptir [19].

Sabit kanatlı İHA'lar, yolcu uçakları ve askerî uçaklar ile benzer yapıda hava araçlarıdır. Sabit kanatlı İHA'larda yatay ekseninde yeterli itki üretildiğinde havada kalmayı sağlayacak bir kaldırma kuvveti oluşmaktadır. Ağırlıkları ve boyutlarına göre kritik kalkış ve perdevites hızları dikkate alınmaktadır. Kalkış ve iniş için yeterli pist uzunluklarına ihtiyaç bulunmaktadır. Küçük boyutlu İHA'lara özgü çeşitli fırlatma ve iniş platformları da geliştirilmiştir. Hız ile orantılı bir dönüş yarıçapı hesaplanır. Bu durum, engel ile çarpışmamak adına dönüş için daha dolambaçlı yollar oluşturulmasını da gerektirebilir. Aerodinamik yapıları, belirli bir pozisyonda sabit kalmayı zorlaştırır. Bütün bunlara rağmen sabit kanatlı İHA'lar, daha yüksek yük taşıma kapasitesine, hız ve daha uzun menzilli operasyonlara imkân tanımaktadır.

İHA kinematik modeline uyumlu üretilen yol, düzgün ve sürekli olmalıdır. Oluşturulan yolda, kinematik kısıtlar kaynaklı kontrol hatalarını da dikkate alan, çevredeki engellere güvenli mesafeyi koruyan bir ayırma mesafesi bulunmalıdır.

Yolculuk süresini en aza indirmek ve yakıt optimizasyonu arasındaki tercih, operasyon önceliklerine göre belirlenir. Operasyonların hızla yerine getirilmesi gereken bölümleri olabileceği gibi havada kalma süresinin öncelikli olduğu bölümleri de bulunmaktadır. Bu tez çalışmasında operatöre, havada kalma süresini önceleyen bir alternatif sunulmaktadır.

Tez çalışmasında, aşağıdaki varsayımlar kabul edilmiştir:

- İHA'nın birden çok noktayı ziyaret ettiği planlamada, her iki nokta arası bir uçuş sortisi olarak değerlendirilir.
- Mobil tehditlerin tehlike oluşturduğu alanlar operatör tarafından belirlenir.
- Rüzgâr ve hava durumu değişiklikleri ihmal edilir.
- İniş ve kalkış süreçleri otonom gerçekleştirilir.

- Görevin yürütülmesi için yakıt yeterlidir.
- İHA arıza yapmamaktadır.
- İHA modeline ait kısıtlar karşılandığı süreçte, İHA'nın, yörüngeyi güvenli bir şekilde izlemesini sağlayan iç ve dış navigasyon kontrolü mevcuttur.

Çevre bir zaman diliminde ancak kısmen bilinir ve varış noktası doğrultusunda araç hareket ettikçe daha fazla ayrıntı ortaya çıkmaktadır. Neticede çevrede daha fazla ayrıntı fark edilince yolun bir çarpışma ile sonuçlanabileceği anlaşılırsa yeniden planlama yapılır.

İHA uçuşu esnasında, yeni bir engel veya bir tehdit tespit edildiği anda yeniden yol planlamasının gerçekleştirilmesi veya bu süre açısından mümkün değilse engelden en makul yönde kaçınırken yeni optimum yolun planlamasının başlatılması istenir. Ani durum tepkisi olarak milisaniyeler seviyesinde oluşturulması gereken yol uygulanır. Ancak hızlı çalışma yeteneğine sahip bir algoritma, acil üretilen bu yolu, en kısa sürede optimize edebilir. Tez kapsamında önerilen algoritma ile hesaplama süresinin kısaltılması ve bu aşamada kullanılabilmesi hedeflenmektedir.

Özetle, bu tez kapsamında önerilen algoritmalar ile, gerçek ortamdan uyarlanmış engel haritaları üzerinde, tanımlanan konfigürasyonda, klasik yol planlama probleminin, İHA kinematik özellikleri ve performans karakteristiği dikkate alınarak havada kalma süresi öncelikli gerçek zamanlı çalışmaya entegre edilebilir bir hızda çözülmesi amaçlanmaktadır.

1.4. Genel Amaç

- Bir İHA için özelleştirilmiş, performans özellikleri temelinde, kinematik kısıtlara uyumlu, ekonomik ve havada kalma süresini uzatan yollar üretilmesi.
- İHA yakıt sarfiyatının azaltılması.
- Hesaplama maliyeti ve süresinin azaltılması.
- Çoklu İHA yol planlamasında da ihtiyaç duyulan alternatif yolların üretilmesi.
- İHA azami operasyon süresi limitinin uzaması ile operasyon başarısının yükseltilmesi.
- Algoritmanın yüksek çözüm hızı ile gerçek zamanlı sistemlere entegre edilebilmesi.

- Daha az yakıt tüketimi ve karbon emisyonunun azaltılması ile çevresel katkı sağlanması.
- İHA özelliklerini dikkate alan yol planlama ile kullanıma bağlı aşınmasının ve bakım masraflarının azaltılması.

1.4.1. Özel amaçlar

- Yol planlama algoritmalarının seçimi dâhil tüm planlama süreçlerinin yönetildiği bir yol planlama test yazılımı oluşturmak. Yazılım üzerinde:
 - harita seçimine,
 - hava aracı seçimine,
 - yol noktaları seçimine,
 - algoritma parametrelerinin seçimine,
 - çoklu hedef belirlenebilmesine,
 - algoritma başarı grafiklerinin gösterimine,
 - alternatif yolların gösterimine,
 imkân sağlamak.
- Üretilen yolların doğruluğunu gösterecek simülasyon ortamı oluşturmak. Bu kapsamda simülasyon yazılımı (FlightGear) ve harita görüntüleme yazılımı (Google Earth) ile yol planlama test yazılımı arasında köprü görevi görecek ve yönetim sağlayacak bir yazılım oluşturmak.
 - İHA'nın üretilmiş yolu simülatör üzerinde takip etmesini sağlamak.
 - Harita görüntüleme yazılımı üzerinde üretilmiş yolu ve İHA'nın takip ettiği yolu ayrı ayrı göstermek.
- Yeryüzü haritalarının yüksek doğrulukla yol planlama algoritmaları tarafından kullanılmasını sağlayacak yazılımı oluşturmak.
- Oluşturulacak yazılım içerisinde algoritmaların performansını ölçmek için gerekli test ortamını hazırlamak.
- Önerilen algoritmanın performansını standart bir Genetik Algoritma'ya (GA) göre bir dizi test problemi ve performans ölçütü ile değerlendirmek.
- Önerilen yaklaşımın uygulanmasında çoklu işlem kullanımı ile paralel mekanizmanın oynadığı rolü anlamak ve analiz etmek.

1.5. Tezin Özgünlüğü

İHA’larda:

- Hücresel Genetik Algoritma (cGA),
- Parçalı Hücresel Genetik Algoritma (scGA)

yöntemleri ile aşağıdaki özellikler ilk defa bu tez çalışmasında birlikte kullanılmıştır:

- parçalı kromozom yapısı,
- sabit başlangıç popülasyonu,
- seçilmiş bir İHA’ya özel performans verileri esas alınarak yol planlama,
- aşırı korumacı değiştirme operatörü (OPRO),
- çok süreçli algoritma yapısı.

Listelenen yöntem ve özelliklerin literatürde farklı çalışmalarda rastlanan örnekleri Bölüm 2 ve 3’te ayrıntılı açıklanmıştır. Bununla birlikte sabit başlangıç popülasyonu kullanan İHA performans modeline dayalı Parçalı Hücresel Genetik Algoritma ilk defa bu tez çalışmasında kullanılmıştır.

1.6. Tezin Güncelliği

İHA’lar havacılık ve uzay teknolojilerinin yakın dönemdeki hızlı gelişimine paralel olarak geniş bir kullanım alanı ile artan ihtiyaçlar doğrultusunda bilimsel faaliyetler ivme kazanmıştır.

Birleşik Krallık, İş Enerji ve Endüstriyel Strateji Departmanı (BEIS) tarafından Düzenleyici Ufuklar Konseyi (RHC) aracılığıyla yayınlanan raporda; (2021) önümüzdeki on yıl içinde, çoklu İHA, çarpışmadan kaçınma ve otomatikleştirilmiş “güvenli en düşük maliyetli yol planlaması”nın insansız hava araçlarının özelliği olmasının beklendiği belirtilerek İHA’lardaki yeni fırsatlara dikkat çekilmiştir [20].

İHA’larda yol planlama güncelliğini koruyan bir araştırma alanıdır. Bu tez çalışması kapsamında önerilen algoritmaların, İHA yeteneklerinin geliştirilmesine katkı sağlayacağı öngörülmektedir.

1.7. Tezin Ekonomik ve Sosyal Katkıları

Önerilen algoritmalar, üretilen yolların uygulanabilir olmasına ek olarak daha uzun havada kalma süresi ve daha uzun operasyonel görevlerin gerçekleştirilmesine imkân

vermektedir. Harcanan yakıt azalmakta ve İHA menzili artmaktadır. Algoritma verimliliği ile daha düşük hesaplama ihtiyacı, yerleşik işlemci kapasitesinin daha az kullanılarak farklı hesaplama ihtiyaçlarına alan açılmasını sağlamaktadır. Daha düşük karbon emisyonu üretilmektedir. Kullanıma bağlı aşınma ve bakım masrafları azalmaktadır.

1.8. Tez Organizasyonu

Bu tez dokümanı beş bölümde yapılandırılmıştır. Bu bölüm ile birlikte ikinci bölümde, tezi mümkün olduğunca bağımsız hale getirecek temel kavramlar açıklanmaktadır. Bu bölüm, tezde yer alan ana konulara genel bir bakış sunmaktadır.

Bölüm 2, İHA'ların potansiyeli ve sınıflandırılması ile başlamaktadır. Yol planlamanın tanımı, önemi ve kriterlerin tartışılmasını içermektedir. Grafik arama algoritmaları hakkında kapsamlı bir literatür araştırması sunmaktadır. 2B ve 3B ortamlarda önerilen yol planlama algoritmalarını tartışmaktadır. Tezin ana çerçevesi dikkate alınarak sezgisel ve evrimsel hesaplama alanından algoritmalar ve teknikler ile tezin temelden ilişkili olduğu GA ve cGA, ayrıntılı olarak açıklanmaktadır. Yöntemlere ilişkin en son teknoloji anlatılmaktadır. EA'nın paralel yaklaşımları ve yol yumuşatma tekniklerine değinilmektedir. Son olarak, hava aracı performansının yol planlamada kullanılmasına ilişkin açıklamalar yer almaktadır. Yol planlama algoritmalarının iyileştirilmesi vurgulanmıştır. Bu bölüm ayrıca algoritmaların gerçek zamanlı yol planlamaya uygulanmasını da göstermektedir.

Bölüm 3, tez kapsamında kullanılacak araç ve önerilen yol planlama algoritmasını tartışmaktadır. Önerilen Hücresel Evrimsel Algoritma'ların amaç, kısıt ve özellikleri ayrıntılandırılmıştır. Ortam haritalarının oluşturulması, hava aracı kısıtlarına uygun yolların gösterimi konularını içermektedir. Uygulanan çoklu işlem teknikleri gösterilmektedir. Testlerde kullanılacak tüm argümanların ayrıntılı olarak açıklanmasına odaklanılır. Oluşturulan yazılım platformu ve sonuçların gösterimi açıklanmaktadır. 2B ve 3B ortamlarda yol planlaması için grafik kullanıcı ara yüzlerini (GUI) göstermektedir. GUI'lerin nasıl kullanılacağına ilişkin yönergeler sağlanmaktadır.

Bölüm 4'te, bu tez kapsamında yapılan test sonuçları gösterilmekte ve açıklanmaktadır.

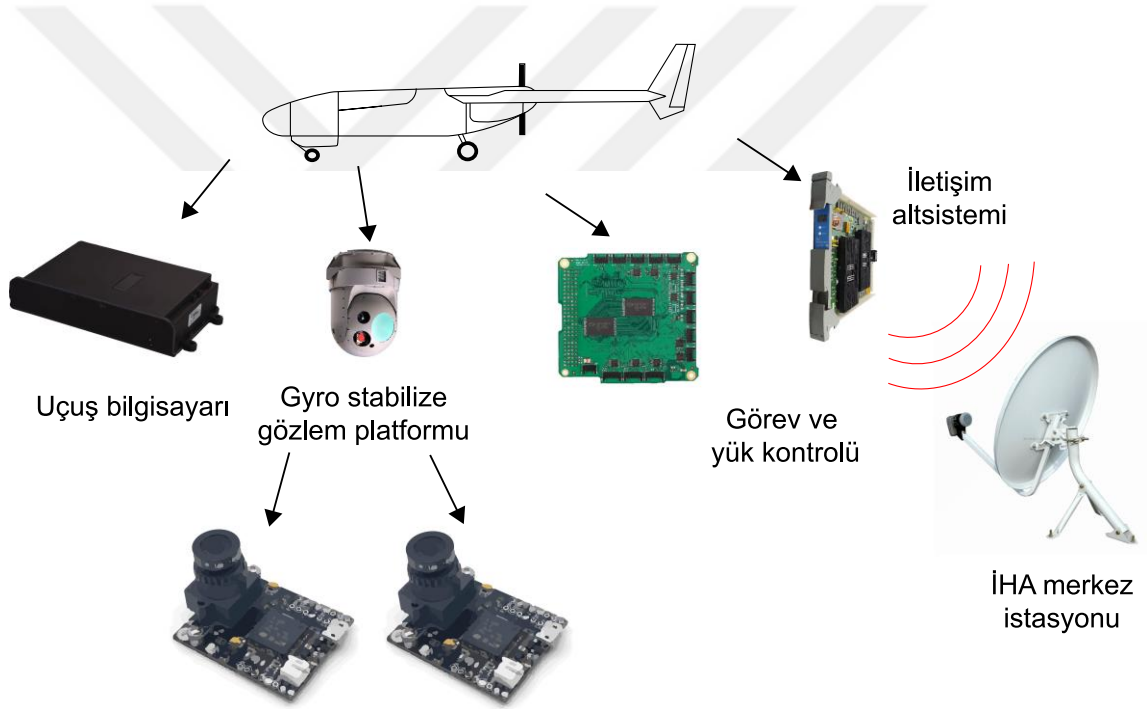
Bölüm 5'de genel sonuçlar ve tez çalışmasını genişletmek için gelecekteki muhtemel araştırma alanları sunulmaktadır.



2. LİTERATÜR ARAŞTIRMASI

Bu bölüm, tez çalışması kapsamında ilişki kurulan tüm konuların ayrıntılı bir literatür taramasını içermektedir. Öncelikle, özelleştirilmiş bir yol planlama için İHA seçimine katkı sağlayacak bir sınıflandırma yapılmaktadır. Yol planlamanın, İHA özerkliği içerisindeki yeri açıklanmaktadır.

Mevcut literatürde İHA'ların gelişimi ve kullanım alanlarına yönelik ortak bir yaklaşım olmadığı görülmektedir. Askerî ve sivil farklı alanlarda İHA çeşitliliği; bir terminoloji, kavram ve sınıflandırma ihtiyacı doğurmaktadır. İHA'ların uygulama farklılıkları ve gelişim hızı; kapsam, amaç ve kompozisyon açısından sınıflandırmasını gerektirir.



Şekil 2.1. İnsansız bir hava aracının tipik bileşimi

İHA sınıflandırmasına ve teknik özelliklerine bağlı olarak Şekil 2.1'de özetlendiği gibi İHA'lar aşağıdaki öğeleri içerebilir [21, 22]:

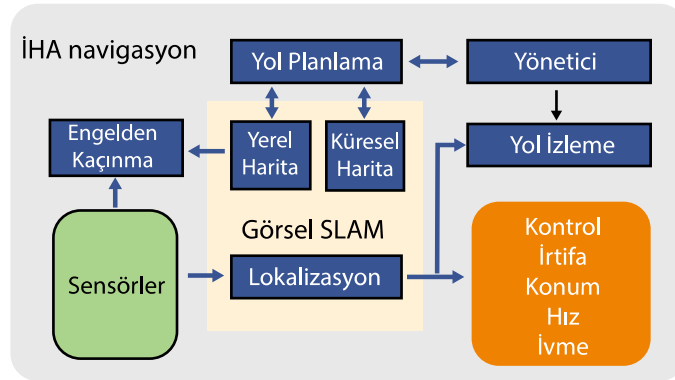
- Hava trafik kontrol ünitesi (ATC) ve iletişim araçları.
- Seyrüsefer ekipmanı
- İHA'ların yöneme bağlı kalkış ve inişini sağlayan teçhizat.
- Uçuş kontrol bilgisayarı, uçuş kontrol sistemi ve otopilot.
- Teknik durumu izlemek için ekipman.

- f) Acil bir durumda uçuşun kontrollü ve güvenli bir şekilde tamamlanmasını sağlayan bir uçuş sonlandırma sistemi.
- g) Kontrol ve izleme hattının arızalanması durumunda İHA'nın kalkış alanına dönüşünü sağlayan teçhizat.
- h) Operasyona özgü ekipman.

Bir İHA sisteminin ana bileşenleri ve faydalı yükler, verilen görevin yürütülmesi için birlikte çalışacak şekilde tasarlanır. İHA'ların bir parçası olduğu İnsansız Hava Sistemleri (İHS) ise şunlardan oluşur [22]:

- İHA/İHA'lar.
- Faydalı yükler (yerleşik sensörler, silahlar vb.).
- Komuta ve kontrol sistemleri (yer veri terminalleri, hava trafik kontrol istasyonu vb.).
- Destek ekipmanı (fırlatıcı, bakım ve kurtarma araçları vb.).
- Radyo iletişim araçları.

İHS'lerin şüphesiz en önemli bileşeni İHA'lardır. Bir İHA tasarımında; dayanıklılığın, menzilin, uçuş irtifasının, faydalı yükün ve özerklik seviyesinin artması istenir. Yol planlama sisteminin bir İHA şeması üzerinde gösterimi Şekil 2.2'de sunulmuştur [21].



Şekil 2.2. Yol planlama sisteminin İHA şeması üzerinde gösterimi

Otonom özellikler, çok farklı alanlarda çalışan sistemler için söz konusudur. Karar verme yetenekleri farklı algoritmalar ile geliştirilmiş çoğu sistem, aynı kapsamda değerlendirilebilir. Yaygın kullanılan sınıflandırma, sistemin kullanım alanlarına ve görevlerde sahip olduğu özerklik düzeyine göre yapılır. İHA'larda özerklik de bu şekilde sınıflandırılabilir.

İHA’larda özerklik, mevcut veya sensörlerden elde edilen bilgilere göre bağımsız karar verme ve uygulama yeteneği olarak tanımlanabilir [23]. Tablo 2.1’de gösterildiği gibi Ulusal Standartlar ve Teknoloji Enstitüsü (NIST)’nün İnsansız Sistemler için Özerklik Düzeyleri (ALFUS) çerçevesinde tanımlanan İHA özerklik seviyelerine göre gerçek zamanlı yol planlaması, düşük özerklik seviyesinde dahi bulunması gereken bir yetenektir [13].

Tablo 2.1. Özerklik düzeyi ölçeği

Seviye	Seviye tanımlayıcısı
10	Tam otonom
9	Sürü farkındalığı ve grup kararı
8	Durumsal farkındalık
7	Gerçek zamanlı ortak görev planlaması
6	Dinamik görev planlaması
5	Gerçek zamanlı iş birliğine dayalı navigasyon ve yol planlaması
4	Gerçek zamanlı engel ve olay tespiti ve yol planlaması
3	Hata veya olay uyarlama
2	Harici bağımsız navigasyon
1	Otomatik uçuş kontrolü

İHA’larda yol planlama, robotlardaki ile büyük oranda benzeşmektedir. En önemli farklılık, İHA’ların karakteristiklerinin yer robotlarından farklılığıdır. Sabit kanatlı (fixed wing) İHA’larda azami dönüş, tırmanma ve alçalma açısı, azami ve asgari uçuş hızı ve irtifası, uçuş süresi, iletişim menzili ve görüntü alabilme mesafesi gibi kısıtların değerlendirilmesi gerekmektedir. Döner kanatlı (Vertical Take-Off and Landing-VTOL) araçlarda ise daha az kısıt bulunmaktadır.

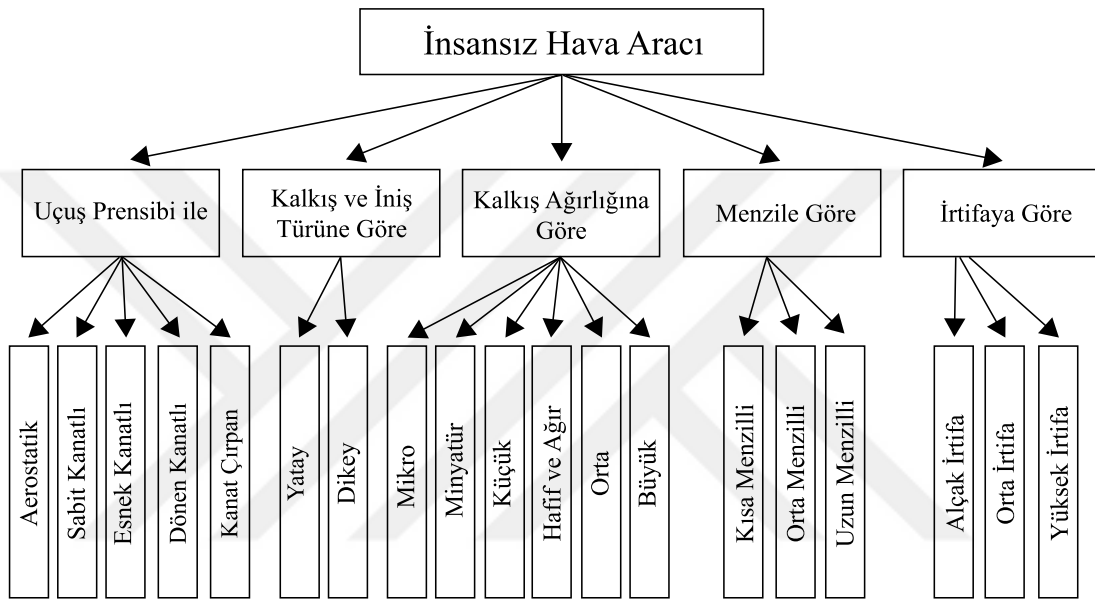
2.1. İnsansız Hava Araçları

Farklı görev alanları ve performans ihtiyaçları, çok sayıda İHA türünün ortaya çıkmasına neden olmuştur. Her ihtiyaca göre bir İHA bulunabilir ancak en doğru seçimi yapabilmek, ayrıntılı bir sınıflandırma ile mümkün olabilir. Bir yol planlama algoritması farklı türde İHA’lar için kullanılabilmekte ise de performans söz konusu olduğunda algoritmalarda araca özgü farklılıklar gereklidir. Genel olarak İHA’ların temel manada

performans ve görev alanı şeklinde sınıflandırmasında; ağırlık, dayanıklılık, irtifa, menzil, faydalı yük kapasitesi gibi özellikler belirleyici olmaktadır [24].

2.1.1. Sınıflandırma

İHA'ları performans özelliklerine göre sınıflandırmasında; dayanıklılık, menzil, ağırlık, yükseklik, motor tipi ve boyutu ile kanat yükü dikkate alınmaktadır. Şekil 2.3'de bir İHA sınıflandırması gösterilmiştir [24].



Şekil 2.3. İHA sınıflandırması

İHA pazarı; ağırlığı gram ile ölçülen İHA'lardan, tonlarca ağırlığa sahip devasa İHA'lara kadar çok geniş bir yelpazeye sahiptir. Tablo 2.2'de gösterilen genel sınıflandırmada az sayıda 2 ton ve üzeri İHA bulunurken, çoğunun hafif olduğu görülebilir [25]. Hafif ve mikro İHA'larda elektrik motorlar yaygınken, orta ağırlıklı İHA'larda pistonlu motorlar, daha ağır İHA'larda ise turbo jetler veya turbo fan motorlar yaygın olarak kullanılmaktadır.

Tablo 2.2. Ağırlığa göre sınıflandırma

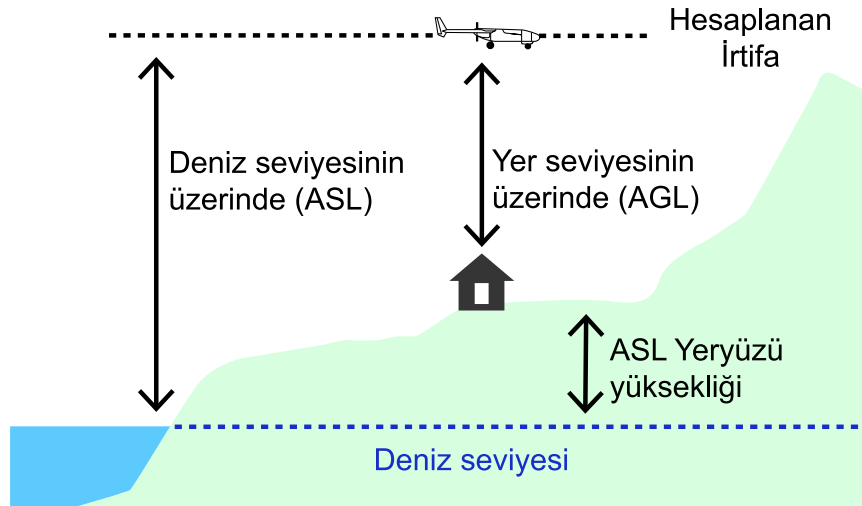
Seviye	Ağırlık	Örnek
Çok ağır	>2000 kg	Global Hawk
Ağır	200 - 2000 kg	A-160
Orta	50 - 200 kg	Raven
Hafif	5 - 50 kg	RPO Midget
Mikro	<5 kg	Dragon Eye

Bir diğ er sınıflandırma yöntemi, dayanıklılık ve menzile göre yapılan sınıflandırmadır. Menzil ve dayanıklılık arasında genel olarak pozitif bir korelasyon bulunmaktadır. Menzil ve dayanıklılığın dikkate alınması sınıflandırmada önemlidir. Tablo 2.3’de menzil ve dayanıklılığa göre bir sınıflandırma gösterilmiştir [25]. Görev tanımına göre bakılacak ilk kıstas, varış noktasının kalkış sahasından ne kadar uzakta olduğudur ki bu İHA tipinin belirleyici özelliğidir. Menzil, yakıt ikmalî seçeneğî dikkate alınmadan hesaplanır ancak her araç için kullanılabilir olmamakla birlikte dezavantajları söz konusu olsa da ikmal ile uzatılabilir.

Tablo 2.3. Menzil ve dayanıklılığa göre sınıflandırma

Kategori	Dayanıklılık	Menzil	Örnek
Yüksek	> 24 saat	> 1500km	Predator B
Orta	5 - 24 saat	100 - 400 km	Silver Fox
Düşük	< 5 saat	< 100 km	Pointer

İHA’larda çalışma irtifası, ortalama deniz seviyesine (AMSL) göre azami irtifa olarak tanımlanmaktadır. Gizliliğın askerî operasyonlar için önemli bir faktör olması nedeniyle İHA’ların askerî alanda kullanımında yüksek irtifada uçabilme yeteneğî, düşman tarafından tespit edilememesinde önemli bir avantajdır. Sivil kullanımda görüntüleme ve keşif amaçlı daha geniş arazinin kapsanabilmesi için yüksek irtifa tercih edilebilir. Burada yer seviyesi yüksekliğine (AGL) göre irtifa önemlidir. Askerî kullanımda İHA güvenliğini sağlamak için en azından küçük silah menziline ötesinde uçulmalıdır. Buna ek olarak, faydalı yük yeteneklerinin kullanılabileceğî aralığa uygun AGL irtifa kritik bir öneme sahiptir (Bkz. Şekil 2.4) [26].



Şekil 2.4. Farklı referans noktalarına göre yükseklikler

Azami operasyonel irtifa veya uçuş tavanı, bir performans ölçütü olarak önemli bir yer tutmaktadır. Faydalı yük ağırlığı uçuş tavanını etkilemektedir. Tablo 2.4'te azami irtifaya göre sınıflandırma gösterilmiştir [25].

Tablo 2.4. Azami irtifaya göre sınıflandırma

Kategori	Azami İrtifa	Örnek
Düşük	<1000 m	Pointer
Orta	1000 - 10000 m	Finder
Yüksek	>10000 m	Darkstar

İHA'ları sınıflandırmada kanat yükleme yetenekleri, sınıflandırmada kullanılabilecek bir başka yöntemdir. İHA'nın kanat yükü hesaplanırken; İHA'nın toplam ağırlığı, kanat alanına bölünür. Tablo 2.5'te kanat yüklemesine göre sınıflandırma gösterilmiştir [25].

Tablo 2.5. Kanat yüklemesine göre sınıflandırma

Kategori	Kanat yükleme (kg/m ²)	Örnek
Düşük	<50	Seeker
Orta	50 - 100	X -45
Yüksek	>100	Global Hawk

İHA'ların genel gövde tipleri sabit ve döner kanatlardır. Kalkış ve iniş yöntemleri açısından çeşitli kullanımlar mevcuttur: otonom pist iniş-kalkış moduna ek olarak; havadan bırakılma, el ile fırlatma, araba veya ray fırlatması, elastik kablo (bungee)/şişirici tabanlı (canister) fırlatma rampası gibi farklı yöntemler kullanılmaktadır.

İHA'lar kapsamlarına göre farklı motor konfigürasyonunda tasarlanırlar. İHA'larda bulunan motor tiplerinden bazıları turbofan, çift vuruşlu (twostrike), pistonlu, döner (rotary), turboprop, it ve çek (push and pull) ve elektrikli motorlardır.

İHA platformları ve uçuşlarına yönelik gerekli düzenlemelerin hazırlanması amacıyla sivil uygulamalar için uluslararası bir standart oluşturmamış olsa da NATO ve EuroControl, 1999 yılında iş birliğine başlamıştır. Buna ek olarak, İHA sistemlerinin yaygınlaşması ve ticari büyüme sonucu birçok uluslararası kuruluş, İHA'ların operasyonel güvenliğini analiz etmektedir. Avrupa Havacılık Güvenliği Ajansı (EASA) tarafından ulusal sertifikasyon kurallarına tabi İHA'ların azami kalkış ağırlığı, (MTOW) 150 kg'dan az ve 150 kg'dan fazla olacak şekilde bir sınıflandırma yapılmıştır. Avrupa

İHS Topluluğuna Genel Bakış [27] raporunda verilen sınıflandırma, Tablo 2.6’da gösterilmiştir.

Tablo 2.6. İHA sınıflandırması

UAS Kategorileri	Kısaltma	Menzil (km)	Uçuş İrtifası (m)	Dayanıklılık (saat)	Azami kalkış ağırlığı (kilogram)
Taktik					
Nano	η	< 1	100	< 1	< 0,025
Mikro	μ (mikro)	< 10	250	1	< 5
Mini	Mini	< 10	150 - 300	< 2	< 30
Yakın Menzil	CR	10 - 30	3.000	2 – 4	150
Kısa Mesafe	SR	30 - 70	3.000	3 – 6	200
Orta Menzil	MR	70 - 200	5.000	6 – 10	1.250
Orta Menzilli Dayanıklılık	MRE	> 500	8.000	10 – 18	1.250
Alçak İrtifa Derin Penetrasyon	LADP	> 250	50 - 9.000	0,5 – 1	350
Alçak İrtifa Uzun Dayanıklılık	LALE	> 500	3.000	> 24	< 30
Orta İrtifa Uzun Dayanıklılık	MALE	> 500	14.000	24-48	1.500
Stratejik					
Yüksek İrtifa Uzun Dayanıklılık	HALE	> 2000	20.000	24-48	12.000
Özel amaç					
İnsansız Muharebe Hava Aracı	UCAV	~ 1500	10.000	~ 2	10.000
Öldürücü	LETH	300	4.000	3 – 4	250
Yem	DEC	0 - 500	5.000	< 4	250
Stratosferik	STRATO	> 2000	>20.000 &<30.000	> 48	UMGT
Ekzostratosferik	EKZO	UMGT	> 30.000	UMGT	UMGT
Uzay	UZAY	UMGT	UMGT	UMGT	UMGT

* UMG: Ulusal mevzuata göre tanımlanacak

Bu bölümde, mevcut İHA teknolojileri ve bunların kuvvet sistemlerine, gövde yapısına ve öngörülen misyonlara bağlı olarak nasıl farklılaştığı incelenmiştir. Sabit kanatlı İHA’lar, operasyonel gereksinimlere göre ağırlık ve boyut açısından farklılık göstermektedir. Bu tez kapsamında operasyonel kapsamda yol planlama ihtiyacı daha fazla olan alçak irtifa uzun menzilli sabit kanatlı İHA'lara odaklanılmıştır.

Sabit kanatlı İHA'lar, uçuş sırasında aerodinamik tasarlanmış bir kanat ile kaldırma kuvveti üretmekte ve kontrol yüzeylerini (irtifa dümeni, istikamet dümeni, kanatçıklar) kullanarak manevra yapmaktadır. Elektrikli veya içten yanmalı motorlar tahrik sistemi olarak kullanılabilir. İHA'lar pistten konvansiyonel kalkış yapar veya mancınık vb. yöntemler kullanılarak fırlatılabilir. Uçuştan sonra piste konvansiyonel bir şekilde veya paraşüt ve kurtarma ağları vb. yöntemler ile iniş sağlanır. Bu sistemler, multirotor İHA'lar gibi havada asılı kalamadığı için iç mekanlarda kullanılmaz. Daha yüksek uçuş süresi, menzil ve taşıma kapasitesi avantajları ile sabit kanatlı İHA'lar, dış mekân uçuş operasyonlarında otopilot ve navigasyon sensörleri ile çeşitli faydalı yükleri taşımaya uygundur.

Alçak irtifada engeller ve tehditler, yüksek irtifaya kıyasla daha fazladır. Yere yakın görevlere yönelik İHA'ların yakın operatör kontrolü yoksa otonom uçuşta alçak irtifa şartlarında güvenliğin sağlanması gerekir. Alçak irtifa İHA'ların üretim maliyetleri daha düşüktür. Arama ve kurtarma operasyonlarında alçak irtifa görüş yeteneği daha yüksektir. Alçak irtifa uçuşu, risklerine rağmen farklı açılardan avantajları nedeniyle kullanılmaya devam etmektedir.

2.2. Yol Planlama Problemi

Yol planlama problemi, literatürde yoğun çalışılmış ve güncelliğini kaybetmeyen bir konudur. Dinamik veya kinematik kısıtlara tabi bir araç için karmaşık bir ortamda yol planlama probleminin zorluğu üst seviyededir. Yolların geçerli ve uygulanabilir olması esastır ancak yol optimalliği kısıtlara göre değişmektedir. Yol planlama gelişime açık bir problem olmaya devam etmektedir.

Yol planlama, bir aracın sadece çevresine anlık tepkiler vermek yerine, önceden yapılan planlama ile daha az anlık tepki ile yol alabilmesine imkân vermektedir. Önceden bir yol planlama yapılmadan verilen tepkiler sonucu başarılı bir yol elde edilmesi genel olarak mümkün değildir. Yol planlama, belirli bir özerklik seviyesine sahip her araç için bir gerekliliktir.

Literatürde yörünge (trajectory) ve yol (path) kavramları kullanılmaktadır. Yörünge, zaman fonksiyonunu içeren yol bilgisidir [28]. Yollar, başlangıç ile varış arasındaki noktalar listesidir. Her iki nokta arasındaki hız ve ivme bilgisine sahip yay, yörünge olarak ifade edilmektedir.

Yol planlama, önceden tanımlanmış güvenlik, yakıt verimliliği gibi kriterlere göre keşif vb. görevler için muhtemel yolları değerlendirerek kabul edilebilir seyahat yolları bulmayı amaçlayan karmaşık bir problemdir. Çok sayıda aday yol içerisinde en uygun yolu belirlemenin basit planlama kısıtları altında dahi hesaplama maliyeti yüksek olabilir [29].

İdeal olarak, İHA iki nokta arasında hareket etmekte, en kısa yolu izlemekte ve tanımlanan görevleri yerine getirmektedir. Yol planlamada kapsamlı bir yaklaşım için başlangıç ve varış noktalarının yanı sıra çevre hakkında da bilgi gereklidir. İHA’larda genel olarak başlangıçtan varış noktasına kadar tam bir yol planı çıkarabilmek için gerekli çevre ve yeryüzü şekilleri gibi tüm bilgiler elde edilmektedir. Küresel bir yaklaşım şeklinde ifade edilebilecek model üç aşamaya dayanmaktadır:

- 1) İHA’nın içinde bulunacağı alan analiz edilerek uçuşa uygun olan ve olmayan bölgeleri içeren çalışma ve yapılandırma alanı belirlenir.
- 2) Yol hesaplanır.
- 3) Yol simülasyon veya gerçek ortamda uygulanır.

Yol planlama konusunda yapılan çalışmalar incelendiğinde ortam bilgisine göre bilinen ve bilinmeyen ortamlar olarak iki kategoriden bahsedilebilir. Çözümüne dair tüm bilgilere sahip olunan durumlarda başlangıçtan varış noktasına kadar tam bir planlama gerçekleştirilebilir. Ani gelişebilecek engellere karşı çarpışma önleme amaçlı gerçek zamanlı yerel çözümler üretecek mekanizmalar geliştirilir. Çözümüne dair kısıtlı bilgiye sahip olunan durumlarda ise yakın geleceğe ve çevreye dair çözüm üretilebilir. Tam olmayan bu çözümler, varış noktasına kadar olan bölgeye dair yeterli bilgiler olmadan geliştirildiği için varış noktasına ulaşılması garanti edilmemektedir.

Bir yol planlama uygulamasının ilk aşaması, çalışma alanının algoritma tarafından tanınmasıdır. Böylece, yapılandırma alanı oluşturulur. İkinci aşamada, yapılandırma alanında yapay zekâ teknikleri ile arama yapılır. Hesaba dayanan geometrik tekniklere dayalı temsil yöntemi ile en kısa yol hesaplanmaya çalışılır. Bilinen ortamlarda planlama yapmak daha kolaydır. Ortamı tanıma ve yol arama algoritmaları çok çeşitlidir. Kullanılan arama algoritmaları, gerçekten en kısa yolu değil ancak geometrik temsil ile belirlenen popülasyon içerisindeki en kısa yolu seçebilirler. Bu işlem popülasyon tabanlı bir aramadır [30].

Yol planlama üç farklı zamanda uygulanabilir:

- 1) Çevrimdışı: Uçuş başlamadan önce yol planlanıyorsa bu planlama çevrimdışı olarak adlandırılır. Çevrimdışı yol planlama, uçuştan önce, İHA yerden ayrılmadan ve genellikle kontrol edilen İHA üzerindeki bütünleşik sistemin haricinde farklı bir bilgisayarda hesaplanmaktadır. Çevrimdışı planlama ile genellikle optimal/optimale yakın yollar üretilmektedir.
- 2) Çevrimiçi: Yol planlama, uçuş esnasında yapılıyorsa çevrimiçi olarak adlandırılır.
- 3) Gerçek zamanlı: Bir engel tespit edildiğinde ya da görevde bir değişiklik olduğu anda gerçekleştirilen yol planlamasıdır.

Yollar; uydu, gözetleme veya daha farklı kanallarla elde edilen çevre verilerine dayalı oluşturulur. Bir başka deyişle, İHA'nın güzergâh üzerinde bulunan herhangi bir engeli fark etmesi, sensörler ya da İHS'den gelen bilgiler ile mümkün olmaktadır. Engel algılanırsa bilgi, yol planlayıcıya iletilir ve çarpışmasız bir yol planlanır. Yol planlayıcı, İHA'nın engel için tepki vermesine uygun bir sürede yeni yolu bulur ve güzergâhı değiştirirse, bu gerçek zamanlı olarak adlandırılır. Çevrimiçi ve gerçek zamanlı yol planlama, uçuş görevi esnasında bir tehdit ile karşılaşılırsa veya görevde bir değişiklik olursa gerçekleştirilen yol planlamadır. Zamansal açıdan planlama türleri arasında gerçek zamanlı hesaplama kaynakları oldukça sınırlıdır ve kaynakların verimli kullanımı İHA'lar için kritik öneme sahiptir.

Çalışma alanı modellemesi küçük bir haritada tam olarak yapılabilir ancak haritanın büyümesi, ayrıştırma için spesifik yöntemler kullanılmasını gerektirebilir. Yoğun hesaplama karmaşıklığı nedeniyle makul sürede çözüm elde edebilmek için kısıtlar gevşetilerek optimallikten ödün verilebilir.

Yüksek kalitede verimli çözümler elde etme potansiyeline rağmen algoritmalar, çözüm üretememe riski barındırır. Bu nedenle, bazı teknikler açgözlü bir yaklaşım benimsemektedir. Çalışma alanını daha küçük parçalara bölerek optimal bir sonuç almaya çalışan yaklaşımlar da bulunmaktadır. Bu bağlamda, yol planlama teknikleri genel olarak iki ana kategoride incelenebilir: küresel ve yerel yol planlaması [31].

Çevrimdışı hesaplamalar, küresel yöntemlerin tercih edilmesine uygundur. Boyutları ve enerji tüketimleri ne olursa olsun çok güçlü bilgisayarlar kullanılabilir. İHA

üzerindeki işlem gücünün az olması bütünleşik planlama yeteneklerini sınırlandırmaktadır. Buna karşın, çevrimdışı planlama, hesaplama sınırlarının genişliği sayesinde yüksek doğrulukta, uzun menzilli ve optimal bir yol oluşturabilir ancak değişken çevre, engel ve tehdit unsurları söz konusu olduğunda yetersiz kalmaktadır.

Yerel yol planlamada, gerçek zamanlı olarak reaktif bir yol planlama yaklaşımı ile yerleşik sensörlerden alınan bilgiler doğrultusunda daha küçük çevrede yollar oluşturulabilir [32]. Bir yöntemin zayıflığını diğerinin güçlü yönleriyle aşmak için hibrit bir yaklaşımı ile küresel ve yerel yaklaşımlar birlikte kullanılabilir [33].

Çevrimiçi planlamada, iletişim koşulları yeterli olduğunda merkezi bir planlayıcı optimal yollar elde edebilir. Böylece, kararlar tüm parametreler dikkate alınarak bütünsel bir bilgi seviyesinde gerçekleştirilir ve özellikle çoklu İHA yol planlamasında her İHA, ekip için en yararlı şekilde yönlendirilir. Sınırlı iletişim veya sınırlı hesaplama kısıtlamaları altında ise görev dağıtımı için daha ağgözlü bir kontrol stratejisi uygulamaya konulmaktadır. Bunun sonucunda her üye takımdan ziyade kendi görevine göre hareket etmektedir.

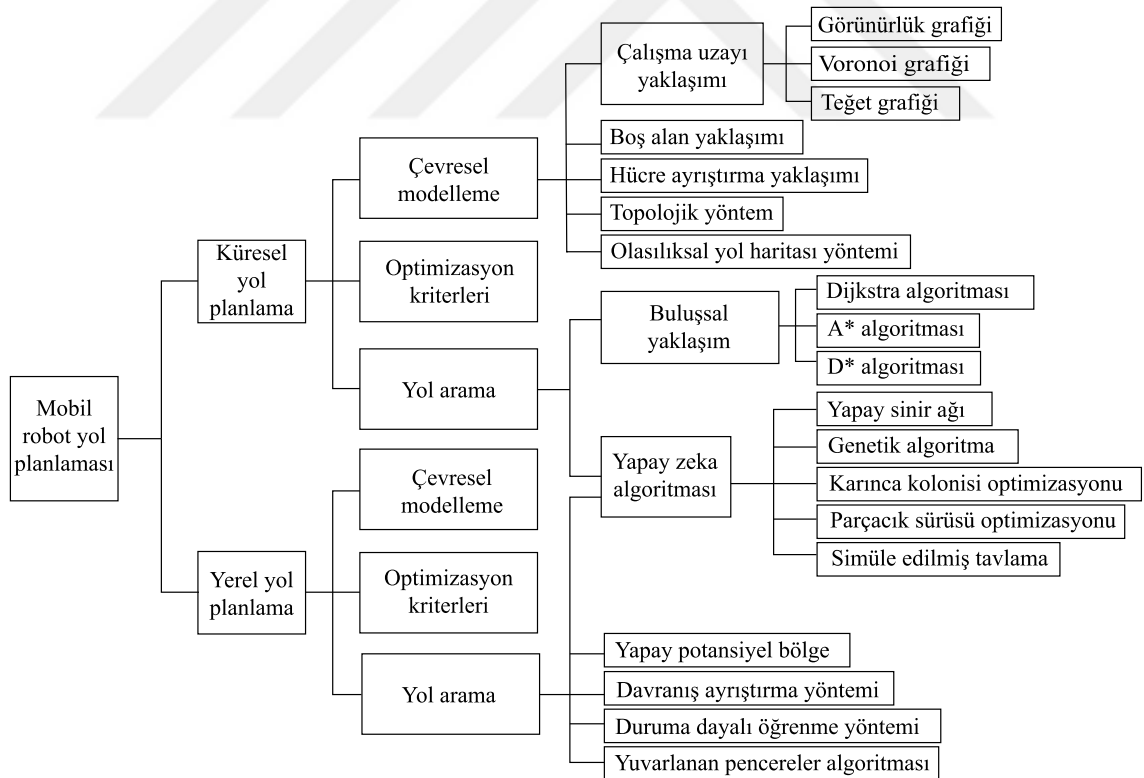
Çevrimiçi planlamada hesaplamalar, ya yüksek hesaplama gücünü karşılayabilecek yerleşik bir sistem ile ya da yer istasyonu vb. farklı bir platformdan gerçekleştirilir. Farklı bir platformun erişilebilir olması, ikinci seçeneğin ön şartıdır. Farklı platformdan yapılan bir çevrimiçi yol planlamanın uygulanmasında zorluklar bulunmaktadır. İletişimde oluşabilecek kesintiler de dikkate alınmalıdır. Planlamanın etkili olabilmesi için hesaplama hızı ve iletişim gecikmesi birlikte değerlendirilerek gerçek zamanlı planlamaya yakın bir hız elde edilmelidir.

Bilgisayarların daha az güç harcayarak daha yüksek işlem kapasitesine ulaşmasını sağlayan teknolojik gelişim, bütünleşik çevrimiçi yol planlayıcılara daha karmaşık kısıtlar ile çözümler üretebilme olanağı vermektedir. Böylece, çevrimiçi uygulamalarda da küresel yöntemler kullanılabilir.

Yol planlama problemleri, hedef ortamın genişliğine, İHA sayısına ve tipine bağlı olarak yüksek zorluk seviyesinde kabul edilmektedir. Yol planlama problemleri genel kapsamda NP-hard kategorisinde tanımlanır [34]. Ancak çoklu yol planlamanın söz konusu olduğu karmaşık durumlarda problemin NP-complete olduğu gösterilmiştir [35]. Örneğin, 3B yol planlamada arama uzayı genişledikçe ızgara noktalarının sayısı oldukça

artar ve yazılım literatüründe “boyut laneti” (curse of dimension) olarak isimlendirilen dramatik ölçüde hesaplama verimliliği azalması sonucunu doğurmaktadır. NP-hard problemlerde kombinatoriyal yöntemlerden ziyade örneklemeye dayalı teknikler tercih edilmektedir.

Yol planlamasında çevresel modelleme, bilinen harita bilgilerine göre oluşturulur. Gerçek ortam, harita özelliklerini de içeren bir çalışma ve yapılandırma alanı formatına dönüştürülür. Yol planlama tipik olarak iki aşamadan oluşur. Öncelikle, sürekli bir çalışma alanı bir grafik oluşturma problemi kapsamında çözülür. Sonrasında, oluşturulan grafik, verilen kriterlere göre belirli bir başlangıç noktasından bir varış noktasına bir yol bulma problemi kapsamında çözülerek yol seçilir [36]. Yol arama algoritmasının işlevi, çalışma alanında başlangıç noktası ile varış nokta arasında; yol uzunluğu, süre, enerji maliyeti vb. bir dizi optimizasyon kriterini karşılaması gereken çarpışmasız bir yol bulmaktır [37]. Yol planlamada en belirleyici kriter genellikle rota boyunca uçuş süresidir [38, 39].



Şekil 2.5. Yol planlama sınıflandırılması

Şekil 2.5’de yol planlama yöntemleri ve kullanılan algoritmalara yönelik bir sınıflandırılma gösterilmiştir [37].

Masehian ve Sedighizadeh, (2007) araç yol planlaması üzerine 1400 makaleyi kapsayan incelemesinde mevcut yaklaşımları klasik yöntemler ve buluşsal yöntemler olarak iki kategoriye ayırmışlardır [40]. A* algoritması veya potansiyel alan yöntemi gibi klasik yaklaşımların yüksek zaman karmaşıklığı zorluğuna sahip olduğunu ve sıklıkla yerel optimuma takılabileceğini, bu nedenle son araştırmalarda; yapay sinir ağları, bulanık mantık ve metasezgisel yöntemler gibi buluşsal yöntemlerin yoğunlaştığını belirtmiştir. Yarı-optimal çözümler elde etmek için genel bir optimizasyon yöntemi olan ve aday çözümlerin yinelemeler üzerinden iyileştirilmesine dayanan meta-sezgisel yöntemler, araştırmada en yaygın yaklaşım olarak görülmektedir [40].

2.2.1. Yol planlama özellikleri

Çevrede İHA'lar için tehlike oluşturabilecek alanlar bulunabilir. Yollar bu alanların dışından geçmelidir ve bir ayırma mesafesi bulunması gereklidir. Güzergâhta ziyaret edilecek veya görüş mesafesinde tutulması gereken alanlar da olabilir [41]. Bir İHA yol planlamasında özel koşullar, optimizasyon probleminin bir parçasıdır. Çevrede mesafenin korunması gereken bölgeler veya istenen temas noktaları buna örnek olarak gösterilebilir [42].

Bulunan yol, tanımlanmış uygunluk kriterlerini karşılamalıdır. Yol optimizasyonunda kriterler değiştiğinde yol, her seferinde yeniden hesaplanır. Genel olarak yol optimizasyonu üzerine yapılan çalışmalarında otonom robotik cihazlar için çerçevesi çizilen optimizasyon kriterleri, İHA'lar açısından da geçerlidir:

- 1) Seyahat süresi: Görevin mümkün olduğunca çabuk tamamlanmasının önem derecesini belirtir.
- 2) Yol uzunluğu: Başlangıç noktasından varış noktasına giden yolun en kısa mesafede kat edilerek geçmenin önem derecesidir.
- 3) Engellere olan mesafe: Aracın hareket ederken engellerden uzak durmasının ne kadar önemli olduğudur [43]. Araç hareket halindeyken engellere olan güvenli mesafeyi korumalıdır.
- 4) Tehlikelere olan mesafe: Çevrede araç için potansiyel tehlikede nesneler varsa, bu tür bir tehlikeye doğrudan temastan kaçınılmalı arada belirli bir mesafe bırakılmalıdır.

- 5) Yolun düzgünlüğü: Yörünge boyunca hareket ederken keskin dönüşler olmamasının ne kadar önemli olduğudur [44]. Bu tür manevralar, hareket yönünü aniden değiştirebilen ve yerinde dönebilen çok yönlü robotlar, multirotor İHA'lar için ek enerji sarfıyatı gerektirmekle birlikte mümkündür. Ancak sağa veya sola ani hareket edemeyen sabit kanatlı İHA'lar için bu kriter hayati bir öneme sahiptir. Kinematik kısıtlar araca özgüdür.
- 6) Görüş hattı: Aracın başlangıç veya varış noktası ile doğrudan görsel temas halinde olması, bir izleme cihazı ile iletişimin üst seviyede sürdürülmesi gerektiği kriterdir. Mobil istasyonların izlenmesi problemi buna bir örnektir [45].
- 7) İletişim merkezlerine mesafe: İHS ile iletişimin sürdürülebilmesinin ne derece önemli olduğu ile ilişkili kriterdir.
- 8) Harcanan enerji: Aracın güzergâh boyunca harcadığı enerjinin en aza indirilmesinin önemini belirten kriterdir.

Bir yol planlama algoritmasından beklenen özellikler aşağıdaki gibi özetlenebilir [46]:

- Sonlu olma: Çalışma alanı ne kadar büyük olursa olsun, hesaplamanın algoritma seyri içerisinde bir kilitlenme olmadan sona ereceği garanti edilmektedir.
- Tamlık: Algoritma geçerli bir çözüm üreterek sonlanırsa bir algoritma tamamlanmıştır.
- Kabul edilebilirlik: Üretilen yolların optimal ya da optimale yakın olduğunun garanti edilmesidir.
- Alan ve zaman karmaşıklığı: Algoritmanın ihtiyaç duyduğu bellek boyutunun ve algoritma çalışma süresinin dallanma faktörüne, çözüm derinliğine, düğüm sayısına vb. bağlı olarak nasıl büyüdüğü önemlidir. Bazı yaklaşımlar, hesaplama süresini iyileştirmek için bellek miktarının büyümesini göze alır. Algoritmanın çözüm üretmesi için geçen süre algoritmanın hesaplama karmaşıklığı ile doğrudan ilişkilidir. Çoklu araç problemlerinde daha fazla araç eklendikçe hesaplama süresinin orantılı olması beklenmektedir.

2.2.2. Çevresel modelleme ve çalışma alanı

Yol planlama yapılabilmesi için çalışma alanının kullanılabilir ve kullanılamaz bölgelere ayrıştırılması (discretization) gereklidir. İlerleyen bölümlerde açıklanacağı üzere bu işlem iki şekilde yapılabilir [47]: Birincisi sürekli ortamın tam olmayan ayrıştırması, ikincisi ise tam ayrıştırma.

Görünürlük diyagramı, Voronoi diyagramları ve hücresel ayrıştırma yaygın uzay ayrıştırma yöntemleridir. Bu yöntemler uygulandıktan sonra yolların oluşturulması, bir optimizasyon problemi olmaktadır. Örneklemeyle dayalı yöntemlerde ve Potansiyel Alan Teorisi'nde (APF) ise modelleme ve yol oluşturma birlikte değerlendirilir.

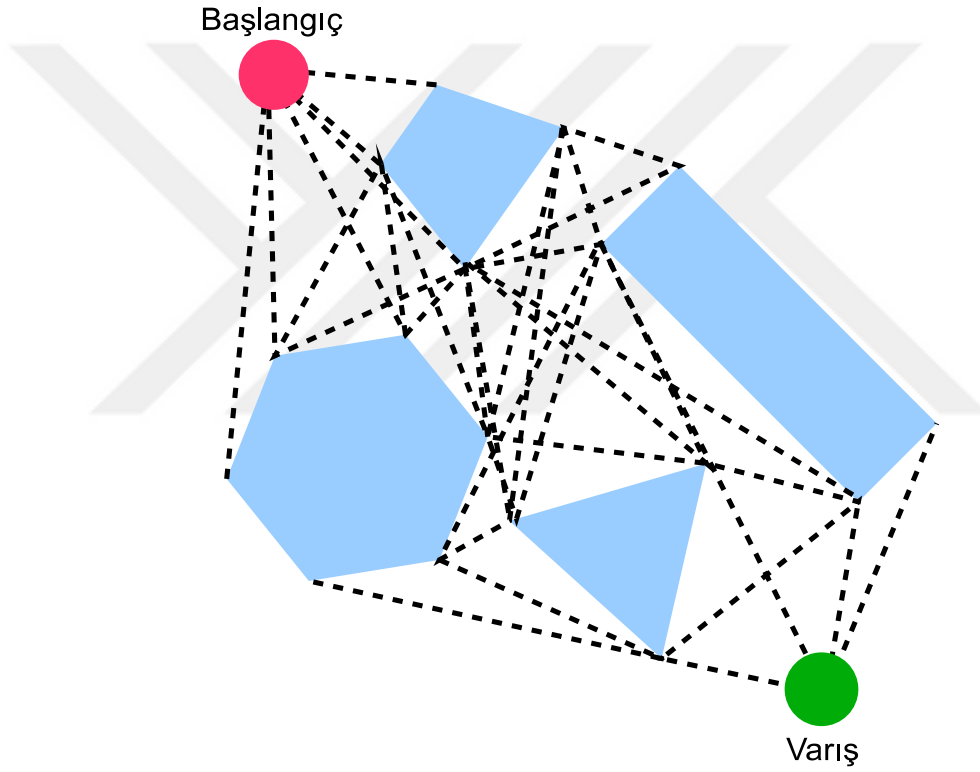
2.2.2.1. Görünürlük diyagramı

Görünürlük diyagramı, Lozano-Pérez ve Wesley, (1979) tarafından önerilmiştir [48]. Düzlemsel bir haritadaki engeller çokgenler şeklinde verildiğinde, başlangıç ve varış noktaları ile engellerin tüm köşeleri, çokgen köşe çiftleri olarak eşleştirilerek bir görünürlük diyagramı oluşturulabilir. Pérez, üretilen yolun robotun boyutuna uygun olması için köşe noktası konumlarını engel konumlarından uzağa kaydırarak başlangıç noktası ile varış noktası arasındaki en kısa yolu bulmak için ürettiği görünürlük diyagramı üzerinde arama algoritması uygulamıştır.

Görünürlük diyagramları, başlangıç koordinatına, varış koordinatına ve her çokgen engelin köşelerine birer nokta yerleştirerek sürekli 2B ortamı grafiklere böler. Sonra görüş hattındaki her köşe çifti için diyagrama bir kenar eklenir [48, 49]. Her tepe noktası, kendisini diğer tüm tepe noktalarına bağlayan kenarlara sahip olabilir. Bu nedenle, görünürlük diyagramlarındaki kenarların sayısı, köşelerinin sayısının ikinci dereceden kuvveti olabilir. Kenar sayısı azaltılabilirse de kenar sayısının üst limiti yine de köşelerin ikinci dereceden kuvvetidir [50]. Görünürlük diyagramlarının ve azaltılmış görünürlük diyagramlarının engelleri çokgen şekilde tanımlanan 2B ortamlarda en kısa yolları bulması garanti edilmektedir [48].

Görünürlük diyagramları, 3B ortamlarda aynı şekilde oluşturulabilir. Ancak, 3B ortamlarda oluşturulan görünürlük diyagramlarının en kısa yolları bulabilmesi garanti edilmez [51]. Ayrıca, çokyüzlü engellerin tipik çokgen engellerden daha fazla köşesi bulunmaktadır. Arama algoritmalarının zorluğu da bu oranda artmaktadır.

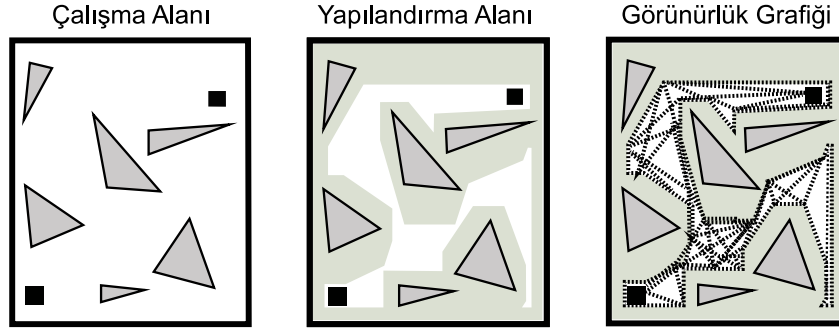
Görünürlük diyagramlarında, çözüm uzayındaki engellerin doğru planlamaya imkân verecek şekilde gösterilmesi ve basitleştirmek adına olduğundan daha geniş gösterilmesi çarpışmayı önlemek için kullanılagelen bir yöntemdir. Bunun için tamamen engelsiz ve çarpışmayı önlemeyi garanti eden bir ön işlem yapmak önerilir. Minkowski toplamı yöntemi ile engelleri temsil eden şekiller bir oranda genişletilir. Böylelikle gereken açıklık sağlanır [52]. N , çokgen engelin köşe sayısı ise hesaplama güclüğü Naïve algoritmasında $O(N^3)$, Sweep algoritmasında $O(N^2 \log N)$, Optimal algoritma ile $O(N^2)$ 'dir. Şekil 2.6'da $O(N^3)$ algoritması tarafından dışbükey engeller ile oluşturulmuş görünürlük haritası örneği verilmiştir [53].



Şekil 2.6. $O(n^3)$ algoritması tarafından dışbükey engeller ile oluşturulmuş görünürlük haritası

Görünürlük diyagramı yöntemiyle aynı veri yapısına sahip tanjant grafik yöntemi Liu ve Arimoto, (1992) tarafından tanımlanmıştır [50]. Teğet grafik yöntemi, kavisli engeller arasında da yol planlamasına imkân verirken yerel en kısa yolları temel alır ve $O(K^2)$ bellek alanı gerektirir. K , mania sınırlarının dışbükey parçalarının toplam sayısıdır.

Çokgen engelleri temel alan yol planlaması, engellerin ve araç konumlarının belirlenmesinde yapılandırma (konfigürasyon) alanı konseptinin ortaya çıkmasına yol açmıştır. Şekil 2.7'de çalışma alanından yapılandırma alanının oluşturulmasına dair bir örnek verilmiştir [50].



Şekil 2.7. Çalışma alanından yapılandırma alanının oluşturulması

Literatürde görünürlük diyagramının kullanıldığı çok sayıda çalışma bulunmaktadır. Medeiros ve Da Silva, (2011) sayısal yükseklik modelleri ile arazi yüksekliklerini temsil ederek görünürlük diyagramı oluşturmuşlar ve Dijkstra algoritması kullanarak yollar üretmişlerdir [54].

Jiang, Seneviratne ve Earles, (1993) görünürlük diyagramı yaklaşımına dayalı 3B yol planlaması için çokgen şeklinde tanımlanabilen engellerin varlığında en kısa yolun geçebileceği kenar dizisini belirlemek için bir sıralama getirmişlerdir [55]. Asgari potansiyel enerji ilkesi kullanılarak özyinelemeli yapılan yol bulma işlemi ile optimal olmayan yollar hesaplanmıştır. Bununla birlikte, işlem süresinin polinom olarak köşe veya düğüm sayısı ile ilişkili olması önerilen yöntemin dezavantajıdır.

Omar, (2012) 2D ve 3D kentsel ortamda görünürlük diyagramı kullanmıştır [56]. Geniş ortamlarda tüm köşelerin kullanımının ortaya çıkardığı işlemsel karmaşıklığın önüne geçmek için nispeten daha az sayıda köşenin seçildiği base line (BL) yöntemi tanıtılmıştır. Böylelikle büyük oranda hız artışı ölçülmüş ve gerçek zamanlı uygulamalara uygulanabilirliği gösterilmiştir. Köşeler, engel kenarları ve belirli açılar ile döndürülmüş bir düzlem arasındaki kesişme noktaları olarak belirlenerek yol planlama için düzlem rotasyon yaklaşımı isminde bir yöntem önerilmiştir. Yöntem MATLAB üzerinde test edilmiştir.

Bygi ve Ghodsi, (2007) görünürlük diyagramına dayalı 3B uzayda, kabul edilebilir bir hesaplama süresi ile 2B grafiği 3B grafiğe genişletmek için 3B görünürlük diyagramı adlı yeni bir yapıyı tanımlamışlardır [57].

Görünürlük diyagramlarının kullanıldığı bir diğer çalışmada, kaza durumunda popülasyon açısından risk değerlendirmesi yapılmıştır [58]. İHA'lar için herhangi bir kaza durumuna karşı toplum güvenli açısından en kısa yolların oluşturulmasına

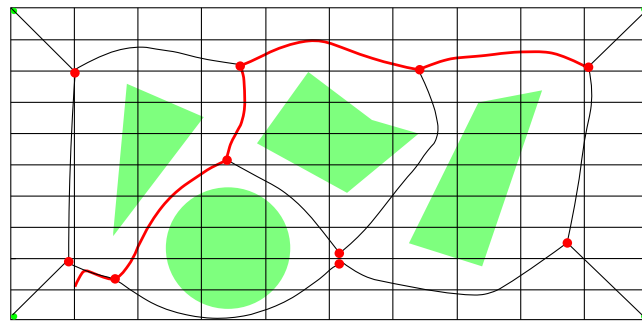
çalışılmıştır. Schøler, La Cour-Harbo ve Bisgaard, (2012) 3B yapılandırma alanında görünürlük diyagramlarını oluşturacak yeni bir yöntem önermişlerdir [28].

Omar ve Gu, (2010) görünürlük diyagramı temelinde bir dizi 3B yol planlama algoritmaları tanıtmışlardır [59]. Bu çalışmada, üzerinde görünürlük diyagramının oluşturulduğu bir dönme düzlemi kavramı önermişlerdir.

2.2.2.2. Voronoi diyagramları

Voronoi diyagramlarında bütün noktalar, iki veya daha fazla engel arasında eşit mesafede olacak şekilde yerleştirilir. En kısa olmasa da en güvenli güzergâh oluşturulur. Bu durumun istisnaları geçersiz yollara neden olabilir [53]. Voronoi diyagramları, tehditlere karşı yol uzaklığını en üst seviyeye çıkarmak için kullanılır. Engellere en uzak yolların üretilmesi askerî alanda oldukça yaygın bir kullanım imkânı vermektedir [60]. Bütün engeller hesaplandığında işlem tamamlanır. 1985 yılında tanımlanan Fortune algoritması, Voronoi diyagramı üreten çok sayıda algoritma içerisinde en performanslısı olarak kabul edilmektedir. Voronoi diyagramları kullanılan çoğu çalışma 2D'dir ancak 3D uygulamalar da mevcuttur. 3D hesaplamada duyulan işlem gücü azımsanmayacak seviyededir. Geniş ve karmaşık ortamlar işlemleri daha da zorlaştırır. Voronoi diyagramının bir çokgende iskelet veya medial eksen olarak ifade edildiği özel bir durumu da bulunmaktadır. Şekil 2.8'de örnek bir Voronoi diyagramı gösterilmiştir [50].

Voronoi diyagramları 2D ortamlarda etkilidir; bu nedenle irtifa kısıtlarına uyumlu değildir. Ayrıca, bu diyagramlar düz çizgiler içerir ve İHA, uçuş kinematiği nedeniyle bu düz çizgiler üzerinde uçamaz ancak işlem sonrası teknikler ile bu sorun giderilebilmektedir.



Şekil 2.8. Voronoi diyagramı

Voronoi diyagramına çok benzer bir diğer gösterim şekli Delaunay üçgenleştirmesidir [61]. Delaunay üçgenlerinin merkezleri Voronoi noktalarıdır. Bir

başka ifadeyle her engel bir üçgenin köşesini oluşturur. Delaunay üçgenleştirmesi, engellerin komşulukları ve birbirlerine mesafelerini kolaylıkla bulunmasını sağlamaktadır. Bu avantaj, iki engel arasındaki mesafenin İHA'nın geçemeyeceği genişlikte olduğu düğümlerin silinmesini kolaylaştırır.

Yollar oluşturulduktan sonra yakıt ve tehdit durumuna göre ağırlıklandırılmış bir maliyet belirlenebilir [60]. Bazı çalışmalarda Delaunay üçgenleştirmesinin merkez noktaları üzerinden yollar oluşturulur [61]. Sadece engeller arasındaki mesafe ve en yakın komşuluğun tespitinde Delaunay üçgenleştirmesinden faydalanılan çalışmalar da bulunmaktadır [62].

Qu, Pan ve Yan, (2005) Voronoi diyagramları üzerinden A* algoritması kullanarak optimale yakın yollar oluşturmuşlardır [63]. Bu yollar, İHA dönme açısı kısıtları dikkate alınarak geometrik yöntemlerle düzeltilebilir yollardır. Aynı çalışmada ikinci bir yöntem olarak, APF'nin en bariz dezavantajı olan yerel optimuma takılmayı önlemek için GA ve APF birlikte kullanılmıştır.

Pehlivanoglu, (2012) çeşitlilik farklılıklarını vurgulayan çoklu frekans titreşimsel GA ile küresel rastgele ve yerel rastgele mutasyon uygulama stratejisini önermiştir [64]. Başlangıç popülasyonu oluşturulurken Voronoi diyagramı ve kümeleme yöntemi kullanmıştır. Yöntem sinüs ve şehir tipi arazi modelleri üzerinde denenmiştir. Tüm popülasyonun tüm genlerine periyodik olarak bir titreşimsel mutasyon operatörü uygulayarak, yerel optimumlardan kaçmak amaçlanmıştır. Çalışmada, başlangıç popülasyonu rastgele sayı operatörü kullanılarak üretilir, BLX-a çaprazlama operatörü ile eşleştirilir ve mutasyon aşamasında titreşimsel mutasyon operatörü kullanılmaktadır.

Men, E ve Yao, (2012) kapasite, yakıt maliyeti, havadan gelebilecek tehditler vb. kısıtları değerlendirerek, Voronoi diyagramı ile oluşturulan yolları Dijkstra ve GA kullanarak seçmişlerdir [65]. Denizaltı arama operasyonlarına ait yol planlamanın başarıyla gerçekleştirilebildiği garanti edilmiştir.

Hu, Song ve Wang, (2009) çok amaçlı GA kullanarak, her İHA'nın kendi yedek yollarına sahip olduğu ve aynı anda varış bölgesine ulaşmayı sağlayan Voronoi diyagramı temelli bir uygulama yapmışlardır [66].

Jian-You, (2007) GA ile tehdit kaynaklarının modellenmesi üzerine çalışmıştır [67]. Voronoi diyagramı temelinde görev planlama çözümleri üretilmiştir.

Eun ve Bang, (2009) başlangıç popülasyonunu kaba 2D Voronoi diyagramları ile oluşturmayı tercih etmişlerdir [68]. Amaç fonksiyonu, çoklu İHA görevlerinin tamamlanması için gereken toplam süre olarak belirlenmiştir. Uyarlanmış kromozom gösterimi, tek tip mutasyon operatörü ve döngü çaprazlama tekniğinin kullanıldığı GA uygulanmıştır.

Sud vd., (2008) birinci ve ikinci dereceden Voronoi diyagramları ile oluşturulan yeni bir veri yapısı ve çoklu sanal araç için navigasyon grafiği kullanarak bir yaklaşım sunmuşlardır [69]. Önerilen yöntem, gerçek zamanlı olarak her araç için rota planlama ve yakınlık hesaplamaları yapmakta kullanılmıştır. Algoritma, her biri ayrı bir hedefe sahip yüzlerce hareketli araç için gerçek zamanlı kalabalık simülasyon senaryolarında takipten kaçınma, arazi keşfi ve çoklu araç planlama için kullanılmıştır.

Krozel ve Andrisani, (1990) 2B düzlem problemini çözmek için dağları çokgenleştirmeye çalışmışlardır [70]. Engelleri modellemek için çokgenlerden alınan tepe noktalarını Delaunay konumları olarak kullanmışlardır.

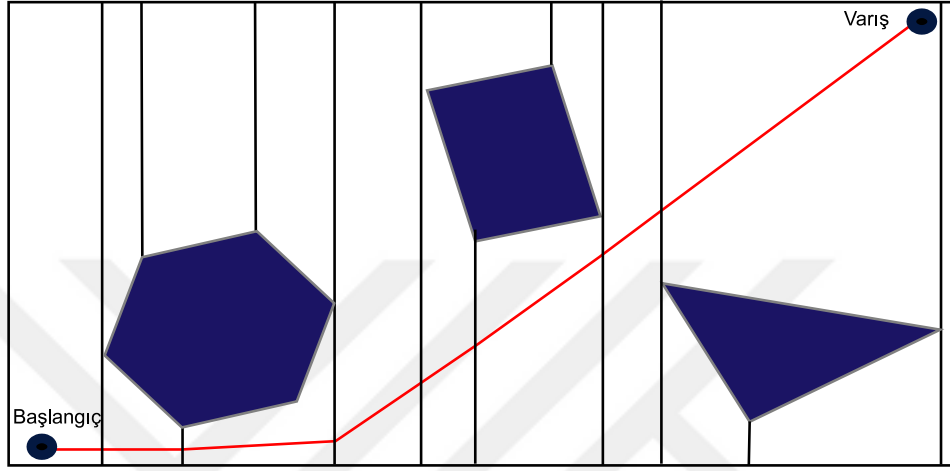
Spritzer, (2004) Voronoi diyagramının hesaplama açısından daha maliyetsiz çözüm sağladığını ifade etmiş ancak optimum çözümü elde edebilmek için görünürlük diyagramı kullanmıştır [71]. Bir savaş ortamının tüm bilinen özellikleri ile arama ve yok etme görev senaryolarında Dijkstra algoritması kullanarak testler gerçekleştirmiştir.

2.2.2.3. Hücresel ayrıştırma

Yol planlaması probleminde, hücre ayrıştırma yaklaşımı geometrik tabanlı yöntemlerin başka bir türüdür [72]. Haritayı engelsiz daha küçük bölgelere bölmek için çokgen engellerin köşeleri kullanılır. Başlangıç ve varış konumlarını içeren ardışık engelsiz komşu bölgeleri bulmak için grafik arama algoritmaları uygulanmaktadır [73].

Hücresel ayrıştırma, tam ve yaklaşık hücresel ayrıştırma olarak ikiye ayrılmaktadır. Tam ayrıştırmada sürekli uzay, engellerin köşeleri esas alınarak birbirine paralel hücrelere bölünmektedir. Hareketin hücre merkezinden yapıldığı kabul edilmektedir. Yaklaşık hücresel ayrıştırmada hücrelerin aynı boyutta olması şart değildir; belirtilen asgari hücre boyutuna kadar kısmi engel içeren hücreler bölünebilirler. Boyut artışı ile bölünen hücre sayısı üssel olarak artmakta ve işlem zorluğunu beraberinde getirmektedir [52]. Şekil 2.9'da görüleceği üzere tam hücresel ayrıştırmada engel köşelerinden başlayan yatay veya dikey çizgiler bölgesel ortak sınırlar oluşturmak için uzatılır.

Yaklaşık hücre ayrıştırması yaklaşımında, yapılandırma alanı üzerinde düzenli bir ızgara tanımlanır [74]. Kullanımı kolaylaştırmak için ızgara hücreleri, şekil ve boyut açısından uygulama öncesinde belirlenir. Bu tez çalışmasında önerilen algorithmada kullanılan bir yöntem olan düzenli ızgara grafiklerine özgü özellikler aşağıda detaylandırılmıştır.

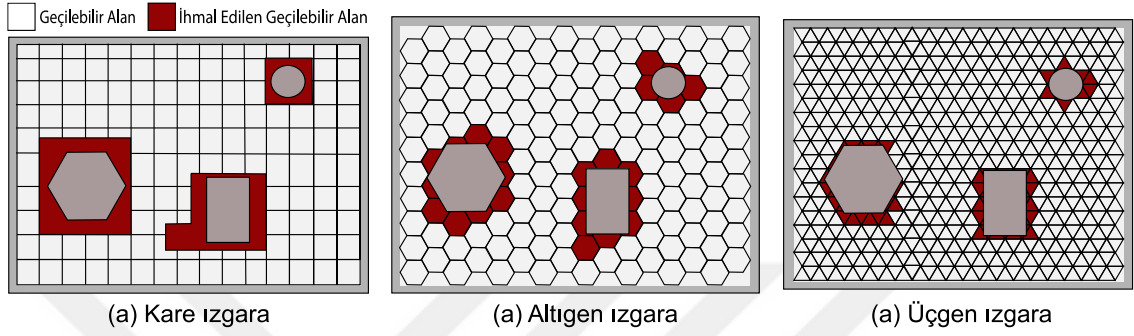


Şekil 2.9. Tam hücre ayrıştırma

Yaklaşık hücre ayrıştırması ızgara grafiğinde belirli bir başlangıç koordinatından bir varış koordinatına, ızgara olarak ayrıştırılmış sürekli kare yol noktaları üzerinde yollar elde edilmektedir. Hücre ayrıştırma teknikleri, ortamı hücrelere bölmektedir. Araç hareketi için uygun olan ve olmayan hücreler tanımlanır. Genellikle kare, daire veya dışbükey çokgen olan hücreler; engelsiz, geçilebilir alanların seçilebilmesine imkân vermektedir. Sürekli ortamda herhangi bir koordinata karşılık gelen ızgara hücresi kolaylıkla belirlenebilir [75].

Sabit zamanda herhangi bir koordinatta ızgara hücresi ile ilişkili herhangi bir bilgi hücrede tanımlanabilir [76]. Hücre ayrıştırması ile her hücre, kapsadığı alana dair çeşitli bilgiler içerebildiği için sürekli ortamda kullanılabilir alanların bir temsili olarak işlev görmektedir. Hücrelerin yer seviyesi yüksekliği, rüzgâr hızı, hava durumu vb. çok sayıda bilgiyi taşıyabilmesi ile bu açıdan görünürlük diyagramından veya yol noktası grafiğinden daha elverişlidir. Hücreler bir veri yapısı olarak hem yol planlamada hem de sistemin diğer bölümlerinin ihtiyaç duyabileceği bilgileri saklamak için kullanılabilir. Hücreler, yol arama algoritmaları tarafından yolları hesaplamak için kullanılmaktadır. Düzenli ızgaralarda rastgele erişim mümkündür [76].

Düzenli ızgaralar, sürekli ortamlarda düzenli çokgenlerden bir mozaik oluşturur. Sürekli 2B ortam mozaiklerinde Şekil 2.10'da örnekleri gösterilen kare, altıgen ve üçgen olarak üç tip düzgün çokgen kullanılır [36, 76, 77]. Düzenli her çokgen, bir ızgara hücresi ya da kısaca hücre şeklinde isimlendirilebilir. Düzenli ızgaralardan oluşturulan grafikler, ızgara grafikleri olarak ifade edilebilir.



Şekil 2.10. Düzenli ızgara grafikleri

Her hücre ayrıştırma tekniğinin gerektirdiği bellek miktarı, ele alınması gereken birçok farklı konudan biridir. Hücre ayrıştırma tekniğinin uygulama zorluğu da diğer bir konudur. Örneğin, daireleri tanımlamak, dışbükey çokgenlerden daha kolaydır ancak dairelerin yapısı itibarıyla tüm alanı taramasında zorluklar ortaya çıkabilmektedir.

Kare ızgaralar, sabit kenar uzunluğu ve bir referans koordinat noktasına sahip oldukları için uygulanması en basit hücre ayrıştırma tekniğidir. Altıgen ızgara grafikleri üzerinde yapılan aramaların, kare ızgara grafikleri üzerinde yapılan aramalardan daha düşük arama karmaşıklığına sahip olduğunu gösteren çalışmalar olmakla beraber [78], altıgen ızgaraların bazı avantajlarına rağmen kare ızgaralara kıyasla daha az tercih edildiğini belirten çalışmalar da bulunmaktadır [79]. Kare ızgaralar üzerinde yol aramada temel hesaplama zorluğu, aracın dikey geçişi ve çapraz geçişi ile ilgili yol uzunluğu farkıdır. Hesaplama algoritması, bu farkı değerlendirmelidir. Kare veya altıgen ızgaralar kadar yaygın kullanılmaları da üçgen ızgaralar da beklentileri karşılamaktadır.

İzgara tabanlı yaklaşımda ortam hücrelere ayrılır. Hücrelerin boyutlarının genellikle aracın güvenle hareket edebileceği kadar geniş olması gereklidir. Li, Sun ve Xu, (2006) yaptıkları 2D bir çalışmada yapılandırma alanını eşit kare hücre ızgara yöntemi ile göstermişlerdir [80].

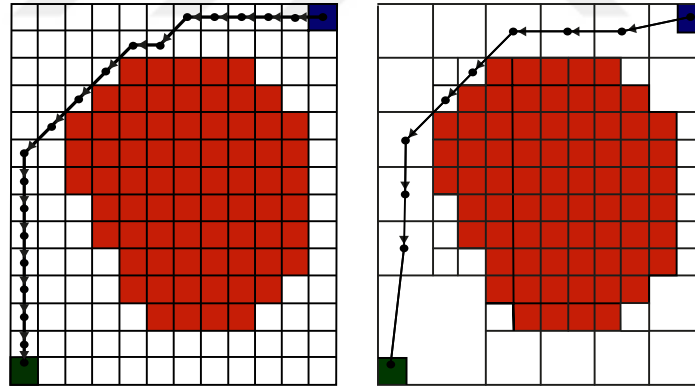
Gao vd., (2012) görerek uçan bir mikro İHA uygulamasında çoklu çözünürlüklü hücresel ayrıştırma yöntemi ile engeller ve rüzgâr bilgisini değerlendirerek geçerli yollar

oluşturmuşlardır [81]. Algılayıcı bilgilerinin kesin olduğu durumlarda çözünürlük yüksek, kesin olmadığı durumlarda düşük olmaktadır.

Düzenli ızgaralar, bilinen 3B ortamları ayrıştırmak için de kullanılabilir. Alanın düzenli çokgenler yerine düzenli çokyüzlülerle mozaiklenmesi ile sürekli bir 3B ortamı mozaikleme için tek çeşit düzenli çokyüzlü kullanılır. Prizmatik üçgenler ve altıgenler de aynı şekilde mozaiklemede kullanılabilir. Kübik ızgaralarda da kare ızgaralara ait istenen özellikler bulunmaktadır.

3D ızgara yöntemlerinden birisi Küp uzayı (C-Space) algoritmasıdır [82]. Bu 3D yöntemde temel prensip uzayın küplere bölünmesi ve her küpün boş veya dolu olarak işaretlenmesidir. Bunun yanında her küp varış noktasına gerçek uzaklığı da saklayabilmektedir [83].

3D ızgara tabanlı ayrıştırma ile A* algoritması kullanılmıştır [84]. Bir diğer çalışmada Krawiec, (2012) Yamaha RMAX insansız helikopter ile testlerini gerçekleştirdiği ve A* algoritmasını kullandığı çalışmasında, yapılandırma alanını 3D ızgara yöntemini kullanarak ayrıştırmıştır [85].



Şekil 2.11. Küp uzayı (solda) ve Octree (sağda) algoritmaları

Octree, C-Space algoritmasına benzerlik gösterir ve çoklu çözünürlükte temsil imkânı veren hiyerarşik bir yapısı bulunmaktadır. Çoklu çözünürlüğün bir avantajı olarak geniş boş uzayda C-Space'e kıyasla hızlı geçiş mümkündür. Küpten kübe geçiş yerine tek seferde birden fazla kübe geçiş yapılabilir. Çalışmalar Octree'nin daha etkin planlama yaptığını göstermektedir. Avantajlarına karşın bellek ihtiyacı çok daha fazladır. Bellek maliyetlerinin az olması bu dezavantajın etkisini azaltmaktadır. Nispeten karmaşık yapısı içerisinde arama işlemleri C-Space algoritmasına göre çok daha yavaştır. Şekil 2.11'de Küp uzayı ve Octree algoritmalarının çalışmadaki farklılıkları gösterilmiştir [86].

2.2.2.4. *Örnekleme tabanlı yaklaşımlar*

Grafik tabanlı yöntemlerin yanında örnekleme tabanlı yaklaşımlar da başlangıçtan varış noktasına giden bir yol oluşturmak için kullanılan metodolojilerdir. Her yöntemin farklı uygulamalarda kendine özgü avantajları ve dezavantajları bulunmaktadır. Bu farklılıklar probleme özgü ideal bir seçim yapılmasına imkân vermektedir.

Temelde grafik tabanlı yöntemlerin ortamı ayrıştırarak hesaplama maliyetini düşürdüğü buna karşın düğüm sayısının çözünürlüğe göre sınırlı kalabileceği önceki alt bölümlerde özetlenmiştir. Farklı olarak, örnekleme tabanlı yaklaşımlar, ayrıştırılmamış örnekleme yapısı ile engellerin nispeten bütün ve büyük olduğu bir ortamda hesaplama açısından daha maliyetli olabilmektedir. Ancak yine de yöntem çözüme kademeli yakınsamaktadır.

Acil karar verme durumunda veya bellek kısıtları açısından tüm çalışma alanının ayrıştırılması imkânsız olduğu durumlarda örneklemeyle dayalı yaklaşımlar tercih edilebilir. Başlangıçta yapılan tam bir ayrıştırmanın ortamın sürekli değiştiği durumlarda gereksiz olacağı anlayışıyla yinelemeli bir ayrıştırma düzeni kullanılmaktadır. Bu yöntem karakteristik olarak arama uzayında rastgele düğümler oluşturarak bir yol bulur ve onu varış noktasına götüren mevcut yerleşik ağaca bağlamaya çalışmaktadır.

Örneklemeyle dayalı yöntemler, göreceli düşük hesaplama gereksinimleri ve yakınsama garantileri nedeniyle çeşitli alanlarda yaygın kullanıma sahiptir [87]. En yaygın kullanılan deterministik yinelemeli örnekleme yöntemleri Rapidly-Exploring Dense Trees (Hızlı-Keşfedilen Sık Ağaçlar) (RDT) algoritmasının yol planlama çalışmalarında sıklıkla kullanılan varyantı Rapidly-exploring Random Trees (Hızlı-Keşfedilen Rastgele Ağaçlar) (RRT) [88] ve Probabilistic Roadmap (Olasılıksal Yol Haritası) Yöntemi (PRM)'dir [89]. Bu yöntemler sırasıyla döngüsel ve döngüsel olmayan rastgele ağaçlar oluşturmaktadır [87].

Olasılıksal yöntemlerin genel ilkesi, izin verilen engelsiz ortamın bir grafikte temsil edilmesidir. Ağ olarak adlandırılan bu durumda, aracın ilk konfigürasyonu (konum ve yönelimi) ilk düğüm olarak belirlenmekte ve diğer düğümler rastgele seçilerek oluşturulmaktadır.

Lavalle, (1998) tarafından tanıtılan RRT yönteminin amacı diğer yol planlama yöntemleri ile benzer şekilde, başlangıç ve varış düğümleri arasında çarpışmasız ve aracın

kinematik kısıtlarına uyarlanmış bir yol bulmaktır [90]. Boş alandan rastgele seçilen noktalar en yakın komşuları ile ilişkilendirilir. Her yinelemede, arama alanında rastgele yeni bir düğüm oluşturulur. Kendisi ile önceden oluşturulmuş düğümlerden herhangi biri arasında yol kullanılabilir ise yola bağlanır. Böylece, yol ağacı yinelemeli olarak p olasılıkla varış noktasına doğru, $p-1$ olasılıkla rastgele olarak genişletilir. Bu şekilde tüm düğümlere başlangıç noktasından erişilebilen bir ağaç grafiği oluşur. Algoritma, başlangıç ve varış düğümleri arasındaki mesafe önceden tanımlanmış bir değere ulaştığında aramayı durdurmaktadır.

RRT algoritmasının bir zayıflığı, dar geçitler içeren ortamlarda çok verimli olmamasıdır. Önceden hesaplanmış yol üzerinde RRT algoritmasının uygulanması ile daha iyi sonuçlar elde edilebilmektedir [91]. Araçların bilinmeyen bir ortamda çalıştığı durumlarda önceden hesaplanmış bir yolun kullanılması mümkün olmaz. Bu yöntem olasılıksal doğası gereği düzensiz veya uzun yollar ürettiğinden çözüm genellikle optimal değildir. Bu nedenle bir yumuşatma algoritması kullanılmalıdır.

İlk adımda tam bir ayrıştırmayı gerektiren A* gibi birçok çevrimdışı algoritmaya kıyasla RRT gibi algoritmalar, çevrimiçi ve sürekli değişen dinamik çevrelerde kullanılabilir. Bruce ve Veloso, (2002) çoklu-robot sistemi bağlamında gerçek zamanlı bir uygulamada, ERRT (Execution extended RRT) algoritmasını başarıyla uygulamışlardır [92].

Yang, Keat Gan ve Sukkarieh, (2013) bilinmeyen karmaşık bir ortamın keşfedilmesi ve gezilmesi için görerek ilerleme prensibi kapsamında RRT ile birlikte Gauss Süreci kullanmışlardır [93]. Bilinmeyen 3D ortamda uygulama görevleri başarıyla gerçekleştirilmiştir.

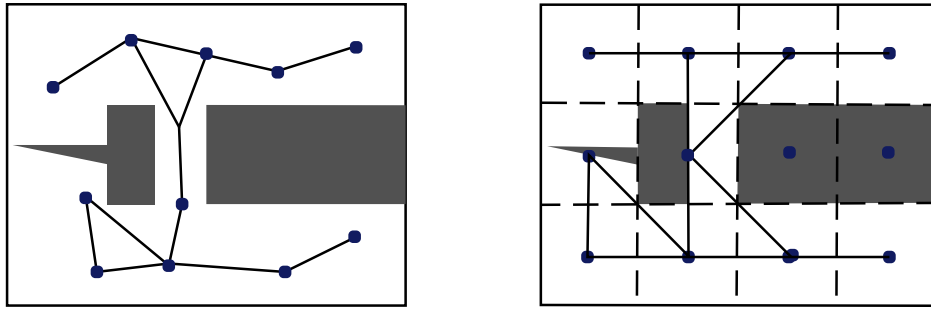
Bir diğer örneklemeye dayalı yöntem PRM algoritması, öğrenme ve sorgulama olarak iki aşamadan oluşur. Öğrenme aşamasında bir yol haritası oluşturmak için örnek noktalar birleştirilirken sorgulama aşamasında bir grafik arama algoritması kullanılarak başlangıçtan varış noktasına kadar uygun bir yolu aramak için önceden oluşturulmuş yol haritası kullanılmaktadır [89].

Standart PRM yönteminin performansını artırmak için literatürde bir dizi yöntem geliştirilmiştir. PRM*, değişken bir mesafe kısıtlaması ile yolu optimalleştirir [94].

PPRM (parçacık olasılıksal yol haritası) ve GPRM (ızgara olasılıksal yol haritası) bir ızgara ortamında çalışan olasılıksal yaklaşım kullanan varyantlardır [95].

FMT algoritması, RRT* ve PRM*'in avantajlarını birleştirebilmek için Janson vd., (2015) tarafından asimptotik optimal olarak önerilmiştir ancak çalışmalarında yol güvenliği ele alınmamıştır [96]. FMT'de düğümler; kapalı, açık ve ziyaret edilmeyen şeklinde sınıflandırılır. Açık kümedeki her düğüm, o düğüme seyahat maliyetiyle ilişkilendirilir. Her yinelemede, açık kümedeki en düşük maliyetli düğüm seçilir ve ziyaret edilmeyen kümedeki tüm komşuları bulunur. RRT* ve PRM* ile benzer şekilde, komşular yineleme sayısı ile azalan mesafeye göre tanımlanır. Bulunan komşular, açık kümedeki en yakın düğümlere bağlanır. Seçilen düğüm açık kümeden çıkarılır ve kapalı kümeye eklenir. Varış düğümüne bir bağlantı kurulursa arama başarılı bir şekilde sona erer. Buna karşın önceden tanımlanmış bir yineleme sayısına ulaşırsa veya ziyaret edilmeyen küme boşalırsa algoritma başarısız olmaktadır.

Genel olarak yol haritaları, bir noktalar topluluğu ve bu noktalar arası bağlantılardır. Yol haritalarının en önemli dezavantajı düğümlerin karakteristik temsil güçlüğüdür. Bir noktanın temsil ettiği alanın sınırlarını belirlemek güç olmakla birlikte çok geniş ortamları yönetebilmek için en uygun yöntemlerden birisidir [30]. Şekil 2.12'de olasılıksal yol haritası ve tek tip ızgara yol haritası karşılaştırılmıştır [97].



Şekil 2.12. Olasılıksal yol haritası (solda), tek tip ızgara yol haritası (sağda)

2.2.2.5. Potansiyel Alan Teorisi

Potansiyel alan algoritması çarpışma önleme algoritması olarak öne çıkmaktadır. Tepki süresinin diğer yol planlama algoritmalarına göre daha kısa olması çarpışma önleme amaçlı çalışmalarda bu algoritmaya yaygın bir kullanım imkânı vermektedir. Bir yol planlama algoritması olarak da kullanılmaktadır. En iyi çözümü değil ama hızlı bir çözümü garanti etmektedir.

Yapay Potansiyel Alan (APF) yöntemi Khatib, (1985) tarafından önerilmiştir [98]. APF, yolları bilinen bir çevre haritasından kolayca hesapladığı için tercih edilmektedir. Aracı engellerin ve varış noktasının ürettiği bir potansiyel alanın etkisi altında hareket eden bir parçacık şeklinde modellemektedir. Bu sezgisel teknik, aracı yerçekimine maruz kalan ve kumaş üzerinde hareket eden bir topa benzetmektedir. Top sürekli olarak aşağı yönlü en dik eğimi takip etmektedir. Top, aracı; kumaş ise çevreyi ifade eden bir temsildir. APF, varış noktasına doğru çekici bir alandan ve engellerin her birine bağlı itici alanlardan oluşmaktadır. Aracın yinelemeli olarak hesaplanan eğime göre iniş hareketi, engellerden kaçınarak varış noktasına ulaşmasına izin vermektedir. Her zaman için aracın bulunduğu yer potansiyel alan tarafından belirlenmiştir. Bununla birlikte, APF'nin saf formlarında kullanılmasının yerel optimumlarda takılıp kalması sorununa yol açtığına dikkat edilmelidir [99]. Yerel bir yöntem olarak APF için araç küresel optimuma yakınsamak yerine, potansiyel alan fonksiyonunun bir yerel optimumuna takılır. Geleneksel APF yöntemi, çekici ve itici kuvvetler eşit olduğunda ve varış koordinatına yakınsamadığında yerel optimumlarda sıkışıp kalabilmektedir. Yerel optimumdan kaçmak için farklı teknikler geliştirilmiştir [100]. Harmonik Potansiyel Alanı (HPF) olarak da bilinen Laplace denklemlerini sağlayan harmonik fonksiyonlar ile sınır tanımlaması yerel optimuma takılmayı önlemektedir [101].

APF'de hesaplanan bilgi, robotun hareketiyle doğrudan ilişkilidir ve hesaplama için ek bir işlem gücü harcanmamaktadır. Bir etki olduğunda eklenen yeni engelin potansiyel alanı, eskisinin üzerine kolaylıkla güncellenmektedir [100].

Gancet vd., (2005) çoklu İHA için geliştirilen bir görev planlama ve kontrol sisteminde, bir potansiyel alan tabanlı algoritmayı arazinin yangın riskini tespit etmek için uygulamışlardır [102]. Zeminde her bir hücre bir potansiyele sahiptir. Azami seviyede başlatılmış değerler, belirli bir kurala göre zamanla azalmaktadır.

Potansiyel alan yöntemleri diğer yöntemlerle birlikte de kullanılmaktadır. Miralles ve Bobi, (2004) önerdikleri Voronoi diyagramına çok benzeyen yol haritası (DMP) yöntemiyle, APF'ye dayalı olarak çeşitli parametreler ile yol planlaması yapabilmektedir [103]. Magid vd., (2006) yinelemeli yol arama algoritmasında APF kullanmışlardır [104]. Tang vd., (2010) yol planlamada engellerden kaçınmak için yerçekimi zinciri ilkesine dayanan yeni bir APF yöntemi sunmuşlardır [105]. Yöntemde, potansiyel bir alandaki

başlangıç ve varış noktası pozisyonlarını engellerle birleştiren bir tür “elastik levha” fikri önerilmiştir.

Mora ve Tornero, (2008) İHA'lar için yapılan çalışmada, dinamik ve belirsiz ortamlarda çok oranlı Kalman filtre tahminleri ile geleneksel yapay APF yöntemini birleştirmişlerdir [106].

Brož, (2006) bir dizi faktör için 3B uygulamalarda kullanılabilecek bilinen, kısmen bilinen veya bilinmeyen ortamlarda çalışan gerçek zamanlı hibrit bir yol planlama tekniği önermiştir [107]. Yöntemde aynı boyutta iki ayrı harita kullanılır. Engel haritası tehlike ağırlıklarını temsil ederken, tehdit haritası bilinen tüm tehditlerin potansiyel alanlarını temsil etmektedir. Bilgiler sürekli güncellenmekte, algoritma, tüm yolların tanımlandığı iki haritayı da kapsayan bir ağ oluşturmaktadır.

Yin, Yin ve Lin, (2009) APF temelli bir yaklaşımı dinamik alanlara uygulamışlardır [108]. Önerilen yöntemde potansiyel fonksiyonlar, sadece robotun varış noktasına ve engellere göreli konumlarını değil aynı zamanda hareketli nesnenin varış noktasına ve engellere göreli hızlarını da hesaba katmaktadır.

Kitamura vd., (1995) dinamik bir 3B ortamda çarpışmasız yollar bulmak için ortamdaki her nesneyi octree kullanarak değerlendirdikleri bir yöntem önermişlerdir [109]. Yöntem, 3D bir ortamda bir araç için APF kullanılarak başlangıç konumundan varış konumuna rastgele arama yapmaktadır.

2.2.3. Yol planlama ve optimizasyon

Yol planlama probleminde en uygun çözüm, en az masrafa (asgari süre, mesafe ya da benzer bir maliyete) sahip yoldur, literatürde en kısa yol problemi (shortest path problem-SPP) şeklinde ifade edilmektedir. Bu problemde amaç, grafik üzerindeki düğümler arasında en düşük maliyetli yolu bulmaktır. SPP'ye örnek uygulamalardan olan gezgin satıcı problemi (TSP) ve optimal arama yolu (OSP) problemi NP-Complete olarak bilinmektedir [30].

Otonom hareketli sistemlerin özelliklerinin önemli bir bölümü yol planlaması gerektirir. Yol planlama algoritmalarının görevi, önceden belirlenmiş bir başlangıç noktasından varış noktasına zamana göre değişen ve değişmeyen çevresel engeller, hava araçları ve çeşitli belirsizliklerin mevcudiyetinde en uygun yolları oluşturmaktır. Yol

planlama, kara araçlarında olduğu gibi hava araçları için de irtifayı sabit kabul edilerek 2B yollar oluşturmaktayken işlem gücünün artmasıyla, karmaşık çevre ve model kısıtlarının karşılandığı 3B yol planlama yöntemleri de kullanılmaya başlamıştır. 2D ve 3D ortamlarda yol planlama sistemi statik ve dinamik bir dizi kısıt ile başa çıkmak zorundadır.

Yol planlama sonrası elde edilen yol, en kısa ya da en ekonomik yol olabilir. Ancak bunun yanında operasyon süresi, yakıt/pil kapasitesi, kapsama alanı, kinematik kısıtlar, azami irtifa, tırmanma hızı, dönüş açısı ve hız gibi parametreler dikkate alındığında her iki nokta arasındaki yol kullanılabilir ya da kullanılamaz olabilir. Kısıtların algoritmalarda dikkate alınması önemlidir. İHA'lar çok çeşitli alanlarda kullanılmaktadırlar buna karşın hangi alanda kullanılırsa kullanılsın yukarıda belirtildiği gibi çoğu ortak kısıtlara tabidirler. Birçok algoritma, İHA yol planlamalarına ilişkin kapsam sınırlılıkları nedeniyle bazı kısıtları ihmal etmiş veya hedeflerden herhangi birine odaklanmıştır. Aynı anda birden fazla hedef içeren amaç fonksiyonu ile çoklu çözüme yönelik algoritmalar, daha verimli bir optimizasyona imkân sağlamaktadır.

İHA'ların görevleri; görüntüleme, fotoğraflama veya askerî amaçlı hedef yok etmeyi gerektirir. Birden fazla İHA söz konusu olduğunda, rota planlayıcının iş yükünü nasıl dağıtacağını hesaplaması da yol planlama probleminin bir parçasıdır. Böyle çalışmalar görev planlama kapsamında değerlendirilmiştir. Planlamada takım hedefleri öncelenir ancak bazı İHA'ların sadece tek görev için özelleştirildiği durumlarda bir çatışma ortaya çıkabilmektedir. Dinamik ortamlarda, İHA'lar arasındaki iletişim, özellikle görevin yerine getirilememesi ihtimali söz konusu olduğunda daha önemlidir. Böyle durumlarda, takım üyeleri görev değişimi durumuna karşı, görevin mevcut durumu hakkında düzenli olarak bilgilendirilmelidir. İHA'lardan birisinin kaybedilmesi veya yeni bir hedef ve tehdit ortaya çıkması halinde iletişim daha da önem kazanmaktadır.

Genel olarak bir çevreyi temsil etmek için farklı grafik tabanlı modeller bir önceki bölümde açıklanmıştır. Grafik üzerinde her düğüm ulaşılabilir bir noktayı temsil eder ve düğümler arası bağlantılar yollarla ilişkilendirilir. Böylece, yol planlama bir grafik arama problemi olarak görülebilir. Bir grafik oluşturulduktan sonra, iki düğüm arasındaki en kısa yolu ve dolayısıyla aracın izleyeceği yolu bulmak için klasik yol arama algoritmaları kullanılmaktadır. Aşağıdaki paragraflar, yolları hesaplamak için kullanılan algoritmaları açıklamaktadır.

Genelde gerçek zamanlı algoritmalar, açgözlü arama stratejileri kullanırken; çevrimdışı algoritmalar küresel arama stratejisi uygulamaktadır. Yerel arama ile çözüm uzayında elde edilen uygun sonuç sezgiseller aracılığı ile geliştirilir. TSP ve SPP gibi çoklu-kriter yol planlama problemleri ve varyantları, yerel arama teknikleri kullanılarak çözülmüştür. Çalışma alanında düğümler arasında gezerek tüm sonuçlara ulaşılabilen komşuluk ilişkisinin tanımlanması gereklidir. Yerel aramanın en önemli zorluğu arama uzayının tanımlanması aşamasındadır.

Hem küresel hem de yerel yöntemler tek başına ideal değildir. Birden fazla yöntemin birlikte kullanımı, yöntemlerin zayıflıklarından kurtularak her birinin avantajlarından yararlanmayı mümkün kılabilir. Yol hesaplama yöntemlerinin çeşitliliği, çok sayıda kombinasyonun çalışılmasına imkân vermiştir. Özellikle küresel bir yöntem ile yerel bir yöntemin kombinasyonu çok sayıda çalışmada uygulanmıştır. Küresel bir yöntem, görevin kapsam genişliğine uygun bir yolu hesaplar. Ardından, seyir esnasında ortaya çıkacak sorunları yerel bir yöntemle yönetmek makul bir seçimdir.

Yol planlama probleminde, daha hızlı veya optimum yolu elde edebilmek amacıyla çok çeşitli yöntemler kullanılmıştır. Yol seyahat süresinin ve mesafenin en aza indirilmesi, tehdit önleme, yakıt ve hava aracı karakteristikleri gibi kısıt ve amaçları dikkate alan çok kriterli çalışmaların geçmişi eskiye dayanmaktadır [110]. Dijkstra [54], A* [111], MILP [112], Doğrusal Programlama [113], Parçacık Sürüş Optimizasyonu [114], Karınca Kolonisi [115], Yapay Arı Kolonisi [116], GA [117], EA [118], DEA [119] bazı örnekler olarak ifade edilebilir. İHA yol planlamada öğrenmeye dayalı yöntemlerin de kullanıldığı çalışmalar bulunmaktadır [120, 121].

Bir yapay zekâ araması öncelikle aday çözümlerden oluşan bir dizi oluşturur. Algoritma, dizi içerisindeki uygun çözümü bulmayı hedeflemektedir. Arama stratejisine ve kullanılan bilgisayarın türüne göre yapay zekâ arama teknikleri şu şekilde sınıflandırılır:

- Tek yönlü tek süreçli,
- Çift yönlü tek süreçli,
- Tek yönlü çok süreçli,
- Çift yönlü çok süreçli.

Algoritmalar tek ve çift yönlü arama yapacak şekilde tasarlanabilirler. Örneğin; Meng, (2010) yol planlama çalışmasında, A* algoritmasını çift yönlü kullanmıştır. Performansı artırmak için, paralelleştirme teknikleri kullanılarak çok çekirdekli CPU veya GPU'lar kullanarak arama uzayında daha hızlı arama yapılabilir [122].

Karma Tamsayı Doğrusal Programlama (MILP) tabanlı yaklaşımların da örnekleri mevcuttur. Schouwenaars vd., (2001) MILP'nin yörünge planlama problemlerine uygulanabileceğini gösteren ilk araştırmacılarıdır [123]. 2B ortamda hareket eden bir araç için yapılan yol planlamasında zamanı modellemek için ayrık zaman adımları uygulamışlardır. Engeller ızgaralara hizalanmış dikdörtgenler olarak modellenmiştir. Problemin birden fazla adımda çözerek hesaplama karmaşıklığını azaltmak için uzaklaşan ufuk tekniğini kullanmışlardır. Bununla birlikte, bu tekniğin engellerin arkasına takılı kalma dezavantajı bulunmaktadır. Schouwenaars, How ve Feron, (2004) çalışmalarında İHA'yı, bir koordinat ekseninde dinamik ve kinematik güvenlik kısıtları ile ek doğrusal eşitsizlikler ve bir ayrık lineer model ile karakterize etmişlerdir [112]. King, Kuwata ve How, (2006) çoklu bir araç kontrol sisteminde, İHA'ların varış noktasına zamanında ulaşmasını sağlayan bir MILP tabanlı uzaklaşan ufuk planlayıcı kullanmışlardır [124]. Chen, Han ve Zhao, (2012) 3D uzayda statik veya dinamik engellerden kaçınan bir yol planlama için doğrusal programlama (LP) çalışması yapmış, gerçek zamanlı uygulamalarda daha iyi performans elde edildiğini belirtmişlerdir [113].

Yukarıda örnekleri verilen deterministik optimizasyon yaklaşımları, günümüzde teknolojinin ulaştığı yüksek hesaplama gücü ile yol planlama problemine uygulanabilmektedir. Optimal çözüm garanti edilebilir ancak hesaplama genellikle oldukça yavaştır. Bu nedenle çevrimdışı hesaplamalar için bir alternatif olduğu değerlendirilirse de çevrimiçi hesaplamalar için bu pek mümkün değildir.

İHA yol planlaması, çarpışmasız bir yol bulmanın yanı sıra belirli kriterlere göre en uygun yolu seçmeyi amaçlamaktadır. Çevresel ve İHA'ya ait kısıtların artması problem karmaşıklığını artırmaktadır. Optimizasyon problemlerinin boyutu büyüdükçe ortaya çıkan yüksek hesaplama ihtiyacı, deterministik algoritmalarından deterministik olmayan algoritmalara geçişi hızlandırmıştır [9]. Klasik optimizasyon yöntemlerinin zorlukları nedeniyle yol planlamada kullanılabilirliğinin azalması, esnek ve sağlam yöntemler olarak sezgisel ve metasezgisel algoritmaları ön plana çıkarmıştır. Metasezgisel

yöntemler, optimal bir çözüm bulmayı garanti etmese de optimale yakın çözümleri verimli bir şekilde bulmaktadırlar.

Optimizasyon algoritmaları, gerçek zamanlı 3B İHA yol planlama problemleri için uygun yöntemlerdir. Grafik tabanlı yöntemler, ilk önce statik ortamlarda uygulanmışlarsa da algoritmalarda yapılan değişiklikler ile göreceli hesaplama basitlikleri sayesinde gerçek zamanlı durumlara da uygulanmışlardır.

2.2.3.1. Sezgisel (buluşsal) ve metasezgisel algoritmalar

Bir buluşsal yöntemin, probleme kabul edilebilir bir çözüm bulmak için probleme özgü bilgilerden yararlandığı, meta-sezgisellerin ise bir tasarım konsepti olarak geniş bir sorun yelpazesine uygulanabilen genel algoritmalar olduğu ifade edilebilir [125].

Buluşsal algoritmaların yaygınlaşmasından önce bilgisayar bilimcisi Edsger W. Dijkstra'nın önerdiği algoritma yol planlama literatüründe bir dönüm noktasıdır [126]. Bir grafikte en kısa yolun hesaplanması problemini çözmeye yarayan referans yöntemlerin ilklerindendir. Dijkstra, özellikle kenarların aynı uzunluklara (ağırlıklara) sahip olmadığı durumlarda uygulanabilir. Başlangıç ve varış noktası arasındaki yolun uzunluğu, alınan kenarların uzunluklarının (veya ağırlıklarının) toplamına karşılık gelir. Algoritma çalışmaya başladığında her köşe ve kaynak arasındaki mesafe sonsuz olarak kabul edilmektedir. Yinelemeli şekilde kaynağa yakınlığına göre bir nokta seçilir ve alt grafiğe eklenir. Başlangıç ile alt grafiğin her bir köşesi arasındaki mesafe daha sonra güncellenir. Son düğüm seçildiğinde algoritma sonlanır.

Dijkstra algoritması, ihtiyaca göre geliştirilebilir. Örneğin yollar ağırlıklandırılarak karayolu trafiğine göre araç yol planlamasında ya da toplu taşıma için güzergâh üzerinde iniş-biniş sayısını en aza indirmek için düğüm değişikliğine bir maliyet eklenerek toplu taşıma yol planlamasında özelleştirme yapılabilir [127]. Dijkstra'nın araç yol planlamasında veya genelde engellerin olduğu her ortamda yaygın bir kullanım imkânı bulunmaktadır [128]. Bu algoritmanın zayıf yönlerinden birisi, işlemsel olarak sonsuz bir döngüye girme ihtimalidir. Bir diğer zayıf yönü de aynı anda en düşük maliyetli birkaç yolun, başlangıç noktasına olan mesafeleri içeren alt grafiğe girmesinin algoritmayı verimsiz hale getirmesidir [129]. Daha hızlı ve daha küçük ağırlıklı ideal yolları seçmek için Fibonacci tekniğini kullanan bir Dijkstra algoritması önerilmiştir [130]. Yaygın bir algoritma olan Dijkstra algoritmasında problemin azami çözüm süresi

$O(m + n \log n)$ 'dir (m kenarların sayısı ve n düğüm sayısı) [131]. Bilinen tanımlamada her adım, kendisinden önceki adımdan bağımsızdır. Ancak birçok yol probleminde olduğu gibi İHA'larda önceki adımlar değerlendirilmelidir. Bu da algoritmanın çözüm süresini artırmaktadır.

En az maliyetli yolu bulabilmek için genellikle, tüm mümkün yollar sıralanır. Dal ve sınır gibi yöntemlerde ise tam bir sıralamanın hesaplama maliyeti yüksektir [132]. Dijkstra algoritmasının dallanmış ve bir yığın ile bağlanmış olması büyük bir hız kazandırır. Bunu ötesinde bu maliyeti azaltmanın yolu umut verici olmayan yolların elenmesidir. A* böyle bir algoritmadır. Dal ve sınır yöntemindeki aksine bu avantajı ile arama ve optimizasyon problemlerinde sıklıkla kullanılmaktadır.

A* algoritması Hart, Nilsson ve Raphael, (1968) tarafından tanıtılmıştır [133]. A*, Dijkstra algoritmasının buluşsal özellikler ile geliştirilmiş bir varyantı olarak görülebilir. Dijkstra'nın en kısa yol algoritması prensipleri temelinde daha hızlı bir çözüm sağlamak amacıyla aramayı varış noktası yönünde yönlendirmeyi mümkün kılan ve bir sonraki düğüm için karar vermeye yardımcı olacak bir buluşsal işlev kullanmaktadır. Buluşsal olması daha hızlı olmasını sağlamaktadır ancak A*'ın buluşsal işlevi yeterli olmadığı takdirde doğru ancak Dijkstra'ya kıyasla alt-optimal bir çözüm üretebilmektedir.

A* bir ilk-en-iyi arama (Best-First-Search) yaklaşımıdır. Algoritma, $f(n)=g(n)+h(n)$ sezgiselinin ürettiği değerlere göre tercihlerde bulunur. $g(n)$, başlangıçtan bulunulan noktaya olan maliyet, $h(n)$ ise bulunulan noktadan varışa tahmini maliyet değeridir. Eğer her düğüm için tahmini $h(n)$ değeri, gerçek maliyeti geçmiyorsa ve bu düğümünden bir başka düğüme olan ve oradan da varış noktasına olan maliyetten küçük veya eşit ise A* her zaman optimal yolu bulmaktadır [30].

A* çevrimdışı bir algoritma olarak ortaya çıkmıştır ancak çevrimiçi uygulamalarda da kullanılmaktadır. A* algoritmasının en önemli dezavantajı bellek kullanımındır, tüm ziyaret edilmemiş düğümleri hafızada tutar. RBFS (Recursive Best-First Search) ve MA* (Memory-Bounded A*) algoritmaları bu dezavantajın üstesinden gelmek için oluşturulmuştur [30].

A* algoritmasının birçok varyantı önerilmiştir. Bilinmeyen ortamlardaki yolları hesaplamak için Stentz, (1997) tarafından tanıtılan D*, (dinamik A*) A* ile aynı şekilde çalışır ancak kenarların maliyetleri ortamı keşfettikçe değişebilir [134]. D* bu nedenle

bir yerel yöntem gibi çalışmaktadır. A*'ın başka bir varyantı olarak Theta* algoritması, bir ağdaki en kısa yolları hesaplamak için tasarlanmıştır [135]. Theta*, görüş hattına dayanan A* üzerine kurulu herhangi bir açıda çalışmaya imkân veren varyanttır. Newaz, Pratama ve Chong, (2013) her yinelemede dört yön sınırlaması ile ağ içinde bir keşif önermektedirler. Yöntem, küresel bir yerelleştirme olmadan bilinmeyen bir ortamda kullanılabilir [136]. Nash, Koenig ve Tovey, (2010) tarafından geliştirilen LazyTheta* adlı başka bir varyant, görüş hattı kontrollerinin sayısını ve planlama süresini önemli ölçüde azaltmaktadır [137]. Bu algoritma, bir İHA'yı çalıştırmak için dış ortamlarda gerçek zamanlı kullanılmıştır [138].

A* algoritması ve varyantları, aynı bölgede hareket eden çoklu araç yol planlamasına uyarlanmıştır [139]. Szczerba vd., (2000) Sparse A* Search (SAS) algoritmasını tanıtmış; çalışmada asgari uçuş bacağı, azami yatış açısı, uçuş uzunluğu, vektörden varış pozisyonuna düzeltilmiş yaklaşım kısıtları kullanmışlardır [41]. Brener vd., (2004) diyagram ve ızgara tabanlı ayrıştırmaı optimale yakın bir sonuç için gerçek zamanlı bir arama içinde kombine etmişlerdir [84]. Tehditler içeren bir ortamda 3D ızgara tabanlı ayrıştırma ile A* algoritması kullanılmıştır.

A * Prune algoritması, uygun olmayan yolları atmak için buluşsal yöntemler kullanmaktadır ancak yolların uzunlukları arttıkça en kısa yolların bulunacağı garanti edilmez [140].

Garip vd., (2018) statik bir ortamda çoklu araç yol planlamada A* ve GA'yı birlikte kullanan bir algoritma önermişlerdir. A*'da düğümlerin ilişkisini değerlendirmek için kullanılan buluşsal işlev, GA'da bir amaç işlevi olarak kullanılmaktadır [141].

Buluşsal yöntemleri kullanan bir diğer arama yöntemi olan K* yöntemi, ağırlıklı grafikte kısa yolları bulabilmek için arama yaparken tüm yol alternatiflerini sırayla geçmek yerine sezgisel bir tahminde bulunur [142]. Algoritmanın her adımında A* ve Dextra algoritmalarının bir kombinasyonu kullanılır.

Artımlı buluşsal arama algoritmalar, varsa önceki aramalardan elde edilen bilgileri yeniden kullanabilir. Böylece, benzer bir planlama problemini en başından itibaren planlama yapmaktan daha hızlı çözebilirler. Kesilmiş Artımlı Arama yöntemleri olan (T)D*, (T)D* Lite ve (T)LPA* bu kapsamda geliştirilmiş algoritmalar [143].

Herhangi bir yinelemeli algoritma konsept olarak, algoritma kendisini sonlandırmadan bitirilse dahi geçerli bir çözüm sunabilir. Benzer şekilde, bu arama metodu A* ve varyantlarına entegre edilmiştir. Herhangi bir zaman yöntemlerinin, A* ve D* üzerindeki entegrasyonu olarak; Anytime D* (AD*), Anytime Repair A* (ARA*) ve Anytime Tree Restoring A* (ATRA*) algoritmaları önerilmiştir [144]. Bu teknikler, alt-optimal hedef sınırlarına dayalı aramanın genişletilmesine odaklanmak için önceki yinelemelerden gelen bilgileri kullanmaktadır [143, 145]. A* algoritması, varyantlarının gelişimi ve diğer tekniklerle entegrasyonu yoluyla çeşitli alanlarda kapsamlı bir şekilde uygulanmaktadır.

Korf, (1990) tarafından tanımlanmış RTA*, A* algoritmasının bir online versiyonudur [146]. RTA* ve A* arasındaki temel fark, $g(n)$ fonksiyonun A* dan farklı olarak başlangıçtan düğüme olan maliyeti değil, düğüm ile önceki düğüm arasındaki maliyeti temsil etmesidir. RTA* ziyaret edilen tüm düğümleri ve tahmini $h(n)$ değerlerini saklar. D* algoritması, A* algoritmasının dinamik ortamlara uyarlanmış şeklidir.

Undeger ve Polat, (2007) RTTES (Real-Time Target Evaluation Search) algoritmasını önermişlerdir [147]. Algoritma, engellerin sınırlarını ve en umut verici yönü değerlendirmek için bir sezgisel kullanmaktadır.

Likhachev ve Stentz, (2006) kısmen bilinen ortamlarda arama uzayını daraltmak için PPCP (Probabilistic Planning with Clear Preference) algoritmasını önermişlerdir [148]. Algoritmada temel prensip olarak, ortamın bilinmeyen bölgeleri için tercih edilen değerler önceden tanımlanabilir.

Doğadan esinlenen birçok meta-sezgisel optimizasyon algoritması mevcuttur ve özellikle yol planlama problemlerinde başarılı sonuçlar ortaya koymaktadır. Karınca kolonisi optimizasyonu (ACO) algoritması, doğal karınca kolonileri metaforuna dayanmaktadır [149]. Gerçek hayatta karıncalar görsel bir bilgiye sahip olmadan sadece feromon bilgisini kullanarak bir besin kaynağından yuvalarına en kısa yolu bulma yeteneğine sahiptir. Karıncalar yürürken yere feromon bırakırlar ve varsa diğer karıncalar tarafından bırakılmış feromonu takip ederler. Daha fazla feromon bulunan yolların diğer alternatif yollara kıyasla seçim ihtimali artmaktadır. Algoritmada her karınca yoldaki feromon miktarına göre stokastik olarak seçim yapmaktadır. Feromon miktarının artırılıp,

azaltılması belirli kriterlere göre gerçekleştirilir. ACO, çok sayıda İHA yol planlaması araştırmasında çalışılmıştır.

Zhou vd., (2011) kendinden uyarlamalı optimize edilmiş, genetik operasyonlar ve yumuşaklık yöntemi ile Geliştirilmiş Karınca Kolonisi Algoritması (I-ACA) kullanarak gerçek zamanlı dinamik bir yol planlama uygulamasını gerçekleştirmişlerdir [115].

Yapay arı kolonisi yaklaşımı (ABC) doğadan esinlenerek üretilmiş ve İHA yol planlama problemlerinde kullanılan bir diğer yöntemdir [150]. ABC algoritması, diğer popülasyon tabanlı algoritmalara kıyasla daha az kontrol parametresi olan bir optimizasyon tekniğidir. Lei ve Shiru, (2012) başlangıç popülasyonunu oluşturmak için keşif arısı kullanmış, karşıtlık tabanlı öğrenme ve lojistik harita kullanan standart ABC için başlangıç aşaması farklı bir strateji önermişlerdir [116]. Xu, Duan ve Liu, (2010) küresel optimum yolu bulma amaçlı çalışmasında, karmaşık savaş alanı ortamında, yapay arı kolonisi algoritması kullanmışlardır [151]. Zhang, Wu ve Wang, (2011) hava muharebe görevleri kapsamında Uygunluk-Ölçeklemeli Kaotik Yapay Arı Kolonisi algoritması ile CABC, PSO ve GA'dan daha iyi sonuçlara ulaşıldığını belirtmişlerdir [152].

Ateşböceği yaklaşımı (FA) 2008 yılında Yang, (2010) tarafından ateşböceklerinin parlaklık davranışından esinlenilerek sunulmuştur. Algoritmanın en belirgin avantajı, aynı anda bireyler arasındaki karşılıklı etkiyi dikkate alarak bireylerin rastgele hareketinin kullanılmasıdır [153].

Hidalgo-Paniagua vd., (2017) statik engellerle mobil robot yol planlaması için FA tabanlı bir algoritma tanıtmışlardır. Bu algoritma ile yol uzunluğu, düzgünlüğü ve güvenliği gibi hedeflere ulaşmışlardır. Yöntem, ortamdaki tekil robot için en kısa çarpışmasız yolun bulunması için kullanılmıştır [154].

Alweshah ve Abdullah, (2015) sınıflandırma problemini çözmek için hibrit bir FA yaklaşımı sunmuşlardır [155]. Statik ve dinamik ortamlarda çok boyutlu sırt çantası problemlerinin çözümü üzerinde testler yapılmıştır. Önerilen yaklaşımın etkinliği GA ve DEA ile karşılaştırılmıştır.

Sutantyo vd., (2013) FA yaklaşımına dayalı olarak bir sualtı mobil robot için çevrenin kısmen bilindiği gerçek sualtı yol planlaması sunmuşlardır [156]. Bir diğer çalışmada, SIHA yol planlama problemi için yeni bir FA varyantı önerilmiştir [157].

Çekirge Optimizasyonu yaklaşımı (GOA), çekirgelerin yiyecek arama davranışından esinlenmiştir [158]. Çekirgeler en büyük sürülerden kabul edilmektedir. Çekirgenin yaşam döngüsü larva ve yetişkin olarak iki temel aşamadan oluşur. Larva evresindeki genç çekirgelerin temel özelliği yavaş hareketleri ve küçük adımlarıdır. Algoritmada optimizasyon aşamaları, çekirgelerin hareket tarzından esinlenerek uygulanmaktadır.

Elmi ve Efe, (2021) yol planlama çalışmasında GOA kullanan iki yaklaşım sunmaktadırlar [159]. Öncesinde engelleri algılamak için bir duyuşal sistem ve sonrasında hızlı bilinmeyen dinamik engelleri tahmin eden bir yöntem geliştirilmişlerdir.

Doğadan esinlenen bir diğer yöntem olan Parçacık Sürü Algoritması (PSO) da İHA yol planlamasında kullanılır. PSO yöntemi, boids modelinden esinlenerek Eberhart ve Kennedy, (1995) tarafından önerilmiştir [160]. Düşük hesaplama maliyetli optimizasyon yöntemi; finans, iletişim ağları veya yol planlama gibi çok çeşitli alanlarda kullanılmaktadır [161].

PSO, temelde sürü halindeki hayvanların (kuşlar, balıklar vb.) bilgi paylaşımı, bulundukları ortama uyum sağlama, yırtıcılardan uzak durma vb. sosyal davranışlarından esinlenen bir sürü zekâsı yaklaşımıdır. PSO algoritması, rastgele oluşturulmuş çoklu parçacıklarla başlatılır. Her parçacık rastgele bir hız alır, yinelemeler ile arama uzayında optimuma ulaşabilir. Her yinelemede her parçacığın bireysel olarak ulaşmış olduğu en iyi durum ve sürüye ait en iyi çözüm kaydedilir. Sonraki pozisyonların hesaplanmasında bu iki bilgi dikkate alınmaktadır.

PSO bütüncül bir yaklaşım ile de kullanılabilir. PSO yöntemi doğrudan araç yol planlamaya uygulanabilir yani parçacıkların konumu araçların konumunu temsil etmektedir. Araçların belirli bir varış noktasına ulaşması istenebilir [161]. Bunun yanında filo yol planlaması için de kullanılabilir [162].

Foo vd., (2006) 3D PSO çalışmasında ilk yolu, başlangıç ve varış noktası arasında düz bir yol olarak kabul ederler [163]. Çalışmada, B-spline eğrileri kullanılmıştır. Li, Sun ve Xu, (2006) 2D PSO uygulamasında, yapılandırma alanını eşit kare hücre ızgara yöntemi ile ayırtmışlardır [80].

Faz açısı kodlanmış ve kuantum-davranışlı parçacık sürüsü optimizasyonu (theta-QPSO) isimli iki yeni PSO varyantı, Fu, Ding ve Zhou, (2012) tarafından önerilmiştir.

Çalışmada, bu yeni algoritmalar, GA, PSO ve DE ile karşılaştırılmışlardır [164]. theta-QPSO'nın diğer yöntemlerden daha performanslı olduğu gösterilmiştir.

Huang ve Tsai, (2011) genelde çok sayıda yineleme gerektiren PSO için yakınsama sürecini hızlandırmada bir GA kullanılmasını önermektedirler [165]. Yöntemde, parçacıkların çözüm kalitesini değerlendirildikten sonra, en iyi genel çözüm bulunmadan ve her parçacığın konum ve hızı güncellenmeden önce bir GA kapsamında çaprazlama ve mutasyon operatörleri uygulanır. Daha sonra elde edilen yol düzleştirilir.

Tavlama benzetim yöntemi, İHA yol planlaması çalışmalarında kullanılmıştır [166, 167]. Ryan vd., (1999) İHA yol planlama çalışmasında Tabu arama kullanmışlardır [168]. Rebollo, Ollero ve Maza, (2007) önerdikleri yöntemi robotlar üzerinde Ağaç Arama ve Tabu Arama olarak iki adımda uygulamışlardır [169].

Zadeh, (1965) karar verme, örüntü tanıma, otomatik kontrol ve veri sınıflandırması gibi yüksek belirsizlik, karmaşıklık ve doğrusal olmama durumlarında kullanılabilen bulanık mantığı tanıtmıştır [170]. Bulanık mantık teorisi, insanın algısal bilgilerle sonuç çıkarım yeteneğinden esinlenmiştir. Bulanık kural tabanlı mantık, akıl yürütme ve karar verme yoluyla elde edilen kurallar için sistematik bir strateji sunmakta, mantıksal çıkarımlar kullanarak eksik ve yanlış bilgi içeren belirsiz sistemleri modellemek için kullanılmaktadır.

Engellerden kaçınmak için Zavlangas ve Tzafestas, (2003) bulanık tabanlı bir kontrolör sunmuşlardır [171]. Yol planlama probleminde iyi tanımlanmış hedefler kümesi elde etmek için problem, bulanık mantık kuralından oluşan basit bileşenlere ayrılmıştır. Sensörler, çarpışmayı önleme ve çevre haritası oluşturmada önemli bir rol oynar. Bulanık mantık tabanlı bir mobil robot, sensör verilerini kullanarak karar vermekte ve bilinmeyen bir ortamda yol alabilmektedir.

Uzman bilgisi ve insan müdahalesi gerektirmesi bulanık kurallar ile yol planlamanın önemli bir sorunudur. Castellano, Attolico ve Distanto, (1997) insan ve makine öğrenimi tekniklerini kullanarak otomatik bulanık kuralların üretimi için bir yöntem önermişlerdir [172].

Sabo ve Cohen, (2012) İHA için 2D yol planlayıcı çalışmasında bulanık çıkarım sistemi ile gerçek zamanlı olarak engeller ve varış noktası ile ilgili bilgi alarak baş açısı ve İHA hızında değişiklikleri üretecek bir sistem tasarlamışlardır [173].

Zhuoning vd., (2010) hesaplanan yolun sürekli olmama veya yerel optimuma düşme sorunlarını önlemek için sanal kuvvetlere ve bulanık mantığa dayalı bir modeli İHA yol planlamada uygulamışlardır [174]. Bilinmeyen engellerin dikkate alınması gerektiği ortamlarda, belirlenen tehlike seviyesini temsil eden bir parametre kullanan sanal kuvvetler ile engellerden kaçınma gerçekleştirilir. Parametrenin değeri, gereken reaktivite seviyesi ve platformun mevcut durumuna (arıza ihtimali, kalan yakıt vb.) göre bulanık mantık kullanılarak belirlenmektedir.

Bulanık mantık, mobil robotlar için çevrimiçi bir çarpışmadan kaçınma stratejisi olan dinamik pencere yaklaşımı (DWA) ile de kullanılmıştır. DWA'nın arama penceresine karşılık gelen hareket hızı ve engellere korunacak mesafe gibi çeşitli kriterler bulunur. Her bir kriterin önemi şartlara göre değişebilir. Hong vd., (2015) kriterleri dinamik olarak ağırlıklandırmak için bulanık mantık kullanmayı önermişlerdir [175].

Motlagh, Hong ve Ismail, (2009) tuzak olarak nitelendirilebilecek durumlara tepki geliştiren reaktif bir yol planlama uygulaması için bulanık bir yaklaşım geliştirmişlerdir [176].

Yapay sinir ağı (ANN), akıllı bir sistem ve matematiksel araç olarak hesaplama, tıp, mühendislik ve ekonomi gibi farklı alanlarda kullanılmaktadır. ANN, insan beynini model almaktadır. Sinir ağı, yapılandırılmış bir grafik topolojisine kümelenmiş ve ardışık katmanlardan oluşan sinaptik ağırlık bağlantılarıyla birbirine bağlanan nöron adı verilen hesaplama bileşenlerinden oluşmaktadır. Sinaptik ağırlıklar genellikle farklı sayısal değerlerle ilişkilendirilir. Böylece, ANN'nin ele alınan problem temelinde davranışını değiştirmesine ve uyarlamasına izin vermektedir [177].

Yapay sinir ağını Mcculloch ve Pitts, (1943) tanıtmışlardır [178]. 1980'lerde daha hızlı bilgisayarların ortaya çıkması, sinir ağı ve öğrenme algoritmalarındaki gelişmeler ile yeni becerilerin keşfedilmesi, yaygın bir kullanıma imkân vermiştir.

Sinir ağları, yol planlama uygulamaları için öğrenme ve genelleme yeteneği, üst düzey paralellik, dağıtılmış görüntüleme ve hata toleransı gibi özellikler sağlamıştır. Janglová, (2004) mobil robot navigasyonu için belirsiz bir ortamda sinir ağının kullanımını tanıtmıştır [179]. Çarpışmasız yol oluşturmak için sinir ağları iki aşamada kullanılır. Birinci aşamada çevreden gelen algılama verileri kullanılarak boş alanlar bulunmakta; ikinci aşamada çarpışmasız, güvenli ve en kısa yol üretilmektedir.

Wang vd., (2008) engellerden kaçınmada üçüncü nesil sinir ağları olarak ifade edilen doğrusal olmayan sınıflandırma ve yüksek boyutlu kümeleme problemlerini çözmeye elverişli bir yöntem olan İğnecikli Sinir Ağları'nı (SNN) kullanmışlardır [180].

Li, Song ve Hou, (2015) ANN tabanlı Hızlı Eşzamanlı Yerelleştirme ve Haritalama (Hızlı SLAM) kullanımı ile belirsiz bir ortamda mobil robot yol planlaması geliştirmişlerdir [181]. Bu yaklaşım, SLAM'ın doğrusal olmayan fonksiyonunun yanlış doğrusallaştırılması ve yanlış odometri modeli tarafından oluşan hata birikimini ortadan kaldırır. Capi, Kaneko ve Hua, (2015) kentsel alanlarda GPS ve pusula sensör verilerinin ANN ile değerlendirildiği gerçek zamanlı bir yol planlama uygulaması gerçekleştirmişlerdir [182]. Medina-Santiago vd., (2014) ultrasonik sensörler kullanarak gerçek zamanlı mobil robot navigasyonu için sinir ağına dayalı bir kontrol sistemi sunmuşlardır [183]. Abubaker, (2012) statik engellerle birlikte karmaşık ve kalabalık ortamlarda mobil robot yol planlamada sinir ağı ve bulanık mantık tabanlı bir yaklaşım tanıtmıştır [184].

Yol planlamada kullanılan doğadan esinlenen optimizasyon algoritmalarına diğer bazı örnekler şunlardır:

Biyocoğrafya Tabanlı Optimizasyon (BBO), göç ve mutasyon olarak iki adımda küresel optimumu aramaktadır [185]. BBO, küresel yakınsamayı iyileştirmek için kaos teorisi ve avcı-av kavramını benimsemiştir.

Gri kurt optimizasyon algoritması (GWO), 2D SİHA yol planlama problemini çözmek için kullanılmıştır [186]. 2D planlama ile tehdit alanlarından kaçınarak güvenli ve asgari yakıt maliyeti öncelikli yolu bulabilmektedir.

Duan ve Huang, (2014) emperyalist rekabetçi algoritma (ICA) önerdikleri çalışmalarında SİHA'nın optimal yol planlaması için ANN'ye dayanan küresel hibrit bir yöntem kullanmışlardır [187]. Yöntemin, evrimsel hesaplamanın belirsizliğini azalttığı ve yerel optimuma düşmekten kaçındığı gösterilmiştir.

2.2.3.2. Genetik Algoritma

Evrimsel Algoritma'lar (EA), doğadan esinlenen bilgisayar tabanlı bir problem çözme algoritmasıdır. Bir başlangıç popülasyonunda, belirli bir sorunun çözümlerini temsil eden bireyler, hayatta kalma ve çoğalma için birbirleriyle rekabet eder. Bireyler,

uygun olmayanların elendiği doğal seçim ile nesilden nesile genetik değişiklikleri biriktirerek adapte olmaktadır. EA ve evrim teorisi arasındaki kurulan benzerliklere karşın örtüşmeyen kavramlar da bulunmaktadır. Örneğin, EA'larda amaç fonksiyon bilindiği için birey genlerinin hedefe yakınsaması kontrollü bir doğal seçim ile sağlanmaktadır. Evrim teorisinde ise bireylerin ulaşacakları hedef durum bilinmeden gen değişimlerinin oluştuğu ve bazen en iyilerin yok olduğu bir doğal seçim söz konusudur. Bununla birlikte EA genel olarak, bireyler arasındaki karmaşık etkileşimleri görmezden gelmektedir.

EA, karmaşık süreksiz ve kısıtlı optimizasyon problemleri ile başa çıkabilen stokastik bir arama tekniğidir. EA; türevlenemez, doğrusal veya dışbükey olmayan, büyük boyutlu, çoklu optimallik veya belirsizlik içeren, büyük hesaplama gerektiren çok çeşitli problemlerin çözümünde elverişli bir kullanıma sahiptir. Bu tür çok amaçlı optimizasyon problemlerinde çözümler; pareto cephesi olarak da adlandırılan, çözüme ilişkin farklı kriterler arasında denge kuran, eşdeğer baskın olmayan uzlaşmalar kümesinde -pareto-optimal kümede- temsil edilmektedir [188]. EA, küresel optimum çözümü bulmayı garanti etmez ancak pareto optimal çözümler üretebilir. Bu çözümler, tüm kriterler değerlendirildiğinde arama uzayında başka hiçbir çözümün daha iyi olmadığı çözümlerdir.

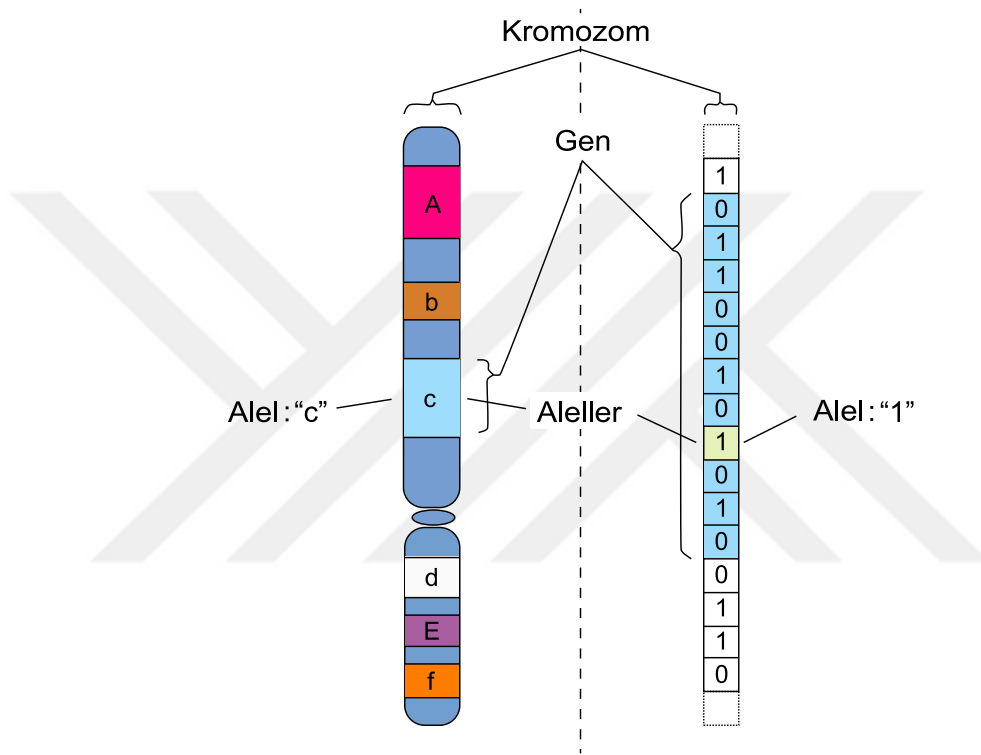
Sürü zekâsı ve EA, yerel arama yöntemlerinden daha sonra ortaya çıkmıştır. Bir EA kapsamında GA, stokastik ışın arama denilen yerel arama tekniğin bir varyantı olarak kabul edilebilir [30]. Popülasyon tabanlı bu teknikler, yol planlama problemlerinde başarıyla uygulanmaktadır.

Holland (1992) tarafından önerilen GA; çok boyutlu, arama alanının geniş ve karmaşık olduğu, problemin belirli bir matematiksel modelde ifade edilemediği, kesin bir çözüm yönteminin olmadığı durumlarda kullanılabilecek etkili ve kullanışlı stokastik bir optimizasyon yöntemidir [190, 191].

GA; yapay zekâ, bilgisayar bilimi ve yöneylem araştırması alanlarının kesiştiği bir araştırma alanıdır. Problemden bağımsız kullanılabilme yeteneği, GA'yı optimizasyon alanında yaygınlaştırmıştır. GA, statik optimizasyon alanında kendisini göstermiş sonrasında dinamik optimizasyon problemleri alanına da uygulanmıştır [189].

Popülasyon içinde her birey bir çözümü temsil etmektedir. Bazı bireyler yok olurken bazıları sonraki nesle aktarılır. Bazı çiftler yavru verir. Her birey DNA'sının bir kısmını yavruya aktarır, yavru DNA'sı, ebeveynlerin DNA'sının bir kombinasyonudur.

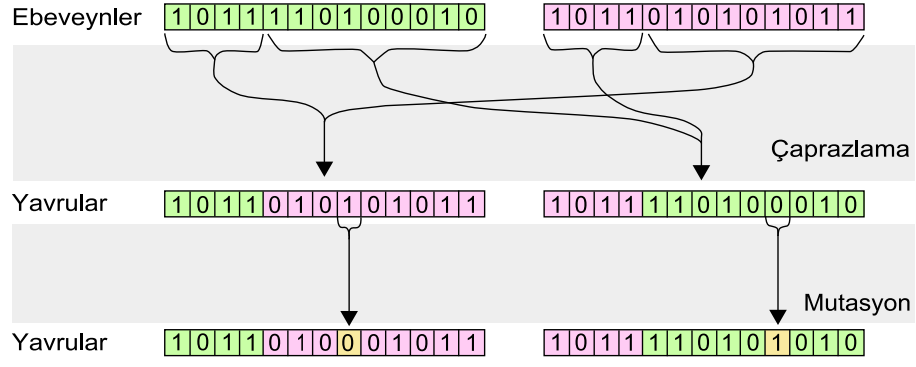
Aday çözümler, biyolojik sistemlerin kromozomlarına benzer şekilde kodlanır. Bir biyolojik kromozom ve hesaplama kromozomu Şekil 2.13'te gösterilmiştir [192]. Kromozomlar bir araya gelerek popülasyonu oluşturmaktadır.



Şekil 2.13. Biyolojik kromozom (solda); hesaplama kromozom (sağda)

Her kromozom, bir aday çözümdür ve amaç fonksiyonuna göre bir uygunluk derecesi bulunmaktadır. Probleme uygunluk derecesine göre en uygun kromozomlar yeni nesle aktarılır [189]. Aday çözümün kalite ölçümü amaç fonksiyonu ile gerçekleşir. Amaç fonksiyonu, matematiksel bir model veya bir bilgisayar simülasyonu olabilir [193].

Genel olarak GA, belirlenen ilişkiler ve belirli kurallar çerçevesinde çaprazlama (rekombinasyon) ve mutasyon işlemleri ile aday çözümün iyileştirilmesi prensibine dayanmaktadır. Çaprazlama, aday çözümlerin (bireylerin) parçalarının birleşiminden yeni bir aday çözüm oluşturulması işlemidir. Mutasyon ise, bir aday çözümün bir parçasının değişimi ile yeni bir çözüm haline gelmesi işlemidir. Algoritmanın sonlanmasında farklı kriterler tercih edilebilir. Ebeveynlerin oğul oluşturma süreci Şekil 2.14'de gösterilmiştir [192].



Şekil 2.14. Ebeveynler ve çocuklar

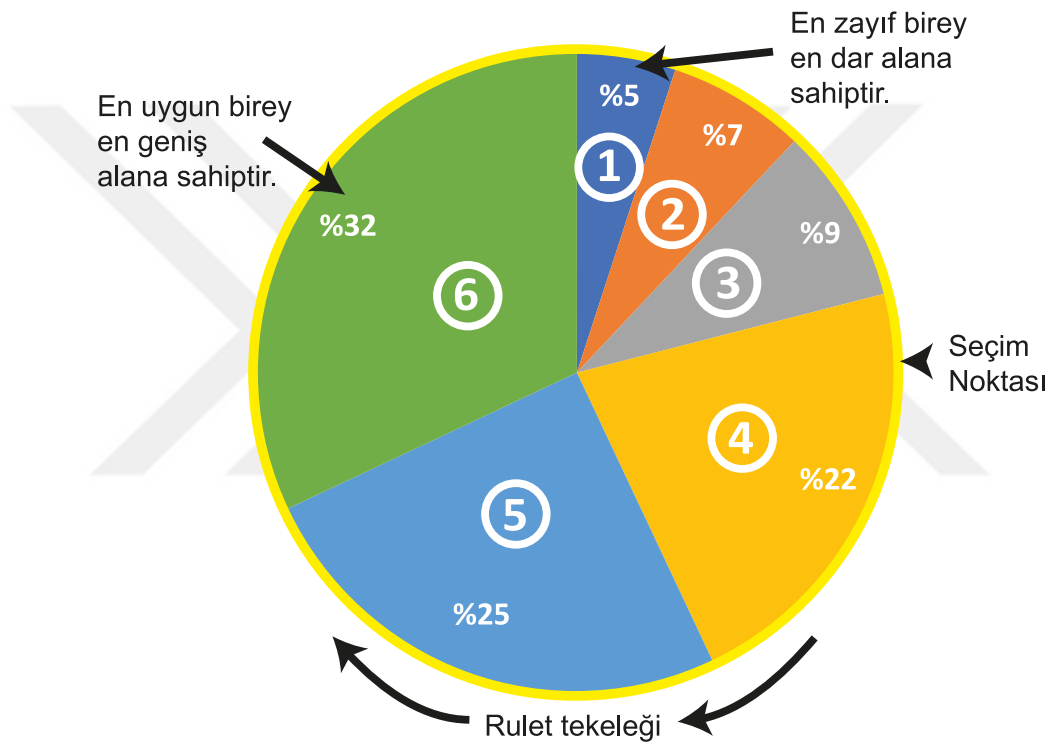
Seçim operatörü, çaprazlama için bireylerin belirlenmesini ve oluşacak yeni birey sayısını belirlemektedir. Seçim yöntemi, genel olarak daha iyi bireylerin seçilme şansını artırarak yeni oluşan yavruların daha iyi uygunluk sonuçları vermesini sağlayan bir filtrelemedir. Bunun yanında, kötü bireylerin popülasyon dışına atılarak yeniler için alan açılması da seçim sürecinin bir diğer amacıdır.

Seçim kriterleri, yüksek uygunluk derecesine sahip çözümlerin seçilme derecesi olan seçim baskısını belirlemektedir. Seçim baskısı çok düşükse, iyi ebeveynlerden gelen bilgilerin popülasyona yayılma hızı düşer ve süreç verimsizleşir. Seçim baskısı çok yüksekse, benzer iyi bireyler popülasyonu devralır ve gelişim için gereken çeşitlilik azalmaktadır. Popülasyondaki tüm bireyler yüksek düzeyde benzerlik gösterir, bu düşük çeşitlilik durumu yerel bir optimuma yakınsamaya neden olmaktadır. İstenmeyen bir seçim süreci sonucu popülasyon, yerel bir optimumda sıkışıp kalabilmektedir. Buna karşın, popülasyon çeşitliliğindeki aşırılık, GA verimliliğinin dramatik bir şekilde azalmasına da yol açabilmektedir. GA'nın tasarımı ve uygulanmasında en iyi bireylerin çoğalması ile arama uzayının alternatif bölgelerinin keşfi arasındaki dengenin sağlanması önemlidir. İyi seçim stratejileri, çeşitliliği hızlı kaybetmeden popülasyondaki yüksek uygunluktaki bireylerin yayılmasına izin vermektedir. Genel olarak, popülasyon çeşitliliğini korumak küresel optimumu garanti etmektedir.

GA literatüründe çeşitli seçim mekanizmaları bulunur ancak probleme özgü bir seçim yöntemi tanımlanmamıştır [194]. Genelde üç temel seçim mekanizması bulunmaktadır: rulet tekerleği seçimi, turnuva seçimi, sıra tabanlı rulet tekerleği seçimi [195].

Turnuva seçim mekanizması ile rastgele iki veya daha fazla birey arasından en iyi uygunluk derecesi olan birey seçilmektedir. Turnuva seçiminde, yeni nesildeki birey

sayısı, eski neslin birey sayısına eşit oluncaya kadar her seferinde popülasyondan iki birey seçilir [196]. Turnuva seçiminde en küçük uygunluk değerine sahip ebeveynin seçilme ihtimali bulunmazken rulet tekerleği yönteminde her bireyin seçilme ihtimali görecelidir. Bireylerin uygunluk derecesi ile orantılı olan bir seçim yöntemi rulet tekerleğidir. Bu yöntem, uygunluk derecesi yüksek bireylerin bir sonraki nesle taşınma ihtimalini artırmaktadır. Şekil 2.15'de görüldüğü gibi bireyler, uygunluk değerleri ile orantılı bir seçilme ihtimaline sahiptir [25]. Ters rulet tekerleği yönteminde bireyler, uygunluk değerleri ile ters orantılı bir seçilme ihtimaline sahip olmaktadır.



Şekil 2.15. Rulet tekerleği seçim yöntemi

Ebeveynler seçildikten sonra daha iyi bireyler oluşturmak için çaprazlama yapılır. Seçilen iki kromozom karşılıklı çaprazlanmaktadır. Çaprazlama, GA'nın en önemli aşamalarından biri olarak kabul edilmektedir. Rekombinasyon terimi daha geneldir ancak çaprazlama terimi daha sık kullanılmaktadır. Çaprazlama farklı yöntemlerle gerçekleştirilebilir. Tek noktalı çaprazlama, kromozomlar üzerinde rastgele bir nokta seçer ve yeni bireyler oluşturmak için dizideki bu noktadan itibaren tüm öğeleri karşılıklı değiştirir. İki noktalı çaprazlamada ise kromozom üzerinde iki nokta seçilir ve bu iki nokta arasındaki veya dışındaki öğeler karşılıklı kromozomlar arasında değiştirilir. Düzgün çaprazlama yöntemiyle, kromozomlar rastgele n noktaya bölünerek çaprazlanmaktadır.

Çaprazlamanın aritmetik yeniden birleştirme veya harman yeniden birleştirme gibi başka türleri de bulunmaktadır [197]. Basit aritmetik çaprazlamada seçilen bir k noktasına kadar tüm aleller iki kromozomun yavrularında aynı şekilde kalırken k noktasından sonraki aleller, iki ebeveynin aritmetik ortalamaları olarak iki yavruda da ortak olmaktadır. Tek aritmetik çaprazlamada, rastgele bir k aleli konumunda iki ebeveynin aritmetik ortalaması hesaplanır, diğer aleller değiştirilmez. Tam aritmetik rekombinasyonda, yavrulardaki her alel, her iki ebeveyndeki alellerin aritmetik ortalamasının değerini almaktadır. Blend Crossover (BLX-a), ebeveynler tarafından oluşturulan bölgelerden daha geniş bir bölgede yavrular oluşturma bir yoludur. Bu fazladan boşluk, ebeveynler arasındaki mesafe ile orantılıdır. Simüle Edilmiş İkili Çaprazlama (SBX) operatörü, gerçek sayı çözümlerini temsil için tek nokta ikili yeniden birleştirmeyi simüle etmeye çalışmaktadır. Tüm operatörlerin hedefi seçilmiş ebeveynlerden yeni genetik yapıya sahip daha iyi kromozomlar oluşturmaktır.

Mutasyon, mevcut kromozomdan yeni bir kromozomun üretildiği mekanizmadır. Mutasyon genellikle, çaprazlama sonrasında bireylerin mutasyon oranı olarak da ifade edilen popülasyonun belirli bir yüzdesine uygulanır [189]. Kromozom çeşitliliğini artırmada etkisi görülmektedir.

Değiştirme operatörü, GA'nın bir diğer aşamasında görev alır. Seçim, rekombinasyon ve mutasyon yoluyla oluşan yavrular, bazı kriterlere göre popülasyona alınır. Seçkin (elitist) yer değiştirme, nesil bazında yer değiştirme ve kararlı durum değiştirme yaygın değiştirme teknikleri arasındadır.

Hayatta kalanların seçimi mekanizması, bir GA popülasyonundaki k ebeveyn ve p yavrudan oluşan popülasyonu, k birey sayısına kadar azaltılması sürecini yönetmekten sorumludur. Belirli sayıda veya kriterde düşük uygunluk değerine sahip ebeveynler elenerek yerlerini yüksek uygunluk değerine sahip oğullar almaktadır. Bireylerin uygunluk değerine göre karar verilmesi en yaygın yöntem olmakla beraber, GA'da popülasyonun yaşına dayalı olarak ebeveynlerin atıldığı ve yavruların kaldığı basit bir eleme de yapılabilir. Zayıf uygunluk değerine sahip çözümlerin sonraki nesillere aktarılması popülasyona genetik çeşitlilik sağlayarak çözümlerin kalitesini iyileştirebilir [198]. Seçim operatörünün etkinliği kadar değiştirme operatörünün etkinliği de popülasyonun sınırlı bir çözüm bölgesinde sıkışıp kalarak erken yakınsamaya yol açmasını önlemelidir.

Devralma süresi, iyi bir bireyin tüm popülasyonu ele geçirmesi için geçen süre şeklinde tanımlanmaktadır. Herhangi bir varyasyon operatörü olmadan yalnızca seçim etkisi altında en iyi bireyin yayılım oranı deneysel olarak tahmin edilebilir. Panmiktik (tek-standart karışım) popülasyonda standart seçim yöntemleri için teorik devralma süreleri türetilmiştir [199]. Bu süreler ile popülasyon büyüklüğü arasında logaritmik bir orantı bulunmaktadır. Daha yüksek seçim baskısında daha kısa devralma süresi ve dolayısıyla daha sömürücü algoritmalar ortaya çıkmakta, seçim baskısı azaldığında ise algoritmalar daha keşfedici hale gelmektedir.

Elitizm yaklaşımı, popülasyondaki en iyi çözüm veya çözümleri sonraki nesillere aktarmak için kullanılmaktadır. Değiştirme operatörü en iyi çözümü ele aldığında uygunluk değerleri karşılaştırılan herhangi bir yavru, daha kötü veya eşitse değiştirme yapılmaz, yavru atılır ve iyi birey tutulur. En kötü çözümlerin değiştirilmesi yaklaşımı elitizm ile benzerlik göstermektedir [198].

Tablo 2.7. *GA terimleri*

Terim	Açıklama
Popülasyon	Bir dizi birey
Birey/ Kromozom	Soruna aday bir çözüm / çözümün genetik gösterimi
Gen	Kromozomun bir ögesi
Alel	Genin alabileceği değer
Uygunluk değeri (fitness)	Bireyin çözüm olarak kalitesini gösteren bir değer
Seçim	Popülasyondan çaprazlama için bireyleri seçme işlemi
Çaprazlama/rekombinasyon	İki yeni yavru elde etmek için seçilen iki ebeveynin bilgilerinin belirli kurallara göre değiş tokuş edilmesi/düzenlenmesi işlemi
Mutasyon	Kromozomdaki bir veya daha fazla geni değiştiren işlem
Değiştirme	Yeni nesilde hangi bireylerin kalacağına karar verme işlemi

GA’da yaygın kullanılan terimler Tablo 2.7’de gösterilmiştir. GA, önlem alınmadığında karakteristik olarak çözüme yakınsarken popülasyon çeşitliliğini kaybetmektedir. Bu sorunun önüne geçebilmek için çeşitli yöntemler önerilmiştir, bu yöntemler dört alt başlıkta listelenebilir [189].

- 1) Bir değişiklikten sonra çeşitliliği artırmak: Hiper mutasyon yöntemi [200],
- 2) Çalışma boyunca çeşitliliği korumak: Rastgele göçmen şeması: [201], paylaşım veya kalabalık mekanizmaları [202], ve termodinamik GA [203],

- 3) Hafıza temelli yaklaşımlar [204-206],
- 4) Çoklu popülasyon yaklaşımları [207-210].

Zamansal kısıtları karşılamak için optimal yolun elde edilmesinin yanı sıra varış noktasına farklı yollardan ulaşan optimuma yakın farklı aday yolların elde edilebilmesi işlevseldir. Çoğu yol planlama algoritmasından farklı olarak GA, alternatif yolların üretilmesi açısından oldukça uygundur [68]. Bu özellik tekil yol planlamada faydalı olmakla beraber çoklu yol planlama için kritiktir.

Rastgele sayılar kullanarak oluşturulan başlangıç popülasyonu [117, 213-216], tek noktalı çaprazlama [213, 216], sezgisel çaprazlama [117], yeni genetik operatörler [213], tek tip mutasyon [216], tek noktalı mutasyon [117, 213, 214] tek tip olan ve olmayan mutasyon operatörlerinin birlikte [117] kullanılması GA'ların genel çerçevede aynı oldukları halde küçük dokunuşlar ile çok farklı sonuçlar verebildiğinin bir göstergesidir.

GA, İHA'larda belirli bir irtifada (2D) [211] veya değişken irtifada (3D) [212], yasak bölgeler içeren bir haritada arazi topolojisini dikkate alan çevrimdışı, çevrimiçi ya da gerçek zamanlı bir yol planlama yapabilmektedir [117].

Cekmez, Ozsiginan ve Sahingoz, (2013) GA kullanarak İHA'lar için en uygun uçuş yolunu bulan yeni bir yöntem geliştirmeyi amaçlamışlardır [217]. Önerdikleri yöntemde 2B haritada, GA'ya ekledikleri ön bilgiler ile paralel yaklaşım kullanarak daha düşük hesaplama süresi elde etmişlerdir.

Nikolos ve Brintaki, (2005) çoklu İHA'ları göz önüne alan çalışmalarında, bilinen farklı başlangıç pozisyonlarındaki araçlar için yörünge boyunca belirli hız dağılımına sahip ardışık yol noktalarından oluşan 2D yörüngeler oluşturmuşlardır [218]. Çalışmada Diferansiyel Evrimsel Algoritma (DEA) kullanılmıştır. Nikolos, Zografos ve Brintaki, (2007) aynı yöntemi Radyal Taban Fonksiyonları ve ANN ile desteklemiş ve İHA yol planlaması için RBF-ANN yöntemini önermişlerdir [219].

Sahingoz, (2013) çoklu İHA'lar için geliştirdiği GA'sında, kromozomlar için fonlar belirlemiş ve yol düzeltirme için Bézier eğrileri kullanmıştır [220].

Roberge, Tarbouchi ve Labonte, (2013) 3D çözüm uzayında; radar kısıtları, yakıt tüketimi ve ortalama irtifa faktörlerini değerlendiren amaç fonksiyonuna sahip PSO ve GA'yı birlikte gerçek zamanlı kullanmışlardır [9]. Her iki algoritma da araç dinamiğine

göre uygulanabilir ve yarı optimal yörüngeler üretmiştir. GA'nın PSO'ya kıyasla, ortalama irtifa ve tehlike bölgelerinden kaçınma açısından daha iyi yörüngeler ürettiği gösterilmiştir.

Özalp ve Sahingoz, (2013) fiziksel kısıtları dikkate alan GA'yı, açık kaynaklı bir ortam tarayıcısı olan World Wind kullanarak 3D ortamda uygulamışlardır [212].

Men, E ve Yao, (2012) Voronoi diyagramları ile oluşturulmuş yollar üzerindeki noktalarda gerçek savaş ortamında vurulma ihtimallerini hesaplamışlardır [65]. Yollar GA kullanılarak seçilmiştir.

Liu vd., (2012) trafik bilgileri toplama amaçlı çoklu İHA uygulamasında GA kullanmışlar ve görev için gereken İHA sayısında %31 azalma gerçekleştirmişlerdir [221]. Trafik bilgileri toplama amaçlı bir diğer çalışmalarında egemen olmayan sıralama operatörünün kullanıldığı GA ile seyir mesafesi %13 azaltılmış, tespit edilen hedef sayısı ise %42 artırılmıştır [222].

Kok, Gonzalez ve Kelson, (2012) İHA için FPGA ile donanım tabanlı yol planlama mimarisi önermişlerdir [118]. Çözüm için GA kullanılmıştır.

Fu, Zhou ve Hu, (2012) okyanus ortamında adaların tehdit kabul edildiği uygulamada gerçek sayı kodlama ile aday yolları temsil etmiş, yol maliyetinin matematiksel modelini kurmuş ve DEA ile yollar üretmişlerdir [223]. Bir diğer çalışmada, İHA'nın yer merkezli radar siteleri tarafından tespit edilmesini önlemek amaçlı çevrimiçi yol planlama çalışmada GA kullanmışlardır [224].

Zheng, Feng ve Lu, (2011) yerel optimuma takılmayı azaltan yakınsama hızını geliştiren bir Parçacık GA önermişlerdir [225].

Pakkan ve Ermiş, (2010) çalışmalarında GA tabanlı bir yaklaşım geliştirmişlerdir [226]. Google haritaları sunucusundan alınan koordinatları MATLAB kullanarak veritabanına aktarmak için bir kullanıcı arayüzü tasarlanmıştır. Algoritma, 2D ortamda her bir İHA için üretilen yolları optimize etmeye çalışmaktadır.

Nikolos I. K., (2001) çalışmada aday yol çözümlerini temsil etmek için farklı kodlama teknikleri kullanmaktadır [227]. Yollar için farklı mutasyon mekanizmaları ile GA-tabanlı çözümler önerilmiştir. Eğri parçaların parametrelerini değiştirmiş, mutasyon,

çaprazlama, yayma, hedefe varma ve mutasyon-eşleştirme gibi farklı operatörler uygulamışlardır.

Mittal ve Deb, (2007) çok amaçlı bir GA ile güvenliği en üst düzeye çıkarmak ve yol uzunluğunu en aza indirmek için yeni bir algoritma önermişlerdir (NSGA-II) [228]. Üretilen yol, B-Spline eğrileri kullanılarak temsil edilmiştir.

Bir diğer çalışmada Oz, Topcuoglu ve Ermis, (2013) GA tabanlı yol planlayıcısında farklı hedefleri dikkate alan azami irtifa kısıtı ile bilgiye dayalı yeni çaprazlama ve mutasyon operatörleri önermişlerdir [229].

Cheng, Sun ve Liu, (2011) çalışmasında Immune Genetic Algorithm (IGA) ile kabul edilebilir bir çözümün daha hızlı sağlandığını göstermişlerdir [230]. Algoritmanın, iyi oğulları yeni nesilde kullanılmak üzere koruduğu belirtilmiştir.

GA tabanlı çoklu İHA görev atama çalışmasında Shima ve Schumacher, (2008) çoklu araç ve çoklu hedefleri temsil etmek için matris gösterimi önermişlerdir [231].

Ergezer ve Leblebicioglu, (2013) azami bilgi toplama amaçlı hedefleri gözetleme problemini, TSP problemine benzer bir şekilde çözmüşlerdir [232]. İstenilen-bölgeye-it (PTDR), yasak-bölgeden-çek (PFFR), ve son-noktaya-it (PTFP) evrimsel operatörleri ile bir insan operatörü deneyimi taklit edilmeye çalışılmıştır.

Sun vd., (2012) İHA yeniden planlama çalışmasında kapsama alanı boşluklarının giderilmesini içeren yol planlama problemini GA kullanarak çözmüşlerdir [233].

Qingtian, Wenjing ve Jia, (2012) çalışmasında uyarlanmış amaç fonksiyonu, çaprazlama ve mutasyon operatörleri kullanılan GA ile 15 yinelemede geçerli sonuç, 50 yinelemede ise optimale yakın sonuca ulaşıldığını belirtmişlerdir [234].

Pierre, Zakaria ve Pal, (2012) önerdikleri yeni algoritma (MSPVEGA) ile GA'nın paralel uygulanması ile optimal sonuçlara ulaştıklarını göstermişlerdir [235].

2.2.3.3. Hücresel Genetik Algoritma

Genellikle doğrusal olmayan ve kombinatoriyal problemleri çözme yetenekleriyle öne çıkan cGA, arama uzayının hem ayrık hem de sürekli değişkenleri birleştirdiği problemlerde iyi performans göstermektedir. Evrimsel yöntemlerle çok amaçlı optimizasyonda çözüm kümesinin (veya pareto kümesinin) mümkün olduğunca doğru bir

şekilde bulunması istenmektedir. Pareto cephesi önceden tahmin edilemez olduğundan nesiller boyunca popülasyon çeşitliliğini korumak önemlidir. Çeşitliliğin korunmasını destekleyen yapılandırılmış popülasyon yapısı ile cGA çok amaçlı optimizasyon problemlerinde etkin bir şekilde uygulanabilmektedir.

GA, çok amaçlı optimizasyon problemleri için popülasyon tabanlı bir tekniktir. Başarılı bir çözüm için popülasyon içinde çeşitliliğe ihtiyaç duymaktadır. Eğer popülasyon çeşitliliği çözüm optimumuna ulaşmadan önce kaybolursa, ihtimaller dar bir alana sıkışır ve alternatif çözüm bölgelerine ulaşmak zorlaşır. Bu durum genetik operatörlerin artık ebeveynlerden daha iyi yavrular üretemedikleri zaman meydana gelmektedir.

Ana genetik operatörler, çaprazlama ve mutasyonun popülasyon içi çeşitliliğin korunmasındaki payı, seçim ve değiştirme operatörlerinden daha az değildir. Her şeye rağmen, GA erken yakınsama nedeniyle başarısız olabilir. Gelişen popülasyonda çeşitliliği teşvik etmek ve dengelemek için kalabalık şemaları [237], uygunluk paylaşımı gibi bir dizi teknik önerilmiştir [238]. Arama sürecine dayatılan yapay kısıtları ile oluşan işlem yükü, bu tür tekniklerin olumsuz bir yan etkisidir. Popülasyonun merkezden uzaklaştırılması, çeşitliliğin korunması ve geliştirilmesi için doğal bir yöntem olarak ön plana çıkmaktadır.

GA'da mekânsal bir tasarım için en önemli motivasyonlardan birisi, algoritmanın çoklu işlem yaklaşımına uyarlanması ihtiyacıdır. Bu konsept ile beklenenin ötesinde bir performans avantajı elde edilebilmiştir. Mekânsal popülasyon yapısına sahip GA'nın işlemsel paralellik sonucu çözüm süresinin azalmasından ziyade çözüm kalitesinin iyileşmesi ön plana çıkmıştır [239].

Paralel Genetik Algoritmalar (PGA) dört ana kategoride sınıflandırılabilir [240]:

- 1) Küresel prosedürler: Tek popülasyon kullanılır ancak genetik operatörler paralel olarak yürütülmektedir.
- 2) Kaba taneli: Aralıklılarla bireylerin değiş tokuş edildiği bağımsız alt popülasyonlar şeklinde çalışmaktadır.
- 3) İnce taneli: Seçim ve çaprazlamanın bireyin çevresi ile sınırlandırıldığı mekânsal bir popülasyon yapısını tanımlamaktadır.
- 4) Hibrit modeller: Tekniklerin çeşitli kombinasyonlarıdır.

PGA'nın popülasyonun genel davranışı üzerinde bir etkisi olduğu açıktır. İnce taneli hücrel modellerde, mekânsal olarak tanımlanmış küçük mahallelerdeki iyi çözümler yavaşça yayılırken devralma süresi uzamakta ve popülasyon çeşitliliği korunmaktadır.

cGA, yapısal paralelleştirme yaklaşımında ince taneli kategoride sınıflandırılabilir bir türdür. Difüzyon modeli olarak da isimlendirilir. GA'nın geleneksel uygulamalarında her birey, bütünsel bir mekânda yer alan aday bir çözümdür. Genetik kalıtım ilkesi taklit edilerek popülasyon içinde her birey bilgi alışverişi yapabilmektedir. cGA'daki temel özellik ise, bireylerin ayrılmış bir yapıda bulunmasıdır [236]. Her ayrılmış yapı hücre olarak isimlendirilmekte ve komşu hücreler ile kurallı bir etkileşimine izin verilmektedir.

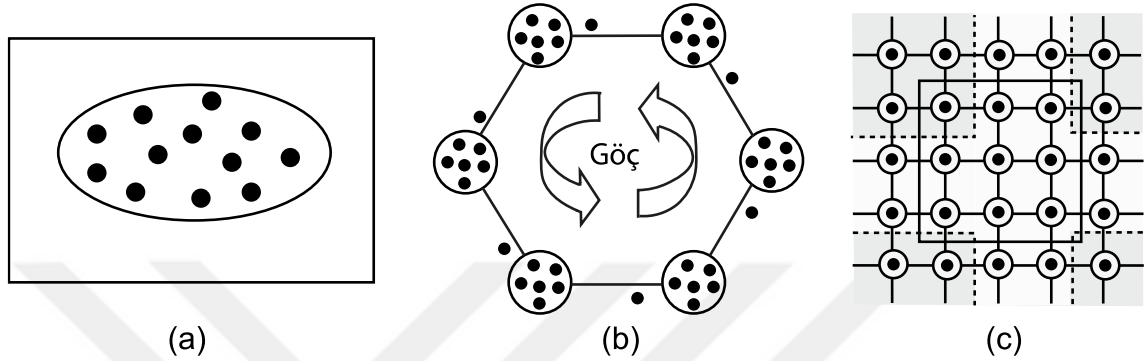
cGA, GA gibi bir problem kapsamında aday çözümlerden oluşan popülasyonu geliştirerek en iyi çözümü aramaktadır. Ancak salt popülasyon odaklı bir perspektiften ziyade yapısal olarak sınırlandırılmış bir yaklaşım uygulanır. Bir cGA'da popülasyon, ızgara şeklinde biçimlendirilmektedir.

cGA'da her birey mekânsal yapılandırılmış uzayın bir mahallesinde bulunmakta, genetik operatörler her hücrenin kendi mahallesinde yerel olarak çalışmaktadır. “Hücrel Genetik Algoritma” terimi Whitley, (1993) tarafından önerilmiştir ancak bu anlayıştaki algoritmalar [242, 243] daha önce de çalışılmıştır [241]. Dağıtılmış popülasyon kullanılması, popülasyon izolasyonunun genetik farklılaşmayı büyük oranda artırdığı fikrine dayanmaktadır [244]. Benzer yaklaşımlar genelde, PGA çerçevesi içinde sınıflandırılabilir [245]. cGA, ince taneli GA [242] ve güçlü PGA [246] olarak da adlandırılmaktadır.

cGA, bireyler arasındaki etkileşimin nasıl belirlendiğini ve popülasyon çeşitliliğinin nasıl korunduğunu modelleyen bir teknik oluşturur. cGA'da bireyin komşusu haricinde popülasyon içi etkileşimi bulunmaz.

Birçok çalışma [247-250], cGA'nın panmiktik GA'dan daha iyi performans gösterdiğini bildirmektedir. Bunun nedeni, cGA'nın popülasyonda daha yüksek bir çeşitliliği koruması böylece, arama uzayının potansiyel her bölgesinin kullanımında başarılı olmasıdır. cGA'da bireyler yerel olarak etkileşime girdiğinden, uygunluğu yüksek bireylerin genetik bilgisi tüm popülasyona yavaşça yayılmaktadır.

Çoğu GA, panmiktik popülasyon kullanmakta ve operatörler bir bütün olarak uygulanmaktadır. Bunun tersine popülasyonun bir şekilde merkezi olmadığı yapılandırılmış birçok GA türü arasında, dağıtılmış ve hücresel modeller iki popüler varyanttır [245]. Şekil 2.16'da Panmiktik, dağıtılmış ve cGA'nın yapısı gösterilmiştir [251].



Şekil 2.16. Panmiktik (a), dağıtılmış (b) ve hücresel (c) GA

cGA konsepti, Hücresel Otomat'lardan (CA) etkilenmiştir. Hücresel Otomatlar (CA'lar), Neumann, (1966) tarafından kendini yeniden üretebilen ve organize edebilen sistemler olarak tanıtılmıştır [252]. CA, her biri basit hesaplamalar yapan, hücreleri ızgara şeklinde sıralanmış bir sistem şeklinde ifade edilebilir. Her hücrenin en yakın komşusuna bağlı olmasıyla bilgi alışverişi gerçekleşir. Tüm hücrelerin işlevsellikte aynı özellikte olmalarına rağmen verilerdeki farklılık sistemin karmaşık davranışını ortaya çıkarmaktadır [253].

CA'ların gerçek problemlere uygulanmasında ortaya çıkan zorluklardan biri uygun kuralların tasarımıdır. Küresel hesaplama gerektiren bir problemi yerel etkileşimlere dayanan bir sistem üzerinde kurgulamada zorluklardan bahsedilebilir. Bununla birlikte, EA temelinde bir çeşit CA modeli kullanılarak, küresel problemlere çözüm üretilebileceği gösterilmiştir [254].

cGA'da, CA benzeri bir hücresel yapı bulunmakta ancak hücre değerlerini güncellemek için farklı bir yöntem kullanılmaktadır. CA'da hücrelerin yeni değeri, belirli kurallar yardımı ile hücrenin geçerli değeri ve komşu hücrelerin değerlerine göre güncellenirken cGA'da hücre değeri, amaç fonksiyonuna göre her yinelemede geleneksel bir GA popülasyonunda olduğu gibi çaprazlama ve mutasyon yapılarak güncellenmektedir.

cGA, popülasyonun H düğüm uzunluğunda ve V düğüm genişliğinde bir ızgara olarak düzenlendiği ve bu ızgaranın her bir düğümünün belirli bir bireyi temsil ettiği yapılandırılmış algoritmalar sınıfına aittir [255]. cGA'nın en önemli özelliği olan popülasyonun mekânsal yapısının; halka, 2B ızgara, toroidal örgü ve 3D küp şekillerinde örnekleri bulunmaktadır [256].

cGA daha önce çeşitli problemlerde uygulanmıştır: Araç rotalama problemi [257], gerçekleştirilebilirlik (satisfiability) problemleri (SAT) [258, 259] ve çok amaçlı optimizasyon [260] bunlardan bazılarıdır.

Mantere, (2009) bir görüntü sınıflandırma ve sıralama problemini, TSP'ye benzeterek cGA ile çözmüştür [256]. Alba ve Dorronsoro, (2004) çalışmalarında azami süre ya da mesafe kısıtı içeren bir kapasite kısıtlı araç yol planlama (VRP) problemini cGA ile çözülmüşlerdir [257]. Anılan çalışmalarda mahalle yapısı kromozomların özelliklerine göre oluşturulmamış, kromozomlar mahallelerde rastgele yer almıştır.

Folino, Pizzuti ve Spezzano, (1998) cGA'nın optimuma yakınsama hızını artırmak için yerel arama ile melezleştirilmiş hücresel bir memetik algoritma önermiş, bir cGA'daki mutasyona uğramış yavrulara ek bir yerel arama aşaması uygulamışlardır [259]. Popülasyondaki tüm bireyler üzerinde kapsamlı bir yerel arama, hesaplama açısından pahalı olabileceğinden bazı çalışmalarda [261] yerel arama, bireylerin bir alt kümesiyle sınırlandırılmıştır.

cGA'da mahalleler arasında kurallar da tanımlanabilir. cGA popülasyonunda, daha iyi bireylerin aynı mahallelere doğru hareket etmesi için bir hiyerarşi oluşturulabilir [262]. Bu şekilde en iyi bireylerin daha hızlı ve daha etkin bir şekilde yakınsaması sağlanırken uzak konumlarda çeşitlilik korunur. Hiyerarşik bir cGA, standart bir cGA'ya benzer ancak her nesilden sonra mahallelerin bitişik hücreleri arasında uygunluk değeri temelli bir takas işlemiyle popülasyon yeniden düzenlenir.

Bir cGA, bir hücreden başlar ve sonlandırma koşulu karşılanana kadar genetik operatörleri kullanarak art arda güncellemeler ile ilerler. Bir cGA'daki bir hücrenin güncellenmesi şu anlama gelir: Her nesilde bireyin mahallesinden iki ebeveyn seçilir, genetik operatörler uygulanır ve elde edilen yavrunun uygunluk değeri daha yüksek ise değiştirilir. Güncelleme zamansal olarak eşzamanlı veya eşzamansız yapılabilir [263]:

- Eşzamanlı (senkron) cGA, tüm güncellemelerin paralel olarak aynı anda yapılmasıdır. Yavrular geçici bir yardımcı popülasyonda depolanır. Her yineleme sonunda mevcut popülasyonun tüm bireyleri, yardımcı popülasyondan gelenlerle bir kerede değiştirilir. Güncelleme tüm bireylere aynı anda uygulanırsa sonraki nesildeki bireylerin hepsi aynı anda oluşturulduğundan cGA eşzamanlıdır [236].
- Eşzamansız (asenkron) cGA'da bireyler sıralı olarak yavru oluşturulduktan hemen sonra yeni nesil güncellenir. Böylece, herhangi bir yardımcı popülasyon ihtiyacı olmaz ve yeni genetik yapı daha hızlı nüfuz eder. Bu yeni bireyler, ebeveynlerinin nesline ait olanlarla etkileşime girebilir. Eşzamansız cGA, popülasyonu, eşzamanlı cGA'dan daha hızlı birleştirir. Eşzamansız güncellemenin birçok yolu bulunmaktadır. Yaygın kullanılan dört eşzamansız güncelleme yöntemi aşağıdaki gibidir [264]:
 - Tekdüze tarama: Güncellenecek hücre rastgele seçilmektedir.
 - Sabit hat taraması veya satır taraması: Hücreler sol üst köşedeki hücreden başlayarak sırayla soldan sağa ve satırdan satıra güncellenmektedir
 - Sabit rastgele tarama: Rastgele bir sıra dizisi üretilir ve bu sabit sıraya göre hücreler sırayla güncellenmektedir.
 - Yeni rastgele tarama: Her tarama öncesinde sabit rastgele taramada olduğu gibi yeni bir rastgele sıralama oluşturularak hücreler bu sırayla güncellenmektedir.

2.2.4. İşlem sonrası ve yol yumuşatma teknikleri

Yol planlama; engellerden kaçınma, azami dönüş yarıçapı, azami tırmanma hızı, hızlanma veya yavaşlama yeteneklerinin vb. hesaplanmasını içermektedir. Otonom sistemlerde araç kısıtlarını karşılamayan ve güvenli şekilde takip edilemeyecek keskin manevralar içeren yollar kullanılamaz. Yol boyunca araca veya uçuş güvenliğine zarar verebilecek manevralardan sakınılması gereklidir. Bu kısıtları hesaba katmayan yollar, riskli kabul edilmektedir.

Yukarıda ayrıntılı açıklandığı gibi grafikler tipik olarak bir arama problemi şeklinde çözülerek en kısa yollar bulunabilir. Bulunan yollar son işleme teknikleri ile geliştirilebilir ancak yolların iyileştirilmesi garanti edilmez.

Oluşturulan yol, araç kısıtlarını karşılamalıdır. İHA'lar hız veya hareket yönünü anında değiştiremez. Ayrıca, İHA'nın hızına ve yatış açısına göre bir dönüş yarıçapı bulunmaktadır. Yörünge, gereksinimleri karşılayan düzgün bir eğri olmalıdır.

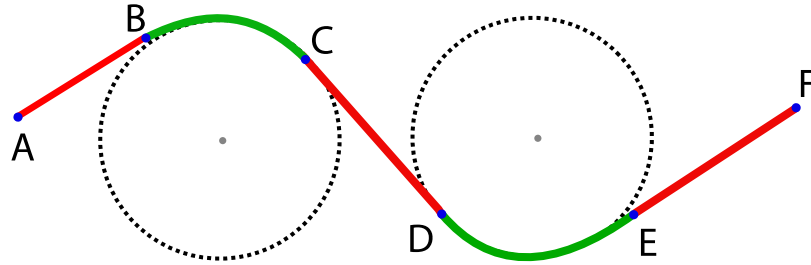
Bir yol, birleşik eğrilerden oluşmaktadır. Bu bakış açısıyla yörüngenin keskin bölümleri, eğrilerin birleşim bölgelerinde düzeltilir. Yolların keskin bölümlerini yumuşatmada aşağıdaki yöntemler kullanılmaktadır:

- Dubins eğrisi yöntemi ile kesik bir çizginin köşesine bitişik çizgi bölümlerinin parçaları, bir dairenin yayı ile değiştirilmektedir [265]. Bu durumda yol düzleştirilmiş olmasına rağmen düzgünlük işlevi, süreksiz bir işlev gibi görünmektedir.
- Polar polinomlar yönteminde kutupsal koordinatlarda bir açıda bir polinom ile verilen düzgün yol, eğrilik fonksiyonunu keserek yolu yumuşatmaktadır.
- Kübik Bézier eğrilerinde, çoklu çizgi bağlantılarında seçilen dört bağlantı noktası kullanılmaktadır. Ortaya çıkan düzleştirilmiş yörüngenin eğriliği bir süreksizliğe sahip olacaktır. Kübik eğrileri kullananların aksine, çift türevlenebilir Bézier eğrisi yedi kontrol noktası kullanmaktadır ve yumuşatılmış yollar, eğrilik fonksiyonunda süreksizliklere sahip değildir [266]. İHA'lar için düzgün, dinamik, uygulanabilir yörüngeler oluşturulmasında Bézier eğrileri yaygın olarak kullanılmaktadır [267]. Bézier eğrisinin kullanılmasının bir avantajı, yolun tam bir geometrik tanımını kullanmaktan ziyade nispeten daha az sayıda parametre kullanarak küçük bir hesaplama maliyeti ile temsil edilebilmesidir.
- Kübik B-spline, eğrilikte sürekli düzgün bir değişimin elde edilmesini sağlamaktadır [268]. En az beş bağlantı noktasının kullanılması gerekirken eğrinin şeklini seçmede istenen serbestlik derecesine göre kontrol noktalarının sayısı artırılabilir.

2.2.4.1. Dubins eğrisi

Lester Dubins, basit bir uçağın zaman açısından optimal yolları üzerinde araştırma yapan ilk kişidir. Geliştirdiği model Dubins eğrisi olarak adlandırılmıştır [24]. Dubins eğrileri şu şekilde listelenir: LSL, RSR, RSL, LSR, RLR ve LRL. Burada S düz doğru parçası anlamına gelir, L sola doğru dairesel bir yaydır ve R sağa doğru dairesel yay anlamına gelir. 2B düzlemde İHA sağa veya sola dönebilir. Böylece, bir konfigürasyonda

sağa döner ve R olarak işaretlenir, diğerinde sola döner ve L olarak işaretlenir. Şekil 2.17'de gösterildiği gibi iki teğet daireye sahiptir. Sınırlı dönüş yarıçapına sahip bir araç için zaman açısından en uygun yörüngeler oluşturulur [38].



Şekil 2.17. Dubins eğrisi

Dubin'in köşe tekniği, yol planlamanın parçası değilse de bir yol oluşturma tekniğidir. Yol vektörlerini yumuşatmada yaygın bir kullanıma sahiptir. Yaklaşım, birbirine bağlı iki çizgi parçası arasındaki köşenin dairesel bir yay ile değiştirilmesini gerektirir. Bir yol noktası seti ile aracın dönüş yarıçapına uygun olarak bir yolu daha uygulanabilir hâle getirebilir [269]. 2D bir düzlemde iki konfigürasyon arasında dairesel yay ve bunların teğet çizgilerinin birleşimi olduğu kullanılabilir en kısa erişim süresine sahip yol bu yöntemle oluşturulabilmektedir.

2.3. Yol Planlamada Hava Aracı Performansı

Yol planlayıcılardan, ortamdaki belirsizliklerin değerlendirildiği; engeller, hedefler ve İHA modeline özgü kısıtlar varlığında yollar oluşturmaları istenmektedir. Yol planlayıcı, hesaplama gücü ve enerji açısından kaynakları verimli kullanılmalıdır [270].

Harcanan enerjinin ihmal edilmesi, kısa veya hızlı yollar üretebilir ancak daha fazla yakıt tüketimine neden olmaktadır. Bazı durumlarda aracın mevcut yakıt ya da batarya gücü ile tamamlayamayacağı bir yol üretilebilir. Gerçekte bir araç, görev esnasında sabit bir oranda enerji tüketmez çünkü her hız ve ağırlıkta seyir, yükselme ve alçalma yakıt karakteristiği farklıdır. Aracın enerjiyi sabit bir oranda tükettiği varsayıldığında görev süresinden önce toplam enerji tükenebilir. Aracın enerji tüketimi manevralara göre de değişiklik gösterebilir. Bu nedenle, ideal bir planlama yakıt sarfiyatı dikkate alındığında mümkün olabilmektedir.

Küçük İHA'ların önemli bir sınırlılığı dayanıklılığı ve menzili sınırlayan yerleşik enerji kapasitesidir. İHA'ların kısıtlı olan enerjisi, hareket organlarındaki ihtiyacın yanı sıra; navigasyon, algılama ve veri işleme gibi ihtiyaçları da karşılamalıdır [271]. Bu

nedenle, görev süresini uzatmak ve kapsama alanını genişletmek için ağ oluşturmada enerji verimliliğini geliştirmek bir öncelik olarak kabul edilmektedir [272, 273].

İHA'ların enerji tüketimini en aza indirmek literatürde yer alan önemli konulardandır. Tran, Hajisami ve Pompili, (2017) İHA ağlarında enerji verimli için bir programlama çerçevesi tanıtmışlardır [274]. Langelaan, yörüngeyi rüzgârlardan faydalanacak şekilde optimize etmiş ve uçuş süresini ve menzili artıran bir yöntem önermiştir [275, 276].

Zeng ve Zhang, (2017) İHA'nın hızı, yönü ve ivmesinin bir fonksiyonu olarak tahrik enerjisi tüketimi üzerine teorik bir model ile bir İHA'nın optimal hareket yörüngesini belirlemek için bir analitik çerçeve önermiş, hız ve enerji optimizasyonuna dayalı olarak İHA'lar arasında iletişim için enerji verimli yörüngeler üretmişlerdir [277]. Floreano ve Mondada, (1996) çalışmalarında bir khepera robotu, genetik bir algoritma ile yalnızca yerleşik sensörlerinden gelen bilgileri kullanarak hesaplamaların uzak makinede yapıldığı yol planlamasında pil seviyesini dikkate alarak varış noktasına ulaşabilmiştir [278]. Williams ve Yakimenko, (2018) İHA'nın pil yönetimi ve değişimi için bir mobil hava gözetim platformu önermişlerdir [279]. Başka bir çalışmada “sonsuz uçuş” fenomeni için bir otomatik pil değişim yöntemi önerilmiştir [280]. Menzil, faydalı yük ve yatay hız çerçevesinde batarya süresi odaklı bir model incelenmiştir [281].

İletişim için harcanan güç tüketimini en aza indirmek enerji verimliliği için çalışılan konulardandır. İletişimin İHA konumuna göre programlanması için bir Döngüsel Çoklu Erişim (CMA) şeması kullanılmış ve bir İHA ile iletişim kuran farklı yer terminallerine zaman tahsisleri optimize edilerek her düğümün verimi yükseltilmeye çalışılmıştır [282].

Yakıt ekonomisini artırmak için çok çeşitli çalışmalar yapılmıştır. Yol planlama çalışmaları bu konuda büyük öneme sahiptir. Öncelik, operasyonellikten ödün vermeden en ekonomik güzergâh ile uçuşun gerçekleştirilmesidir. Uçuş bölgesindeki rüzgârın yakıt ekonomisine etkileri [283], formasyon uçuşları [284] yakıt ekonomisini artırmak için yapılmış çalışmalardandır.

İHA'larda yol planlamanın en temel amacı en kısa güzergâhı hızlı bir şekilde bulabilmek olarak tanımlandığından aslında kısa bir güzergâhtan kimi zaman daha önemli olan ekonomik olma durumu çoğu çalışmada değerlendirilmemiştir. Kısa yol ekonomiktir genel yargısı yanlışla sebep olabilmektedir. Alternatif yollar arasından seçilen güzergâh

en kısası olsa bile hava aracı performansı değerlendirildiğinde daha uzun yolların daha ekonomik olduğu görülebilmektedir.

ATR, yayınladığı yakıt ekonomisi konulu raporunda, (2011) daha yüksek seyir irtifası, düşük hızlı tırmanma, düşük itki ile alçalma gibi yöntemler ile daha az yakıt sarfiyatını örneklendirmiş, ATR 72-500 ile 8 dk daha uzun uçuşta, 111 kg yakıt tasarrufu yapıldığı gösterilmiştir [285].

Mayer, (2002) Eurocontrol'da yayınlanan 71 temel uçak verisine göre yörünge planlaması yapmıştır [286, 287]. Li vd., (2012) uçuş planlaması ile ilgili çalışmalarında, navigasyon veritabanı, coğrafi bilgi sistemi ve çeşitli uçakların gerçek performans verilerine dayanarak hesaplanan bilgisayarlı uçuş rotası parametrelerini, uçuş rotalarının hesaplanmasında kullanmışlardır [288].

Labonte, (2015) pervaneli bir uçağın sabit hızda ihtiyaç duyduğu yakıt miktarını, Ding vd.'nin (2006) yaklaşımı ile dördüncü dereceden Runge-Kutta yöntemi kullanarak tahmin eden bir diferansiyel denklem ile yüksek hassasiyette göstermişlerdir [289, 290].

Konuyla ilgili yayınların çoğu, harcanan yakıtın yaklaşık tahmini üzerine kurulmuştur. Genel olarak hava aracına özgü uçuş performans özellikleri yerine genel varsayımlar kullanılmıştır. Bu nedenle hesaplama hassasiyeti düşüktür [290].

3. MATERYAL VE YÖNTEM

Bir İHA için özelleştirilmiş performans özellikleri temelinde kinematik kısıtlara uyumlu ekonomik ve havada kalma süresini uzatan yollar üretilebilmesi için ilk aşama bir İHA'nın belirlenmesidir. Bu bölümde; İHA seçimi, seçilen İHA'nın yakıt karakteristiğinin elde edilmesi, elde edilen verilerin yazılıma nasıl yansıtıldığı açıklanmaktadır. İHA'larda yol planlama problemine daha önce uygulanmamış, GA'nın bir türü olan cGA ve scGA detaylandırılırken amaç fonksiyonunun, başlangıç popülasyonunun ve kromozom şemalarının önemi vurgulanmaktadır. Genetik operatörler gösterilmektedir. Algoritmaların çoklu işlem yeteneğine sahip varyantları açıklanmaktadır. Gerçek zamanlı uygulama kapasitesi tartışılmaktadır.

Bu bölümde ayrıca, ortam haritalarının ve yapılandırma alanının oluşturulması, hava aracı kısıtlarının tespit edilmesi ve yolların gösterimi konularına değinilmektedir. Yol planlama yazılımına ilişkin tüm kavramların ayrıntılı olarak açıklanmasına odaklanılmaktadır. 3B ortamda yol planlaması için grafik kullanıcı arayüzleri (GUI) gösterilmektedir. GUI'lerin nasıl kullanılacağına ilişkin yönergeler sağlanmaktadır. Üretilen yolların geçerliliğinin ve kullanılabilirliğinin gösterimi için simülasyon platformu açıklanmaktadır.

3.1. İnsansız Hava Aracı Seçimi

Aerosonde, uzun menzilli hava durumu verilerinin alınması için tasarlanmış küçük otonom bir İHA'dır (Bkz. Görsel 3.1'de solda). Aerosonde İHA, Amerikan şirketi Insitu, Inc. tarafından geliştirilmiş ve Amerikan AAI Corporation'ın stratejik bir girişimi olan Avustralya merkezli Aerosonde Ltd. tarafından üretilmiştir. Güncel motor ve tasarım özellikleri ile 2007'den itibaren Textron şirketi Aerosonde üretimine devam etmektedir (Bkz. Görsel 3.1'de sağda).



Görsel 3.1. Aerosonde "Laima" İHA (solda), Aerosonde MK 4.7 İHA (sağda)

Özellikle okyanus üzerinde ve zorlu koşullarda keşif amaçlı geliştirilmiş Aerosonde, düşük maliyetli tasarımı ve operasyonel esnekliği ile çok çeşitli uzaktan algılama uygulamaları için kullanılmıştır. Aerosonde geliştirme konsepti 1992'de McGeer'in araştırma makalesinde önerilmiştir [291]. Ön geliştirme çalışmaları için ana yüklenici kuruluşlar, ABD merkezli InsituGroup ve Avustralya Meteoroloji Bürosu olmuştur [292].

ABD Deniz Araştırmaları Ofisi tarafından yapılan bir sözleşme kapsamında Washington Üniversitesi'ne teslim edilen on İHA'dan birisi Kuzey Atlantik'i geçen ilk İHA olmuştur. Kod adı "Laima" olan İHA, Newfoundland'daki Bell Adası'ndan havalanmış ve 26 saat 45 dakikalık otonom bir uçuşun ardından Dış Hebridler'deki Benbecula Adası'na güvenli bir şekilde inmiştir [292]. Aerosonde İHA Özellikleri Tablo 3.1'de gösterilmiştir [292].

Tablo 3.1. *Aerosonde İHA özellikleri*

Özellik	Değer
Kanat açıklığı	2,9 m
Brüt ağırlık	13-14 kg
Yük	1 kg
Seyir hızı	20-30 m/s
Uzunluk	1,7 m
Kanat açıklığı	2,9 m
Yükseklik	0,6 m
Azami hız	30-40 m/s
Azami irtifa	4500 m
Menzil	2500-3000 km
Dayanıklılık	30-35 saat
Yunuslama limiti	-15° 30°
Yatış limiti	-30° 30°
Rpm limiti	400 rpm 1000 rpm
Gaz kolu pozisyonu limiti	0,1 (rolanti) 1,3 (askerî)
Motor tipi	Modifiye Enya R120, 1,74 hp (1280 W)
Fırlatma	Araç tavan rafı
İniş	Göbek üzerinde
Operasyon	Özerk
Navigasyon	Küresel konumlama sistemi
İletişim	UHF Radyo, Uydu (geliştirme aşamasında)
Hava durumu verileri	Rakım, rüzgâr, basınç, sıcaklık, nem

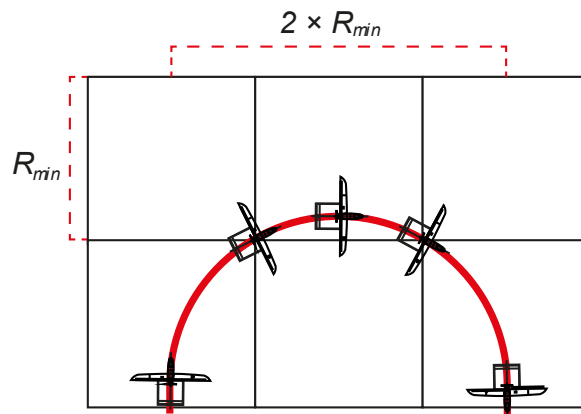
Modifiye Enya R120 model motora sahip Aerosonde; küçük bir bilgisayar, meteorolojik ölçüm araçları ve navigasyon için bir GPS alıcısına sahiptir. Sonraki geliştirmeler ile keşif ve gözlem fonksiyonu kazanmıştır. Gelişmiş bir komuta ve kontrol sistemi ile bütünleşik Aerosonde'nin en büyük esnekliği, hemen hemen her yerden konuşlandırılabilmesi ve kontrol edilebilmesidir. Literatürde çok sayıda çalışmada incelenen bir hava aracı olarak öne çıkmaktadır [293-297].

3.2. Çevresel Modelleme

Yol planlamasından önce bir çevresel model oluşturulması ve çevresel bilgilerin kullanıma uygun hale getirilmesi hesaplamaların büyük ölçüde azaltılmasına yardımcı olmaktadır.

Düzenli ızgaralar oluşturulurken çevre kapsamlı değerlendirilir. Bu da daha sonra yapılacak işlemlerde kolaylık sağlamaktadır. Yol planlamada işlem sonrası teknikler, düzenli ızgaralarda, diğer hücre ayrıştırma tekniklerine kıyasla daha verimli bir şekilde uygulanabilmektedir.

Izgara grafiklerinde engelin bir kısmını içeren her hücre kullanılamaz olmaktadır. Bir engelin yalnızca küçük bir bölümünü içeren hücrelerin kullanılamaz olarak değerlendirilmesi bir sayısallaştırma hatasıdır. Sayısallaştırma hatasını azaltmak için ızgara hücrelerinin boyutu azaltılabilir ancak ihtiyaç duyulan bellek miktarı ve ızgara grafiğinin aranmasında işlem yükü artacaktır. Uygulamalarda hücre boyutu ile hesaplama ihtiyacı arasında bir denge gözetilmektedir.



Şekil 3.1. İHA'nın ızgara hücreleri üzerinde asgari yarıçapta dönüşü

Izgara grafiğinde iki nokta arasında bir yol varsa, gerçekte de bir yol olduğundan emin olmak için kritik sayısallaştırma hataları önlenmelidir. Örneğin, birbirine komşu

dört hücreden çapraz konumda olan ikisi arasından geçecek aracın, genişliği ve belirli bir ayırma mesafesi kadar alanın engel içermediğinin teyit edilmesi açısından geçiş koordinatlarındaki engel durumu komşu hücrelere tanımlanmalıdır.

Bu çalışmada kullanılan ızgara tabanlı yaklaşımda ortam hücrelere ayrılmıştır. Hücre boyutları, İHA'nın güvenli bir şekilde hareket edeceği büyüklükte tanımlanmaktadır [80].

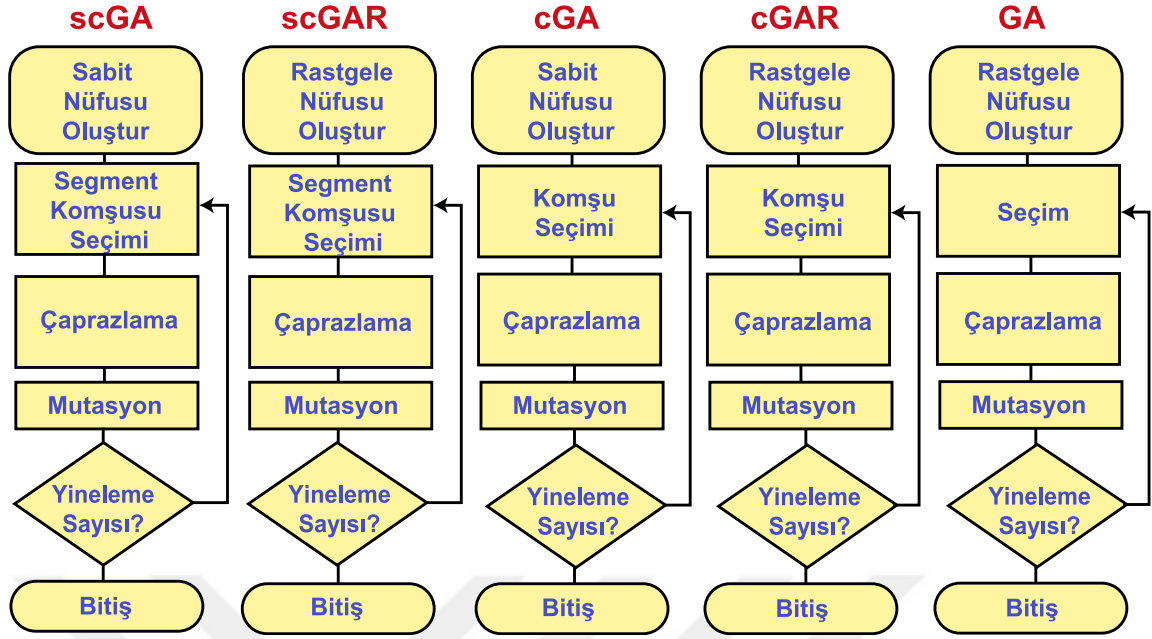
Hesaplama kolaylığı açısından ortam ayrıştırması aşamasında ızgara hücrelerinin genişliği (Bölüm 3.5'te açıklanacağı üzere) en az asgari dönüş yarıçapı (R_{min}) olarak belirlenmiştir. Böylece, Şekil 3.1'de gösterildiği üzere, hava aracının ek bir hesaplama gerektirilmeksizin harita üzerindeki tüm yolları güvenli bir şekilde dönebilmesi mümkün olmuştur. Test senaryolarında bu genişlik 500 m boyutunda tanımlanmıştır.

3.3. Hücresel Genetik Algoritma

Yol planlama problemlerinde popülasyona dayalı teknikler başarıyla uygulanmaktadır. İHA yol planlaması alanında GA'nın çok sayıda örneği bulunmaktadır ancak cGA'nın İHA yol planlama probleminde kullanılmadığı görülmüştür. Bu tez çalışmasında cGA'nın başarılı bir varyantı olan scGA önerilmiştir. Algoritmalar ve alt prosedürleri kategoriler halinde sahte kodlar verilerek detaylandırılmıştır.

3.3.1. Algoritma

cGA yapılandırılmış popülasyon yapısının başarılı bir örneğidir. Popülasyon içindeki her bireyin, optimizasyon problemi çözümüne uygunluk derecesi onun hayatta kalma ve üreme şansını belirlemektedir. cGA'lar ve genel olarak GA'ların temelini çeşitlilik ve gelişimi sağlayan varyasyon (çaprazlama ve mutasyon) ve kaliteyi sağlayan seçim süreçleri oluşturmaktadır. cGA'nın GA'dan en önemli farklılığı olan merkezi olmayan popülasyon yapısı, daha yüksek genetik farklılaşmayı mümkün kılar ve standart bir GA'ya eşdeğer hesaplama maliyeti ile elde edilen çözümlerin kalitesini iyileştirmektedir. Diğer bir ifade ile aynı kalitede çözüme daha hızlı ulaşılabilir. Bu bölümde, tez çalışmasında önerilen algoritmaların yanı sıra karşılaştırma yapabilmek için geleneksel bir GA yaklaşımı da açıklanmaktadır. Algoritmaların basitleştirilmiş akış şemaları Şekil 3.2'de gösterilmiştir.



Şekil 3.2. $scGA(R)$, $cGA(R)$ ve GA algoritmalarının basitleştirilmiş akış şemaları

GA ve cGA 'da her bir kromozom, bir aday yolu temsil etmektedir. Popülasyon her nesilde genetik işlemlerle (çaprazlama, mutasyon, seçim, değiştirme) tekrar tekrar şekillenir. Kromozomlar uygunluk değerlerine göre sonraki nesle aktarılır ya da aktarılmaz. Genel hedef, amaç fonksiyonunun değerini olabildiğince azaltmak ve ideal kromozomu bulmaktır. Bu kromozom, optimale yakın çözüm olarak adlandırılır.

Bir cGA , mekânsal olarak belirlenmiş sabit bir yapıya sahiptir. Her düğüm, popülasyondaki bireylerden birini içermektedir. Her birey, yalnızca bir şablon ile belirlenmiş bir dizi komşu ile iletişime sahiptir.

Uygulanan cGA algoritması Tablo 3.2'de gösterilmiştir. Genelde cGA 'larda öncelikle kromozomlar oluşturulur sonrasında kromozomların uygunluk değerleri belirlenir. Ardından her birey için her yinelemede aşağıdaki işlemler gerçekleştirilir:

- 1) Birey ve komşularından birisi seçilir.
- 2) Çaprazlama yoluyla bir yavru oluşturulur ve uygunluğu hesaplanır.
- 3) Yavru belirli bir oranda mutasyona uğratılır ve uygunluğu hesaplanır.
- 4) Ebeveynin yavruyla değiştirilip değiştirilmeyeceğine karar verilir. Yavrunun uygunluk değeri ebeveynin uygunluk değerinden daha iyiye değişim yapılır (Farklı olarak $scGA(R)$ 'de uygulanan OPRO yöntemi aşağıda açıklanacaktır).

Tablo 3.2. *cGA(R)* algoritması

S.N.	İşlev
1	<i>mahalleYapısınıOluştur(p)</i>
2	$p \leftarrow \text{başlangıçPopülasyonuOluştur() // cGA 41 sabit 14 rastgele / cGAR: 55 rastgele}$
3	$uygunluk \leftarrow \text{uygunlukDeğeriHesapla}(p)$
4	Döngü başlat , durma koşulu sağlanana kadar:
5	Döngü başlat , popülasyondaki her birey (<i>i</i>) için:
6	$mahalle \leftarrow \text{komşularıHesapla}(p,i)$
7	$ebeveynler \leftarrow \text{seçim}(p,mahalle, uygunluk)$
8	$yavru \leftarrow \text{çaprazlama}(ebeveynler,mahalle)$
9	$yoluDüzeltilir(yavru)$
10	$uygunluk \leftarrow \text{uygunlukDeğeriHesapla}(yavru)$
11	$p,i \leftarrow \text{dahaİyiyeDeğiştir}(yavru, uygunluk)$
12	$yavru \leftarrow \text{mutasyon}(yavru)$
13	$yoluDüzeltilir(yavru)$
14	$uygunluk \leftarrow \text{uygunlukDeğeriHesapla}(yavru, uygunluk)$
15	$p,i \leftarrow \text{dahaİyiyeDeğiştir}(yavru, uygunluk)$
16	Döngü sonu
17	Döngü sonu
18	bitir $\leftarrow \text{enİyiBirey}(p)$

Tablo 3.3. *scGA(R)* algoritması

S.N.	İşlev
1	<i>ızgaraYapılandırmasınıTanımla()</i>
2	<i>mahalleYapısınıOluştur(p)</i>
3	$p \leftarrow \text{başlangıçPopülasyonuOluştur() // scGA 41 sabit 14 rastgele / scGAR: 55 rastgele}$
4	$uygunluk, parcaUygunluk \leftarrow \text{uygunlukDeğeriHesapla}(p)$
5	Döngü başlat , durma koşulu sağlanana kadar:
6	Döngü başlat , popülasyondaki her birey (<i>i</i>) için:
7	$mahalle \leftarrow \text{komşularıHesapla}(p,i)$
8	$ebeveynler \leftarrow \text{seçim}(p,mahalle, uygunluk)$
9	$yavru \leftarrow \text{çaprazlama}(ebeveynler, mahalle, uygunluk, parcaUygunluk)$
10	$yoluDüzeltilir(yavru)$
11	$uygunluk, parcaUygunluk \leftarrow \text{uygunlukDeğeriHesapla}(yavru)$
12	$p,i \leftarrow \text{dahaİyiyeDeğiştir}(yavru, uygunluk, parcaUygunluk)$
13	$yavru \leftarrow \text{mutasyon}(yavru, parcaUygunluk)$
14	$yoluDüzeltilir(yavru)$
15	$uygunluk, parcaUygunluk \leftarrow \text{uygunlukDeğeriHesapla}(yavru)$
16	$p,i \leftarrow \text{dahaİyiyeDeğiştir}(yavru, uygunluk, parcaUygunluk)$
17	Döngü sonu
18	Döngü sonu
19	bitir $\leftarrow \text{enİyiBirey}(p)$

Tablo 3.4. Geleneksel GA algoritması

S.N.	İşlev
1	$p \leftarrow \text{başlangıçPopülasyonuOluştur()} // 55 \text{ rastgele}$
2	$uygunluk \leftarrow \text{uygunlukDeğeriHesapla}(p)$
3	Döngü başlat , durma koşulu sağlanana kadar:
4	Döngü başlat , popülasyondaki her birey (i) için:
5	$ebeveynler \leftarrow \text{seçim}(p, uygunluk)$
6	$yavru \leftarrow \text{çaprazlama}(ebeveynler)$
7	$yoluDüzelt(yavru)$
8	$uygunluk \leftarrow \text{uygunlukDeğeriHesapla}(yavru)$
9	$p, i \leftarrow \text{dahaİyiyeDeğiştir}(yavru, uygunluk)$
10	$yavru \leftarrow \text{mutasyon}(yavru)$
11	$yoluDüzelt(yavru)$
12	$uygunluk \leftarrow \text{uygunlukDeğeriHesapla}(yavru, uygunluk)$
13	$p, i \leftarrow \text{dahaİyiyeDeğiştir}(yavru, uygunluk)$
14	Döngü sonu
15	Döngü sonu
16	bitir $\leftarrow \text{enİyiBirey}(p)$

cGA'da, GA'da olduğu gibi genetik operatörler uygulanır ve bu işlemler mahalle yapısı kapsamında her yinelemede tekrarlanır. cGA'da hesaplama karmaşıklığının, genelde hücre sayısı ile doğru orantılı olarak değiştiği gösterilmiş olmakla beraber [246] her yinelemede kullanılan seçim ve varyasyon yöntemlerine göre farklılık göstermektedir [248].

Mahalle içinde uygulanan genetik operasyonlar sırayla (eşzamansız) ya da paralel (eşzamanlı) gerçekleştirilebilir [298]. Tablo 3.2'de gösterilen cGA ve Tablo 3.3'de gösterilen scGA'da popülasyon, yeni popülasyona ait bireyler oluşturulduktan hemen sonra güncellendiği için eşzamansız olarak sınıflandırılır. Böylece, yeni bireyler, ebeveynlerinin nesline ait olanlarla etkileşime girebilmektedir.

Tez çalışması kapsamında geliştirilen yazılımdaki GA (Bkz. Tablo 3.4), cGA(R) (Bkz. Tablo 3.2), scGA(R) (Bkz. Tablo 3.3) algoritmalarının çalışma şekli aşağıda özetlenmiştir.

Öncelikle cGA(R) algoritmasında seçilen hava aracına ait parametreler ve yakıt bilgileri okunur (yol uzunluğu optimizasyonunu esas alan algoritmalarda yakıt verileri değil, Bölüm 3.3.5'te açıklanan, yol uzunluğu ve yükselme durumunda uygulanacak bir ceza puanı kullanılır). Seçilen başlangıç ve bitiş noktaları alınır. Çok süreçli

algoritmelerde hava aracı parametre, yakıt bilgileri ve harita bilgileri veritabanına yazılır ve süreç uygulamaları çalıştırılır. Hesaplama süresi başlatılır. GA, scGAR ve cGAR algoritmalarında 55 rastgele başlangıç popülasyonu oluşturulur. cGA ve scGA için 41 sabit ve 14 rastgele başlangıç popülasyonu oluşturulur. cGA ve scGA’da, sabit yollar veritabanından okunur ve başlangıç/varış noktalarına göre ayarlanır.

14 rastgele popülasyon, her kromozomun en veya boy değerinin sırasıyla; 1/5, 2/5, 3/5 ve 4/5'üne denk gelen dikey ya da yatay eksen alellerine göre sıralanır. Dikey veya yatay koordinatlarından hangisine göre sıralama yapılacağı belirlenirken başlangıç ve bitiş noktalarının konumlarına göre daha dar olan düzlem seçilir. Rastgele oluşturulan yolları temsil eden kromozomlar, sabit kromozomlar dizisinin her n elemanının arasına sırayla ($n \bmod 4 = 2$) şartına göre yerleştirilir.

scGA(R) için ızgara yapılandırmasındaki satır sayısı, ortadaki (sabit başlangıç popülasyonunda en düz yola sahip) kromozoma göre; düğüm sayısı sırasıyla 4’ten az ise 2, 12’den az ise 3, 16’dan az ise 4 olarak tanımlanır. 16’dan büyük düğüm sayısında, düğüm sayısının 0.2 çarpımından elde edilen tam sayı kullanılır. Tüm popülasyonun uygunluk değeri hesaplanır. cGA(R) ve scGA(R)’de mahalle yapısı oluşturulur.

Çok süreçli cGA (Bkz. Tablo 3.5), scGA (Bkz. Tablo 3.6) ve GA (Bkz. Tablo 3.7) algoritmalarında, süreç görevlerinin başlama bilgisi veritabanına yazılır. Başlangıç popülasyonu, süreç sayısı kadar parçalara ayrılarak veritabanına yazılır. Ancak cGA(R) ve scGA(R)’de, süreçlere ayrılan her popülasyonun bir önceki ve bir sonraki bölümlerindeki kromozomlardan mahalle yapısı gereği çaprazlamaya katılacak olanlar çoğaltılarak ilgili popülasyona dâhil edilmektedir. Süreç görevleri tamamlanana kadar beklenir. Görevler tamamlanınca oluşturulan son popülasyon, veritabanından alınır ve güncel popülasyon oluşturulur.

Süreç uygulamaları ise çalıştırıldıktan sonra, parametreleri veritabanından okur ve görev tanımlanana kadar bekler. Görev tanımlanma durumu veritabanından kontrol edilir. Görev başladığında hava aracına ait parametreler, yakıt bilgileri, harita bilgileri ve başlangıç popülasyonu veritabanından okunur. Yineleme sayısı kadar çaprazlama ve mutasyon operatörleri çalıştırılır. Elde edilen popülasyon her yinelemede veritabanına yazılır. cGA(R) ve scGA(R)’de mahalle yapısı gereği, fazladan alınmış olan bireylerin, diğer süreçler tarafından geliştirilen kopyalarında bir iyileşme varsa güncellemek için bir

önceki ve bir sonraki süreçlerin popülasyonu da veritabanından okunur. İyileşen bireyler mevcut popülasyonda güncellenir.

Tek süreçli algoritmalarda yineleme sayısı kadar çaprazlama ve mutasyon operatörleri çalıştırılır. Eğer başlangıç ve varış noktası arasında durak noktaları belirlenmişse her durak noktası başlangıç ve varış noktaları olarak tanımlanarak yukarıdaki işlemler tekrar edilir. Üretilen yollar birleştirilerek bütüncül sürekli bir yol elde edilir. Böylece, hesaplama süresi tamamlanmış olur ve istatistiki bilgilerin hesaplanmasına geçilir. İstatistikler hesaplanır, kaydedilir ve görüntülenir.

Literatürde çeşitliliğin karşılaştırılması için genel bir kabul yoktur [299]. Önerilen yaklaşımda çeşitlilik oranı için özgün bir hesaplama geliştirilmiştir. Tüm popülasyon için ζ_p aşağıdaki gibi (E.3.1) hesaplanmaktadır.

$$\zeta_p = \begin{cases} f_{(\zeta)} & \text{geçerli yol varsa} \\ 0 & \text{geçerli yok yoksa} \end{cases} \quad (3.1)$$

t , yol sayısı; n , toplam nokta sayısı; k , benzersiz nokta sayısı; m , toplam uyum değeri olmak üzere $f_{(\zeta)}$, denklem (E.3.2) ile hesaplanır:

$$f_{(\zeta)} = \frac{t \cdot k \cdot 10^9}{n \cdot m^2} \quad (3.2)$$

$f_{(\zeta)}$ farklı algoritmaların; aynı harita, başlangıç ve varış noktalarında karşılaştırılmasına olanak veren değerler üretir. Her harita ve rota için farklı aralıktaki değerler alacağından sadece aynı konfigürasyonda karşılaştırma açısından elverişlidir. Farklı harita ve noktaların birbirleri ile karşılaştırılması için uygun değildir.

Çoklu işlem algoritmalarında süreçler her yineleme tamamlandığında kendi popülasyonunu veritabanına yüklemekte ve komşu popülasyonları okumaktadır. Ancak yinelemeler aynı anda tamamlanmadığından güncellemeler farklı yineleme zamanlarında erişilebilir olmaktadır. Örnek olarak 1. süreç, 5. yinelemedeyken; 2. Süreç 7. yinelemede olabilir. Bu senaryoda, 1. süreç daha iyi bir yavruyu 5. yinelemede veritabanına yüklediğinde, 2. süreç bunu ancak 8. yinelemede kullanabilecektir. Bu durum, çok süreçli ve tek süreçli algoritmalarda test sonuçlarına da yansıyan bir farklılığa sebep olmaktadır.

Tablo 3.5. Çok süreçli *cGA(R)* algoritması

S.N.	İşlev
1	veritabanı $\leftarrow$ <i>yakıt,parametre</i>
2	süreçleriÇalıştır(<i>süreçSayısı</i>)
3	mahalleYapısınıOluştur(<i>p</i>)
4	<i>p</i> $\leftarrow$ başlangıçPopülasyonuOluştur() // <i>cGA</i> 41 sabit 14 rastgele / <i>cGAR</i> : 55 rastgele
5	<i>uygunluk</i> $\leftarrow$ uygunlukDeğeriHesapla(<i>p</i>)
6	veritabanı $\leftarrow$ görevleriBaslat(<i>p,süreçSayısı</i>);
7	Bekle: <i>görevlerTamamlandı</i> $\leftarrow$ <i>gorevDurumu</i> () $\leftarrow$ veritabanı
8	Eğer <i>görevlerTamamlandı</i> =1 ise <i>p</i> $\leftarrow$ veritabanı , değilse Bekle
9	Bitir $\leftarrow$ enİyiBirey(<i>p</i>)
Süreç Alt Uygulaması	
1	<i>parametre</i> $\leftarrow$ veritabanı
2	Bekle: <i>görevVar</i> $\leftarrow$ <i>gorevDurumu</i> () $\leftarrow$ veritabanı
3	Eğer <i>görevVar</i> =0 ise Bekle , değilse:
4	<i>p</i> $\leftarrow$ veritabanı
5	Döngü başlat , durma koşulu sağlanana kadar:
6	Döngü başlat , popülasyondaki her birey (<i>i</i>) için:
7	<i>mahalle</i> $\leftarrow$ komşularıHesapla(<i>p,i</i>)
8	<i>ebeveynler</i> $\leftarrow$ seçim(<i>p,mahalle, uygunluk</i>)
9	<i>yavru</i> $\leftarrow$ çaprazlama (<i>ebeveynler,mahalle</i>)
10	<i>yoluDüzelt</i> (<i>yavru</i>)
11	<i>uygunluk</i> $\leftarrow$ uygunlukDeğeriHesapla(<i>p</i>)
12	<i>p,i</i> $\leftarrow$ dahaİyiyeDeğiştir(<i>yavru, uygunluk</i>)
13	<i>yavru</i> $\leftarrow$ mutasyon (<i>yavru</i>)
14	<i>yoluDüzelt</i> (<i>yavru</i>)
15	<i>uygunluk</i> $\leftarrow$ uygunlukDeğeriHesapla(<i>p</i>)
16	<i>p,i</i> $\leftarrow$ dahaİyiyeDeğiştir(<i>yavru, uygunluk</i>)
17	Döngü sonu
18	veritabanı $\leftarrow$ dahaİyiBireyleriVeritabanınaYükle(<i>p</i>)
19	<i>p</i> $\leftarrow$ dahaİyiBireyleriVeritabanındanAl() $\leftarrow$ veritabanı
20	Döngü sonu
21	veritabanı $\leftarrow$ göreviBitir(<i>p</i>)

Tablo 3.6. Çok süreçli *scGA(R)* algoritması

S.N.	İşlev
1	veritabanı $\leftarrow$ <i>yakıt,parametre</i>
2	<i>süreçleriÇalıştır(süreçSayısı)</i>
3	<i>ızgaraYapılandırmasınıTanımla()</i>
4	<i>mahalleYapısınıOluştur(p)</i>
5	$p \leftarrow$ <i>başlangıçPopülasyonuOluştur()</i> // <i>scGA 41 sabit 14 rastgele / scGAR: 55 rastgele</i>
6	<i>uygunluk, parcaUygunluk</i> $\leftarrow$ <i>uygunlukDeğeriHesapla(p)</i>
7	veritabanı $\leftarrow$ <i>gorevleriBaslat(p,süreçSayısı);</i>
8	Bekle: <i>görevlerTamamlandı</i> $\leftarrow$ <i>gorevDurumu()</i> $\leftarrow$ veritabanı
9	Eğer <i>görevlerTamamlandı=1</i> ise $p \leftarrow$ veritabanı değilse Bekle:
10	Bitir $\leftarrow$ <i>enİyiBirey(p)</i>
Süreç Alt Uygulaması	
1	<i>parametre</i> $\leftarrow$ veritabanı
2	Bekle: <i>görevVar</i> $\leftarrow$ <i>gorevDurumu()</i> $\leftarrow$ veritabanı
3	Eğer <i>görevVar=0</i> ise Bekle değilse:
4	$p \leftarrow$ veritabanı
5	Döngü başlat, <i>durma koşulu sağlanana kadar:</i>
6	Döngü başlat, <i>popülasyondaki her birey (i) için:</i>
7	$mahalle \leftarrow$ <i>komşularıHesapla(p,i)</i>
8	$ebeveynler \leftarrow$ <i>seçim(p,mahalle, uygunluk)</i>
9	$yavru \leftarrow$ <i>çaprazlama (ebeveynler)</i>
10	<i>yoluDüzeltil (yavru)</i>
11	<i>uygunluk, parcaUygunluk</i> $\leftarrow$ <i>uygunlukDeğeriHesapla(yavru)</i>
12	$p,i \leftarrow$ <i>dahaİyiyeDeğiştir(yavru, uygunluk, parcaUygunluk)</i>
13	$yavru \leftarrow$ <i>mutasyon (yavru)</i>
14	<i>yoluDüzeltil (yavru)</i>
15	<i>uygunluk, parcaUygunluk</i> $\leftarrow$ <i>uygunlukDeğeriHesapla(yavru)</i>
16	$p,i \leftarrow$ <i>dahaİyiyeDeğiştir(yavru, uygunluk, parcaUygunluk)</i>
17	Döngü sonu
18	veritabanı $\leftarrow$ <i>dahaİyiBireyleriVeritabanınaYükle(p)</i>
19	$p \leftarrow$ <i>dahaİyiBireyleriVeritabanındanAl()</i> $\leftarrow$ veritabanı
20	Döngü sonu
21	veritabanı $\leftarrow$ <i>göreviBitir(p)</i>

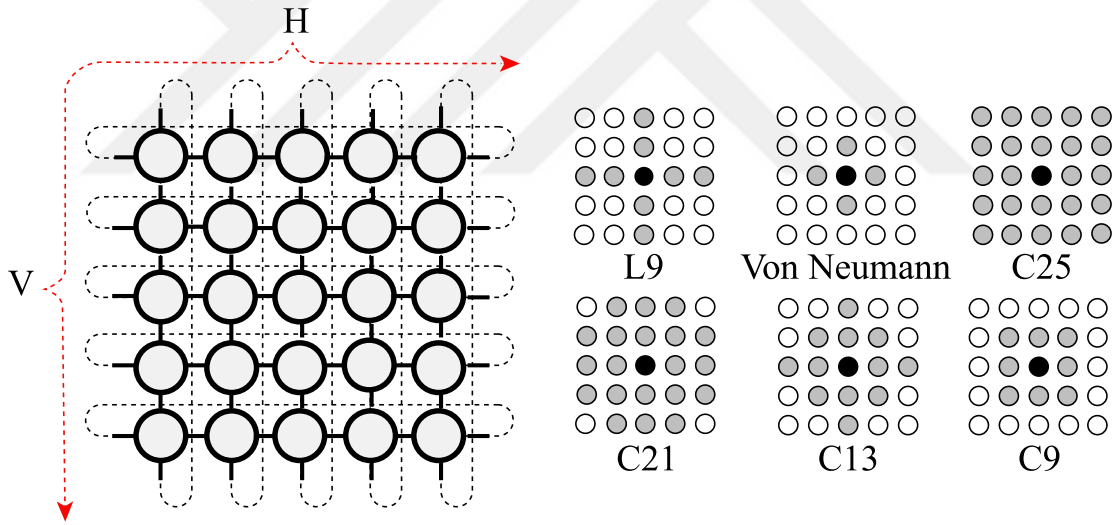
Tablo 3.7. Çok süreçli geleneksel GA algoritması

S.N.	İşlev
1	veritabanı $\leftarrow$ yakıt,parametre
2	süreçleriÇalıştır(süreçSayısı)
3	$p \leftarrow$ başlangıçPopülasyonuOluştur() // 55 rastgele
4	$uygunluk \leftarrow$ uygunlukDeğeriHesapla(p)
5	veritabanı $\leftarrow$ görevleriBaslat(p ,süreçSayısı);
6	Bekle: görevlerTamamlandı $\leftarrow$ görevDurumu() $\leftarrow$ veritabanı
7	Eğer görevlerTamamlandı=1 ise $p \leftarrow$ veritabanı , değilse Bekle
8	Bitir $\leftarrow$ enİyiBirey(p)
Süreç Alt Uygulaması	
1	$parametre \leftarrow$ veritabanı
2	Bekle: görevVar $\leftarrow$ görevDurumu() $\leftarrow$ veritabanı
3	Eğer görevVar=0 ise Bekle , değilse:
4	$p \leftarrow$ veritabanı
5	Döngü başlat , durma koşulu sağlanana kadar:
6	Döngü başlat , popülasyondaki her birey (i) için:
7	$ebeveynler \leftarrow$ seçim(p , $uygunluk$)
8	$yavru \leftarrow$ çaprazlama ($ebeveynler$)
9	$yoluDüzelt$ ($yavru$)
10	$uygunluk \leftarrow$ uygunlukDeğeriHesapla(p)
11	$p,i \leftarrow$ dahaİyiyeDeğiştir($yavru$, $uygunluk$)
12	$yavru \leftarrow$ mutasyon ($yavru$)
13	$yoluDüzelt$ ($yavru$)
14	$uygunluk \leftarrow$ uygunlukDeğeriHesapla(p)
15	$p,i \leftarrow$ dahaİyiyeDeğiştir($yavru$, $uygunluk$)
16	Döngü sonu
17	veritabanı $\leftarrow$ dahaİyiBireyleriVeritabanınaYükle(p)
18	$p \leftarrow$ dahaİyiBireyleriVeritabanındanAl() $\leftarrow$ veritabanı
19	Döngü sonu
20	veritabanı $\leftarrow$ göreviBitir(p)

3.3.2. Mahalle yapısı

Önerilen cGA ve scGA yaklaşımlarında hücreler arasında mekânsal bir mahalle yapısı tanımlanmaktadır. Hücrelerin yapılandırıldığı n boyutlu uzayda her hücreden en yakın k hücre, komşular olarak kabul edilmektedir.

Klasik bir GA'da popülasyon benzersiz tek panmiktik grupta yapılandırılırken, cGA'da popülasyon, mahalle ilişkisinin tanımlandığı bir ızgara üzerinde düzenlenmektedir. Aynı boyutta, panmiktik ve kare ızgara popülasyonu karşılaştırıldığında popülasyon üzerinde oluşan seçim baskısının kare ızgara popülasyonda daha zayıf olduğu gösterilmiştir [300]. Mahallelerin tanımlanmasının, Cn kompakt formu ve Ln lineer formu (Bkz. Şekil 3.3) olarak tanımlanan iki klasik yolu bulunmaktadır [236]. Burada n her komşuluğun boyutunu gösterir. Bir cGA'nın davranışı, mahalle boyutu ve geometrisine bağlıdır. cGA'da halka ve dizi topolojileri seçim baskısı açısından karşılaştırılmıştır [301].

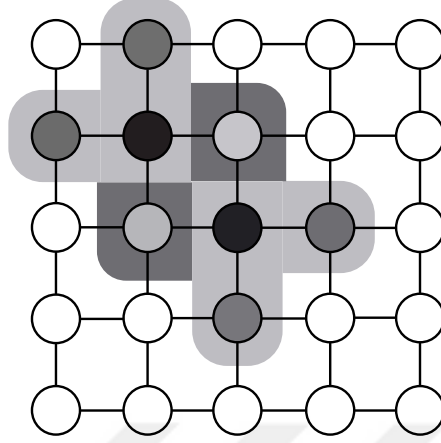


Şekil 3.3. Tipik olarak kullanılan mahalleler

Hücresel otomatlarda olduğu gibi, Von Neumann ve Moore mahalleleri en popüler olanlardır. 2D bir kare kafeste, Von Neumann mahallesi, bir hücre ve onun dikey ve yatay komşularından oluşurken, Moore mahallesi diyagonal komşular içermektedir.

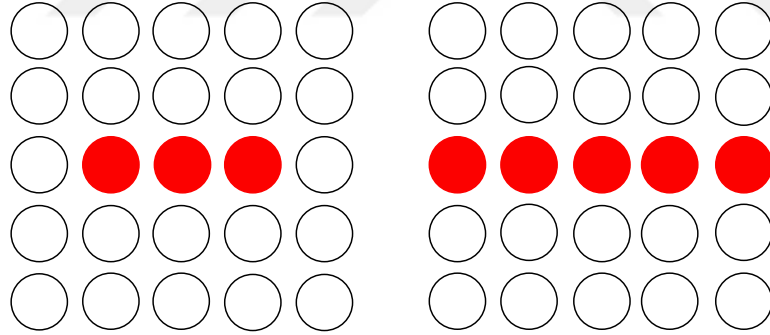
Farklı mahalle modellerinden birinin seçimiyle birlikte, popülasyon içinde etkileşim ve iletişim süreçleri sınırlandırılmış olmaktadır. Bir cGA'da her birey yalnızca kendi yerel komşusu ile genetik işlemlere tabi tutulmaktadır. Bu davranış yapısı, meta-sezgisel aramayı önemli ölçüde farklılaştırır. Bireyin sınırlı komşuluğu, yerel iyi bireylerin tüm popülasyona hızla yayılmasını önlemektedir. Bununla birlikte mahallelerin

örtüşmesi, (Bkz. Şekil 3.4) iyi bireylerin özelliklerinin farklı mahallelere taşınmasına imkân vermektedir.



Şekil 3.4. Üst üste binen mahalleler

Düğümler arası etkileşim, mahalle yapısına göre ve belirli sayıda bireyle gerçekleşir. Literatürde birçok ızgara şekli ve komşuluğu önerilmiş ve incelenmiştir. Şekil 3.5’de tez çalışmasında kullanılacak L3 ve L5 mahalle yapıları örnek olarak gösterilmiştir [255].



Şekil 3.5. Bazı cGA mahalle yapıları: L3 ve L5

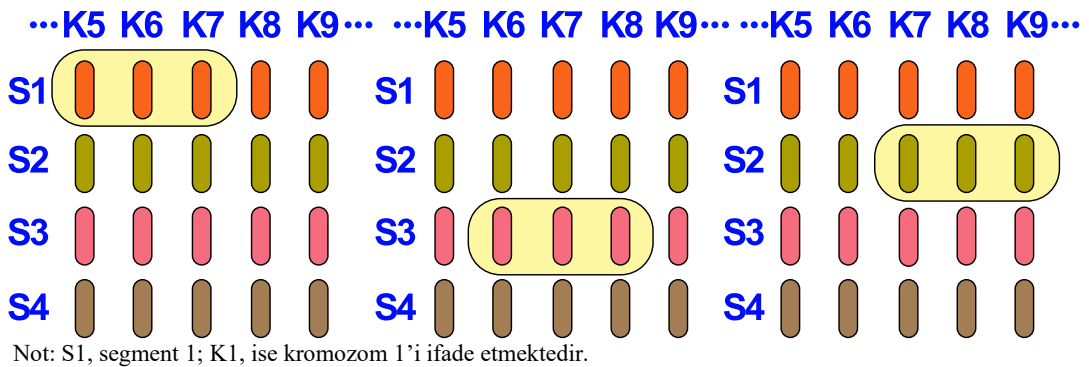
Önerilen tez çalışmasında komşu kromozomlar koordinatlarına göre belirlenmektedir. Başlangıç popülasyonuna göre birbirine en yakın yolları temsil eden kromozomlar, mahalleler içerisinde de yan yana bulunmaktadır. Böylece, komşulukların, kromozomların başlangıçtaki ortak özelliklerine göre belirlendiği bir yaklaşım tanımlanmış olmaktadır. Bu tanımlama yol planlama çalışmaları özelinde algoritma açısından büyük bir kazanım olarak değerlendirilmektedir. İlerde açıklanacağı ve sonrasında test sonuçları üzerinde tartışılacağı üzere çaprazlamanın birbiri ile ortak özelliği olmayan kromozomlar yerine aynı bölgede çözümü temsil eden kromozomlar arasında yapılması algoritmanın başarısını artırmaktadır.

3.3.3. Kromozom şeması

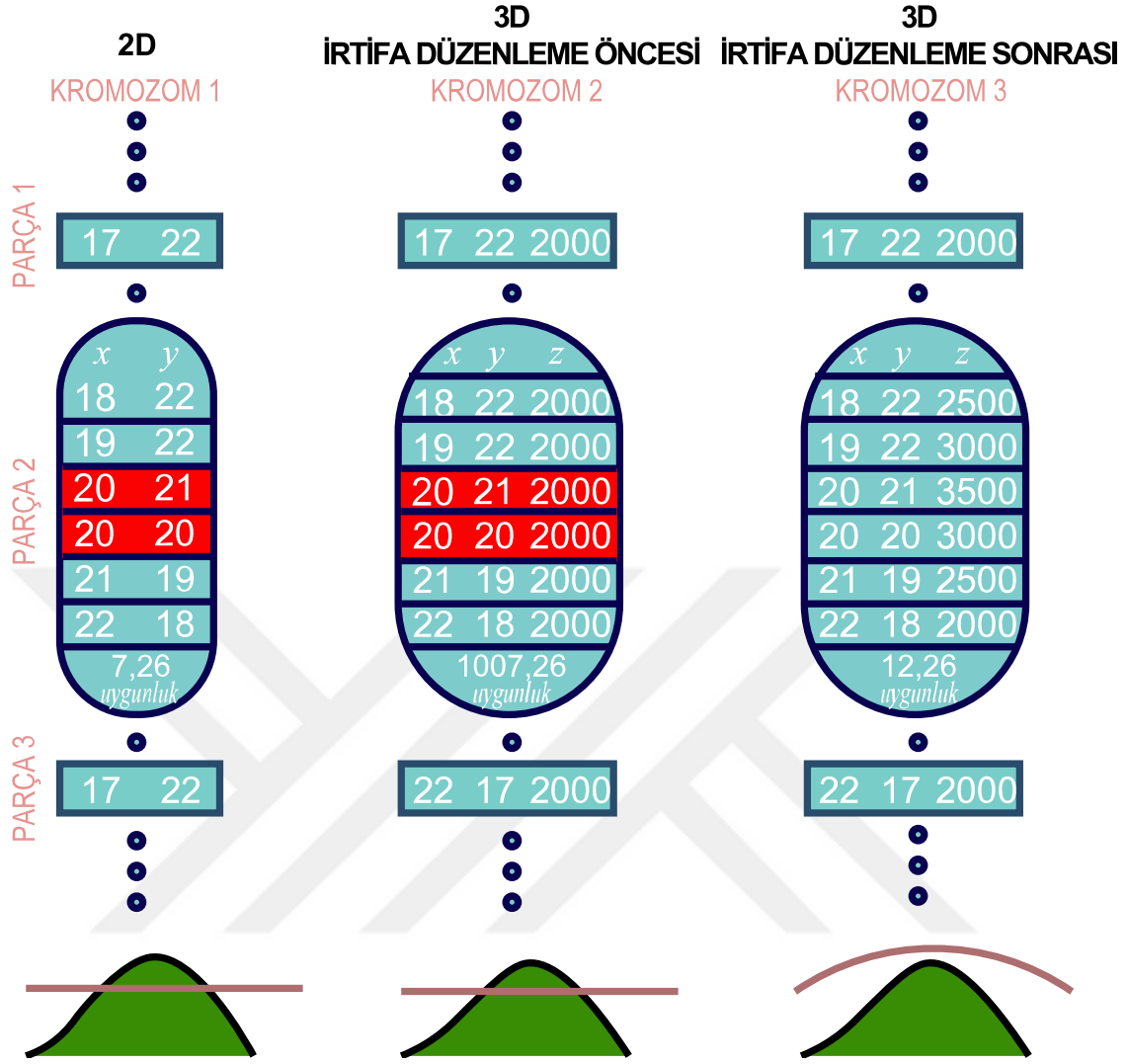
Kromozom kodlama şemasının çaprazlama ve mutasyon gibi genetik operatörlerin tasarımı üzerinde etkisi büyüktür. Kromozomdaki karakterler, semboller, kayan noktalı sayılar ve ikili rakamlar genler olarak isimlendirilir. Bunlar arasında ikili ve gerçek parametre gösterimleri, genetik kümeleme algoritmaları için yaygın şekilde kullanılmaktadır [302]. Gerçek değerli gösterimler, hayata uygunlukları nedeniyle daha pratiktir [303]. Bu tez çalışması kapsamında algoritmalarda kromozomlar gerçek değerler kullanacak şekilde tasarlanmıştır.

Kromozomlar genellikle bir bütün olarak kabul edilmektedir. Önerilen yöntemde parçalı kromozom yapısı tercih edilmiştir. Farklı GA türlerinde parçalı kromozom yapısının örnekleri bulunmaktadır [304]. Kromozom parçalarına uygunluk değeri verilebilir [305]. scGA(R) algoritmasında kromozomlar sabit sayıda alt kromozoma ayrılmıştır ve her alt kromozom bir uygunluk değerine sahiptir. Böylece, yolun tamamı yerine, yolun engeller ile çarpışan geçersiz parçaları algoritma tarafından daha az tercih edilmektedir. Yolun sadece bir kısmında engel varsa parça uygunluk değeri kullanımı, yolun diğer parçalarının göz ardı edilmesini önlemektedir. Böylece, sorunlu parça, bir çaprazlama işleminde sorunsuz bir parçayla değiştirilerek yola şans verilebilir. Sorunlu parçanın onarılmasında mutasyon operatörü de işlevseldir. Şekil 3.7’de scGA’nın kromozom yapısı ve 3D yollar için yüksekliği ayarlama örneği gösterilmiştir [306].

Sorunun doğasına bağlı olarak çeşitli mahalle yapıları daha iyi sonuçlar vermektedir [250]. scGA ve cGA arasındaki en önemli fark, mahallenin bireylerde kullanıldığı gibi kromozom parçalarında da kullanılmasıdır. Böylece, aynı mahalledeki kromozom parçaları referans alınarak çaprazlama yapılabilir. Şekil 3.6’de L3 mahalle yapısının parçalı kromozomlarda kullanımına bir örnek gösterilmiştir [306].



Şekil 3.6. L3 mahalle yapısının parçalı kromozomlarda kullanımı

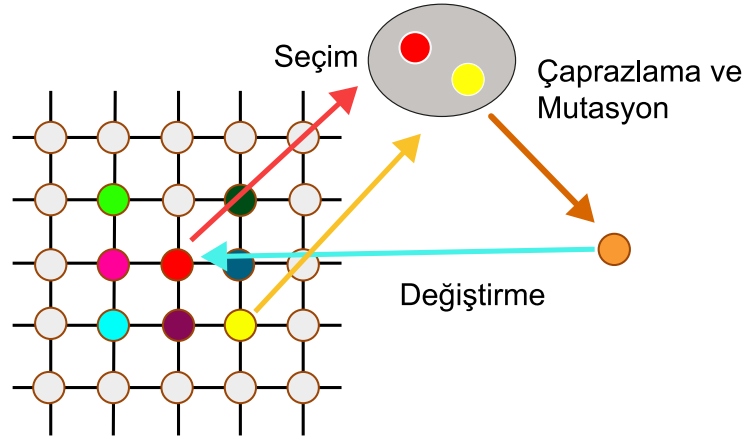


Şekil 3.7. scGA'nın kromozom yapısı ve 3D yollar için irtifayı ayarlama

3.3.4. Genetik operatörler

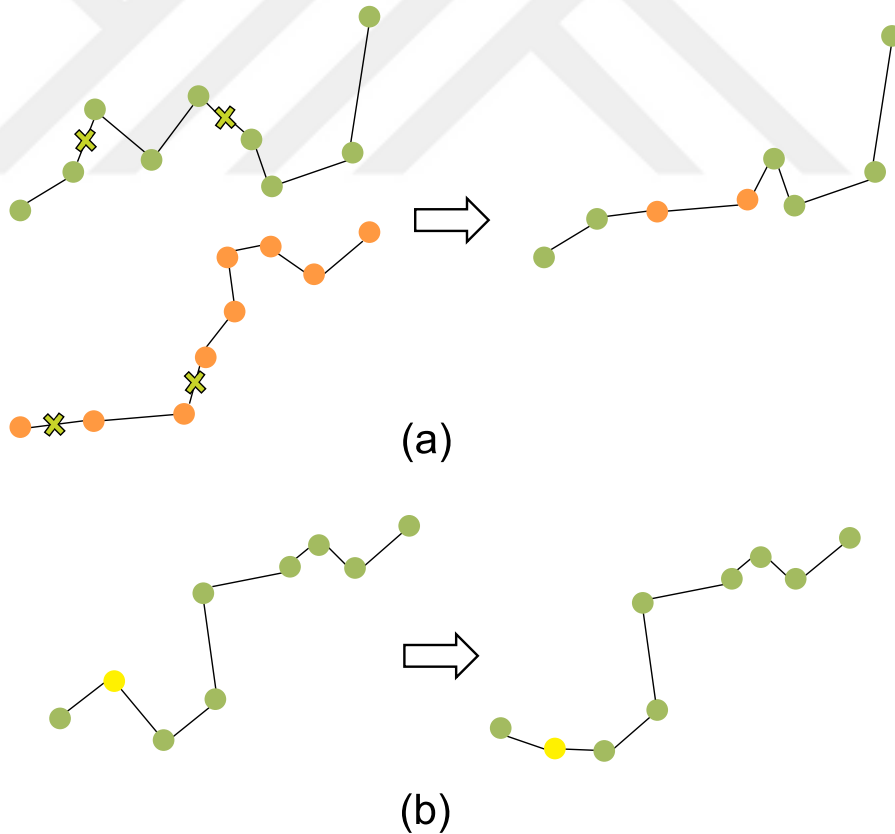
Bölüm 2.2.3.2.'de gösterildiği gibi geleneksel genetik operatörler, seçim, çaprazlama ve mutasyon olarak ifade edilmektedir. Bu bölümde, önerilen algorithmada kullanılan operatörler özetlenmektedir.

GA, her bireyin bir amaç fonksiyonu vasıtasıyla değerlendirildiği bir başlangıç popülasyonu ile başlamaktadır. Her nesilde bireyler uygunluk değerlerine göre çoğaltılmakta veya elenmektedir. Bireyler, GA operatörleri uygulanarak şekillendirilir. GA'da genel olarak kromozomların gelişimini sağlayan iki operatör bulunmaktadır: Çaprazlama ve mutasyon operatörü. Çaprazlama, popülasyon genelinde bir düzenleme yaparken; mutasyon, bireysel bir işlemdir. Mutasyonun sık uygulanması tamamen rastgele bir aramaya yol açacağından genellikle, düşük bir mutasyon oranı belirlenir. Şekil 3.8'de genetik operatörler gösterilmiştir.



Şekil 3.8. Genetik operatörler

Her genetik operatör, bireyin gen yapısında bir değişim sağlamaktadır. Önerilen algoritmalarda, çaprazlama ve mutasyon olmak üzere iki genetik operatör tanımlanmıştır. Buna ek olarak genleri değil bireyleri düzenleyen, seçim ve değiştirme operatörleri kullanılmaktadır.



Şekil 3.9. (a) çift nokta çaprazlama ve (b) tek nokta mutasyon

- Seçim: Seçim operatörü, yüksek kaliteli kromozomların bir sonraki nesle taşınabilmeleri için daha yüksek şansa sahip olmalarını sağlayarak popülasyonun

ortalama kalitesini iyileştirmeyi amaçlamaktadır. İki ebeveyn, en yaygın yöntemlerden olan rulet tekerleği (roundrobin) seçim yöntemi kullanılarak uygunluklarına göre seçilmektedir. Bu yöntem nispeten basit ve etkilidir.

- Çaprazlama: Çaprazlama, iki bireyi seçmek ve bu seçilen iki bireyden belirli bölümleri değiş tokuş etmektir. Çaprazlama ile yavrular üretilir ve yavrular, ebeveynlerin temel özelliklerini miras almaktadır. Çaprazlama operatörü, iki ana unsurdan oluşur: çaprazlama konumlarının belirlenmesi ve gen değişiminin nasıl gerçekleştirileceğidir. Bu tez kapsamında geleneksel iki nokta çaprazlama kullanılır. Her kromozomda iki çaprazlama noktası seçilir ve daha sonra seçilen noktalardan bölünen kromozom parçası diğer kromozom ile birleştirilir. Şekil 3.9’te çift nokta çaprazlama prosedürünün bir örneği gösterilmektedir.
- Mutasyon: Her yinelemede belirli oranda aday kromozom, genlerinden birinin değişikliği ile mutasyona uğrar. Kromozom, rastgele ya da bir seçim yöntemi ile seçilir. Mutasyon işleminde, ebeveyn kromozomunda bir gen seçilerek değişim yapılır. Mutasyon, bir kromozomdaki mevcut genlerin yerine yeni genlerin yerleştirilmesidir. Mutasyon iyi veya kötü sonuçlanabilir.
- Değiştirme: Yavruların ebeveynleri ile karşılaştırılarak sonraki nesle hangi bireylerin taşınmasına karar verildiği operatördür. Belirli kurallara veya modellere göre mevcut popülasyondaki bir bireyin sahip olduğu yüksek uygunluğun sonraki nesle aktarılması önemlidir. Genel olarak, daha yüksek uygunluk değerine sahip bireylerin bir sonraki nesle aktarılması sağlanır. Daha iyi olanın sonraki nesle taşınması ile iyi bireylerin korunması amaçlanır. Baskın olmayan iyi çözümleri depolamak için harici bir arşiv stratejisi kullanılan araştırmalar bulunmaktadır [307, 308]. Geri besleme mekanizması temelinde, arşivden seçilen baskın olmayan çözümler, popülasyondaki bazı bireylerin yerini almaktadır. Önerilen yaklaşımda yavrular daha iyi oldukları takdirde yeni nesle aktarılırlar ve bir arşiv mekanizması uygulanmaz.
 - GA’larda genelde yeni oluşan yavru kromozomun uygunluk değeri, ebeveyn kromozomdan daha iyi ise değişim yapılmaktadır. scGA(R)’de diğerlerinden farklı olarak aşırı korumacı bir yaklaşım ile yavru kromozom uygunluk değeri, ceza puanından (yolun uygulanamaz olması durumunda eklenen puandan) küçükse değişim gerçekleştirilir.

- Eğer yol çaprazlamaya veya mutasyona rağmen halen cezalıysa değişim yapılmaz. Ancak parça uygunluk değerlerine bakılır. Cezadan yüksek kromozom parçası sayısı ebeveyn kromozoma kıyasla azalmışsa yavruda bir iyileşme olduğu değerlendirilerek cezalı olsa dahi yavru tercih edilir. Bu aşırı korumacı değiştirme operatörü (OPRO), scGA(R)'yi diğerlerinden ayıran en belirleyici farklılıklardan birisidir.

Literatürde aşırı korumacı değiştirme operatörü kavramı ile karşılaşılmamıştır. Aşırı seçkinci kavramını kullanan çalışmalarda, bu kavram ile kastedilen operatör: en uygun belirli sayıda bireyin popülasyondan rastgele seçilen bir dizi bireyle çaprazlanması [309] veya popülasyonun belirli sayıda iyi bireyinin bir sonraki nesli doğrudan etkilemesi [310] işlemleridir. Önerilen algoritmalarda uygulanan operatör farklı bir konsepttedir.

Durdurma Kriterleri: GA'da yeni popülasyon, yinelemeli olarak eski popülasyon ve yavrular arasından seçilerek oluşturulur. Durdurma kriteri yoksa sonsuz devam eden bir döngü bulunmaktadır. Sonlandırma koşuluna ulaşıldığında yineleme bitirilir. Problemin türüne bağlı olarak durdurma kriteri, problemi çözmek için harcanan süreye veya yineleme sayısına bağlıdır [9]. Bazı algoritmalar kalıcı bir çözüme ulaştıklarında veya belirli sayıda nesil için yakınsama ilerlemezse durdurulur. Önerilen yaklaşımda algoritmalar belirli bir yineleme sayısı ile tamamlanır.

Tez çalışması kapsamında geliştirilen yazılımda; cGA(R) (Bkz. Tablo 3.8), scGA(R) (Bkz. Tablo 3.9) ve GA (Bkz. Tablo 3.10) algoritmalarında seçim ve çaprazlama operatörlerinin çalışma şekli aşağıda özetlenmiştir:

Seçim operatöründe GA'da çaprazlanacak her iki kromozom da rulet tekerleği seçim metoduyla alınır. cGA(R) ve scGA(R)'de her birey için 1. kromozom, satır tarama yöntemiyle sırayla alınır. cGA(R)'de 2. kromozom, rulet tekerleği seçim metoduyla elde edilirken scGA(R)'de seçim farklıdır. scGA(R)'de öncelikle, 1. kromozomun parçalarının ters uygunluğuna göre rulet tekerleği seçim metoduyla bir kromozom parçası seçilir. Bu kromozom parçasının mahallesindeki kromozom komşuluğunda, rulet tekerleği seçim metoduyla 2. kromozom belirlenir. Yani seçilen parçanın iyileştirilebilmesi amacıyla seçilen 2. kromozom, genellikle iyi parçanın bulunduğu kromozom olmaktadır.

Seçilmiş olan kromozomlarda çaprazlama noktaları aşağıdaki şekillerde belirlenir:

- GA, cGA ve cGA(R)'de 1 ve 2. kromozomun başlangıç ve bitiş genleri, kromozomların ilk ve son geni arasından rastgele seçilir.
- scGA(R)'de 1. kromozomun başlangıç ve bitiş genleri, seçili kromozom parçasının dışında, bir önceki kromozom parçasının ilk geni ile bir sonraki kromozom parçasının son geni arasından rastgele seçilir.
 - 2. kromozomun seçiminde, yarı yarıya bir ihtimalle gerçekleşen iki yöntem izlenir:
 1. Yöntem: 2. kromozomun başlangıç ve bitiş genlerinin her birisi 1. kromozomdaki eşlerinin en fazla %70 öncesinde veya sonrasında olacak şekilde rastgele seçilir.
 2. Yöntem: Diğer algoritmalarda olduğu gibi kromozomun ilk geni ve son geni arasından rastgele seçilir.

1. kromozomda seçilen genlerin arası silinir ve 2. kromozomda seçilen genlerin arası kromozomdan kopyalanarak bu aralığa yerleştirilir. Birleşim noktalarında yol sürekliliği sağlanır. Yolda, bu işlem sonrası kesişim oluşursa yol düzeltilir. Böylece, hava aracı kinematik kısıtlarına uyumsuz bir yola izin verilmez.

Tablo 3.8. cGA algoritması seçim ve çaprazlama operatörü

S.N.	İşlev
1	$i, p, mahelle, uygunluk \leftarrow fonsiyonGirişBilgisi$
2	$ilkKromozom \leftarrow i$
3	$ikinciKromozom \leftarrow ruletSeçimi(p, mahelle, uygunluk)$
4	$ilkKromozomNoktaları[1;2] \leftarrow rastgeleİkiNoktaSeç(ilkKromozom)$
5	$ikinciKromozomNoktaları[1;2] \leftarrow rastgeleİkiNoktaSeç(ikinciKromozom)$
6	$parçayıSil(ilkKromozom, ilkKromozomNoktaları[1;2])$
7	$kesit \leftarrow parçayıAl(ikinciKromozom, ikinciKromozomNoktaları [1;2])$
8	$yavru \leftarrow parçayıKromozomaBirleştir(ilkKromozom, kesit)$
9	Bitir $\leftarrow yavru$

Tablo 3.9. *scGA algoritması seçim ve çaprazlama operatörü*

S.N.	İşlev
1	$i, p, mahelle, uygunluk \leftarrow fonsiyonGirişBilgisi$
2	$ilkK \leftarrow i$
3	$ikinciK \leftarrow ruletSeçimi(p, mahelle, uygunluk)$
4	$kParcas \leftarrow tersRuletSeçimi(ilkK, parçaUygunluğu)$
5	$ilkKNok[1;2] \leftarrow rastgeleİkiNoktaSeç(ilkK, kParcas-1, kParcas+1)$
6	$x \leftarrow rastgeleSayı(0;1)$
7	Eğer $x=1$ ise
8	$t \leftarrow rastgeleSayı(0,3;0,7)$
9	$ilkKromozomNoktaları[1;2] \leftarrow rastgeleİkiNoktaSeç(ikinciK, ilkKNok[1;2]*t)$
10	Değilse
11	$ikinciKromozomNoktaları[1;2] \leftarrow rastgeleİkiNoktaSeç(ikinciK)$
12	$parçayıSil(ilkK, ilkKromozomNoktaları[1;2])$
13	$kesit \leftarrow parçayıAl(ikinciK, ikinciKromozomNoktaları [1;2])$
14	$yavru \leftarrow parçayıKromozomaBirleştir(ilkK, kesit)$
15	Bitir $\leftarrow yavru$

Tablo 3.10. *Geleneksel GA algoritması seçim ve çaprazlama operatörü*

S.N.	İşlev
1	$p, uygunluk \leftarrow fonsiyonGirişBilgisi$
2	$ilkKromozom \leftarrow ruletSeçimi(p, uygunluk)$
3	$ikinciKromozom \leftarrow ruletSeçimi(p, uygunluk)$
4	$ilkKromozomNoktaları[1;2] \leftarrow rastgeleİkiNoktaSeç(ilkKromozom)$
5	$ikinciKromozomNoktaları[1;2] \leftarrow rastgeleİkiNoktaSeç(ikinciKromozom)$
6	$parçayıSil(ilkKromozom, ilkKromozomNoktaları[1;2])$
7	$kesit \leftarrow parçayıAl(ikinciKromozom, ikinciKromozomNoktaları [1;2])$
8	$yavru \leftarrow parçayıKromozomaBirleştir(ilkKromozom, kesit)$
9	Bitir $\leftarrow yavru$

Yeni oluşan yavru kromozomun uygunluk değeri, ilk kromozomdan daha iyi ise değiştirilir. scGA(R)'de OPRO ile eğer yol çaprazlamaya rağmen halen cezalıysa değişim yapılmaz. Bunun yanında kromozom uygunluk değeri, ceza puanından küçükse veya cezalı olsa dahi parça uygunluk değerlerinde cezadan yüksek kromozom parçası sayısı ebeveyn kromozoma kıyasla azalmışsa yavru tercih edilir ve değişim yapılır.

Son olarak, çok süreçli algoritmalarda, yeni kromozomu içeren güncel popülasyon veritabanına yazılır. Mahalle yapısına göre bir önceki ve bir sonraki süreçlerin oluşturdukları güncel popülasyonlar, mahalle yapısına göre veritabanından alınarak komşu kromozomlarda bir iyileşme varsa mevcut popülasyonda güncelleme yapılır.

Tablo 3.11. *Mutasyon operatörü*

S.N.	İşlev
1	$i, p, uygunluk, mutasyonAralığı \leftarrow fonsiyonGirişBilgisi$
2	$konum \leftarrow rastgeleKonumSeç(i)$
3	$eskinokta \leftarrow i[konum]$
4	$parçayıSil(i, konum - rastgeleSayı(1, mutasyonAralığı),$ $konum + rastgeleSayı(1, mutasyonAralığı))$
5	$x \leftarrow rastgeleSayı(eskinokta - rastgeleSayı(1, mutasyonAralığı),$ $eskinokta + rastgeleSayı(1, mutasyonAralığı))$
6	$yeniNokta \leftarrow rastgeleKoordinatBelirle(i+x)$
7	$yavru \leftarrow noktayıKromozomaBirleştir(i, x)$
8	Bitir $\leftarrow yavru$

Tablo 3.12. *scGA mutasyon operatörü*

S.N.	İşlev
1	$i, p, uygunluk, mutasyonAralığı \leftarrow fonsiyonGirişBilgisi$
2	$kromozomParçası \leftarrow tersRuletSeçimi(ilkK, parçaUygunluğu)$
3	$konum \leftarrow rastgeleKonumSeç(i, kromozomParçası)$
4	$eskinokta \leftarrow i[konum]$
5	$parçayıSil(i, konum - rastgeleSayı(1, mutasyonAralığı),$ $konum + rastgeleSayı(1, mutasyonAralığı))$
6	$x \leftarrow rastgeleSayı(eskinokta - rastgeleSayı(1, mutasyonAralığı),$ $eskinokta + rastgeleSayı(1, mutasyonAralığı))$
7	$yeniNokta \leftarrow rastgeleKoordinatBelirle(i+x)$
8	$yavru \leftarrow noktayıKromozomaBirleştir(i, x)$
9	Bitir $\leftarrow yavru$

Tablo 3.11-12’de gösterilen mutasyon operatörleri, her yinelemede bireylere 0.1 ihtimal ile uygulanır. Öncelikle kromozom üzerinde rastgele bir gen seçilir. scGA(R)’de mutasyona uğratılacak nokta ters rulet yöntemiyle seçilen kromozom parçası üzerinde rastgele belirlenir. Bu genin pozisyonundan, 1 ile "mutasyon aralığı" arasında rastgele üretilen bir sayı kadar uzaklığı kromozomdan silinir. Silinen genin dikey ve yatay alellerine (koordinatlarına) ayrı ayrı, 1 ile "mutasyon aralığı" arasında rastgele sayılar eklenerek yeni aleller oluşturulur. Yeni gen, yol sürekliliği sağlanarak silinen bölgede kromozoma bağlanır. Yavru kromozom uygunluk değeri daha iyi ise ebeveyn ile değiştirilir.

scGA(R)’de (Bkz. Tablo 3.12) yukarıda çaprazlama operatöründe de açıklandığı gibi aşırı korumacı bir yaklaşım ile OPRO; yavru kromozom uygunluk değeri ancak ceza puanından küçükse değişimi gerçekleştirir. Eğer yol mutasyona rağmen halen cezalıysa değişim yapılmaz ve ebeveyn kromozom korunur. Ancak parça uygunluk değerlerinde ceza puanından yüksek kromozom parçası sayısı ebeveyn kromozomdaki duruma kıyasla azalmışsa yavru da yeterli bir iyileşme olduğu değerlendirilerek yavru kromozom cezalı durumda olsa dahi tercih edilir.

3.3.5. Amaç fonksiyonu

Genelde yol planlamada optimizasyon, yakıt tüketimi veya kat edilen mesafe ile ilgili olarak amaç fonksiyonunca hesaplanan uygunluk değerinin en aza indirilmesidir. Popülasyondaki bir kromozomun çözüme uygunluğunu belirlemek için kullanılan amaç fonksiyonu, GA’nın en önemli parçasıdır [311]. Amaç, maliyet ve uygunluk vb. isimler alan bu fonksiyon çözümün doğruluk seviyesini belirlemektedir. Amaç fonksiyonu, ürettiği değeri en aza indiren ve dolayısıyla istenen tüm nitelikleri en iyi şekilde karşılayan yolun belirlenebilmesini sağlamaktadır.

Yol planlama algoritmasının en önemli belirleyici unsuru olan amaç fonksiyonu, İHA performans özelliklerine uyumlu, tehlikeli bölgelerden kaçınan, yol için kat edilen mesafeyi ya da harcanan yakıtı en aza indirdiği derecede düşük bir değere sahip olmaktadır. Amaç fonksiyonları farklı kısıtlar ile farklı hava araçlarına uyarlanabilmektedir.

Amaç fonksiyonu, karşılaştırılan kromozomlar arasında daha iyi kromozomun seçilmesinde kullanılan, probleme özgü yol maliyetine ilişkin skaler bir değer

üretmektedir. GA'nın yakınsama hızını doğrudan etkileyen bir unsurdur. En doğru sonuç, en doğru amaç fonksiyonu ile elde edilebilir. Genel olarak yol uzunluğu, amaç fonksiyonlarında belirleyici olurken bazı çalışmalarda yakıt sarfıyatı ön plana çıkmaktadır [312].

Amaç fonksiyonuna göre kullanılması mümkün olmayan yolların maliyeti, mümkün yolların maliyetinden daha yüksek olmalıdır. Bir ceza puanı ekleyerek mümkün olmayan çözümler, mümkün olanlardan kesin bir çizgi ile ayrılır. Ceza puanı kullanımı ile tüm geçersiz yolların maliyeti, geçerli yollardan daha yüksek bir değere sahip olmaktadır. Örnek olarak geçerli yolların maliyeti 0 ile 1000 arasındayken geçersiz yolların maliyeti 1000'den fazladır [9].

En kısa yolu üretmek için amaç fonksiyonu şu şekilde tasarlanmıştır:

d , başlangıç noktasından (x_s, y_s) varış noktasına (x_f, y_f) öklid mesafesidir.

$$d = \sqrt{(x_s - x_f)^2 + (y_s - y_f)^2} \quad (3.2)$$

$$P_d = \begin{cases} 0 & \text{yol engelsizse} \\ c_1 & \text{yol üzerinde engel veya erişimi imkânsız yükselti varsa} \end{cases} \quad (3.3)$$

Yolun tehditlerle kesişip kesişmediği belirlenir. Yol tehditlerle kesişiyorsa kullanılacak c_1 ceza sabiti, Bölüm 4'te uygulanan testlerde, geçerli yolların uygunluk değerlerinin ulaşamayacağı bir değer olarak 1000 değeri kullanılmıştır.

Amaç fonksiyonları, farklı veri kaynaklarını kullanabilir. Yaygın kısıt işleme tekniği ceza puanı yöntemidir [313]. Literatürde çoğu çalışmada hava aracının yükselme durumunda bir ceza puanı eklenmektedir [314, 315]. Yol uzunluğu optimizasyonunu esas alan amaç fonksiyonu denklemi:

$$f_c = \sum_{i=0}^n d_i \cdot c_i + P_d \quad (3.3)$$

$$c_i = \begin{cases} 1 & A_i = A_{i-1} \\ 1.25 & A_i > A_{i-1} \end{cases} \quad (3.4)$$

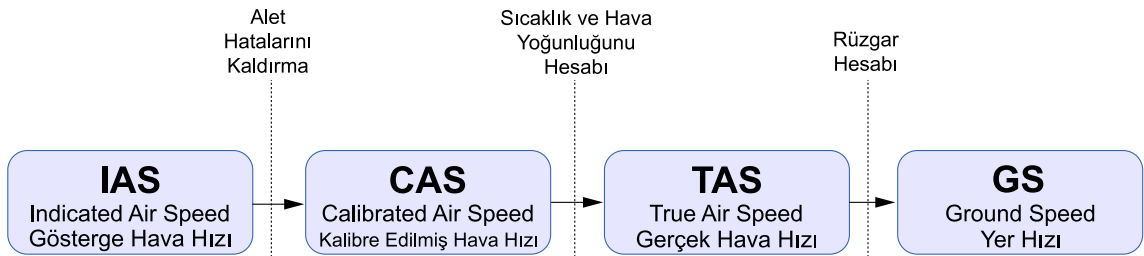
Önerilen algoritmalarda tercih edilen yöntem, hava aracı performans modeli temelinde bir yaklaşım ile hava aracı hareketinin gerçeğe yakın bir değerlendirmesine odaklanır. Tez kapsamında önerilen yöntemler için uygunluk değeri denklemi:

$$f_c = \sum_{i=0}^n d_i \cdot c_{(V_i, A_i, T)} + d_i * 10^{-3} + P_d \quad (3.5)$$

şeklindedir. Amaç fonksiyonu f_c denklemi (E.3.5) içerisinde; d_i , yol uzunluğu (m); V_i , hız; A_i , irtifa; T , seyir, yükselme ve alçalma olarak uçuş fazı; $c_{(V_i, A_i, T)}$, her irtifada yükselme seyir ve alçalma için önceden hesaplanmış g/m cinsinden yakıt değeridir. Burada hesaplanan yol uzunluğu (d_i) harcanan yakıtın hesaplanmasında kullanılır. Buna ek olarak birbirine çok yakın yakıt tüketimine sahip yolların karşılaştırmasında daha kısa olanın tercih edilmesini sağlamak için 10^{-3} çarpanı ile uygunluk değerine eklenir. d_i 'nin bir çarpanla yeniden eklenmesi ile kapsamın algısal olarak netleştirilmesi de amaçlanmıştır. Uygunluk değerinin daha doğru bir değer üretebilmesi için V_i 'nin, Şekil 3.10'da gösterildiği gibi yer hızı alınması tercih edilmektedir. Tırmanma ve alçalma fazlarında (tırmanma ve alçalma oranı düşük olduğu için üretilen değere etkisi az olsa da), yere göre öklid mesafesi hesaplanarak gerçek hız ve yol değerleri kullanılmıştır.

Bölüm 4'de tez kapsamında önerilen hava aracı performans modeli temelinde yol planlama ile birlikte yol uzunluğu optimizasyonunu esas alan yol planlama testlerine karşılaştırma amaçlı yer verilmiştir.

Çalışmamızda, Bölüm 3.5'te açıklanacağı üzere İHA uçuş verileri, hava aracının MATLAB modelinden elde edilerek, İHA'nın uçabileceği her irtifa için (25 m hassasiyette) yükselme seyir ve alçalma yakıt sarfiyatları tespit edilmiş $c_{(V_i, A_i, T)}$ sabiti amaç fonksiyonunda kullanılmıştır. Böylece, hesaplama maliyeti azaltılarak çözüm süresi kısaltılmıştır.

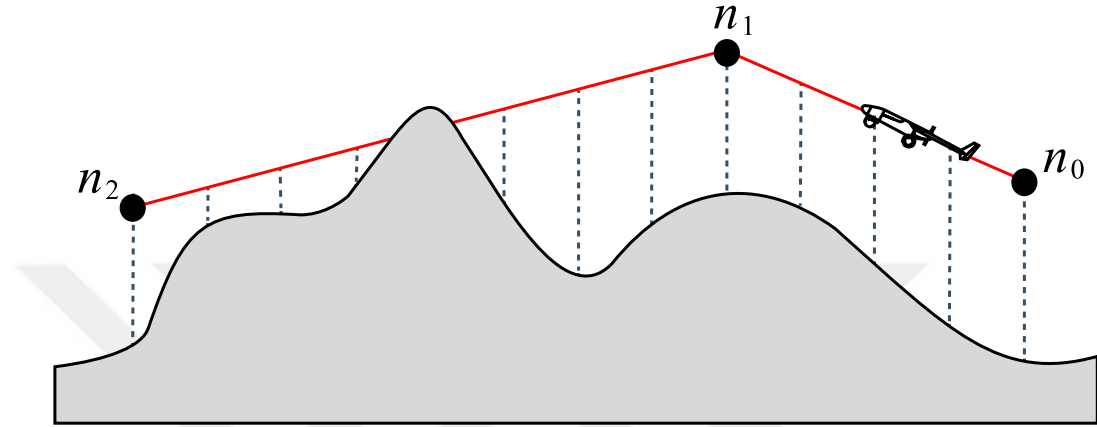


Şekil 3.10. IAS, CAS, TAS ve GS ilişkisi

Önerilen yaklaşım ile 3D yollar üretilmesi hedeflenmektedir. Bu yöntemde sabit irtifa (2D) üretilen bir yol, güzergâhtaki yeryüzü yükseltileri veya engeller dikkate alınarak irtifa düzenlenmektedir [316]. İHA kinematik kısıtları dikkate alınarak yükselme ve alçalma gereksinimlerine göre 3D yollar hesaplanmaktadır. İrtifa belirlenirken hava

aracının en ekonomik uçuş sağlayacağı irtifa ve hız değeri tercih edilmektedir. Tırmanma ve alçalma oranları da aynı şekilde seçilmektedir.

Algoritmada, tüm çaprazlama ve mutasyon işlemlerinden sonra, irtifa ayarlama prosedürü çalışmakta ve sonrasında uygunluk değeri hesaplanmaktadır. Şekil 3.11’de irtifa ayarlama işlemi öncesindeki bir çarpışmalı yol örneği gösterilmiştir.



Şekil 3.11. İrtifa ayarlama işlemi öncesi çarpışmalı yol gösterimi

Yol, ara noktaların bağlanmasından oluşur. Bu nedenle herhangi bir ara noktadaki yörüngenin yüksekliği o hücredeki zemin yüksekliğine göre belirlenir. Izgara üzerindeki dik veya çapraz tüm yol geçişlerinde öklid mesafesi dikkate alınmıştır.

Uygunluk değerleri hesaplanırken öncelikle basit kontroller ile imkânsızlığı tespit edilebilen yollar imkânsız olarak işaretlenir. Eğer başlangıç irtifası, varış irtifasından, azami tırmanma ve alçalma oranı ile dahi ulaşamayacak kadar yüksekse veya alçakta ise yol imkânsızdır. Daha sonra her yol noktasının yüksekliği, başlangıçtan varış noktasına sabit ivmeli hareket ile yükseliş veya alçalma çizgisi oluşturacak şekilde belirlenir. Ekonomik uçuş, seçilen hava aracında daha düşük irtifalarda gerçekleştiği için ek bir kontrol yapılmaz.

Yol adım adım taranır. Her noktada yolun imkânsızlığa düşüp düşmediği kontrol edilir. Eğer nokta, başlangıç veya varış noktasına göre yükselinemez veya alçalınamaz bir zemin yüksekliğine sahipse veya nokta, bir engel üzerindeyse yol imkânsız olarak işaretlenir (scGA(R)’de kromozom parçası da işaretlenir). Diğer durumlarda zemin yüksekliği, mevcut irtifadan yüksekse tepe noktalarının tutulduğu bir dizi değişkene eklenir. Böylece, her nokta kontrol edilmiş olur. Zemin yüksekliğine, istenen asgari yerden yükseklik değeri (AGL) eklenmiştir.

Tablo 3.13. Hava aracı performans modeline dayalı amaç fonksiyonu

S.N.	İşlev
1	Eğer (başlangıç irtifası, varış irtifasından; azami yükselme ve alçalma oranı ile ulaşamayacak kadar yüksekte veya alçakta) ise yolu <i>imkânsız</i> olarak belirle
2	Döngü başlat , yolun her <i>noktası</i> için:
3	<i>nokta</i> yüksekliğini başlangıçtan varış noktasına sabit ivmeli bir yükseliş veya alçalma hareketinde sahip olacağı irtifa olarak belirle, (ekonomik uçuş irtifası kapsamında)
4	Eğer <i>nokta</i> , başlangıç veya varış noktasına göre yükselinemez veya alçalınamaz zemin yüksekliğinde veya engel üzerindeyse yolu <i>imkânsız</i> olarak belirle
5	Değilse:
6	Eğer <i>nokta</i> , <i>nokta</i> irtifasından yüksek ise “ <i>tepe noktası</i> ” dizisine ekle
7	Döngü sonu
8	“ <i>tepe noktası</i> ”nı büyükten küçüğe sırala
9	Döngü başlat , “ <i>tepe noktası</i> ”ndaki her <i>düğüm</i> için:
10	Eğer <i>düğüm</i> , zeminden yüksek ise:
11	Eğer yükselme veya alçalma ihtiyacı yükselme ve alçalma oranı ile uyumlu ise:
12	Yolun her <i>noktası</i> için gereken irtifayı tanımla
13	Değilse:
14	Yolu <i>imkânsız</i> olarak belirle
15	Döngü sonu
16	Eğer yol <i>imkânsız</i> değilse:
17	Döngü başlat , Yolun her <i>noktası</i> için:
18	<i>nokta</i> irtifasında, gereken yükselme ve alçalma oranını sağlayacak en ekonomik hızı seç ve tanımla
19	<i>noktanın</i> irtifa, hız, yükselme-alçalma oranı ve faza göre (yükselme, alçalma veya seyir) her hücrenin gerçek boyutuna ölçeklenmiş yakıt miktarını <i>uygunluk değeri</i> toplamına (scGA(R) için parça uygunluğuna da) ekle (amaç fonksiyonundaki gibi d_i de 10^{-3} çarpanı ile eklenir)
20	Eğer <i>noktadaki</i> hız, önceki <i>noktaya</i> göre artmış ise hız artışı puanı ekle
21	Döngü sonu
22	Eğer yol <i>imkânsız</i> ise:
23	<i>uygunluk değeri</i> toplamına, yol uzunluğunu temsili bir yakıt değeri ile çarpımı ve ceza puanı ekle (scGA(R) için parça uygunluğuna da ekle)

Tepe noktaları dizisi, büyükten küçüğe doğru sıralanır. Tepe noktaları dizisindeki her düğüm için, düğümün irtifası zeminden alçaktaysa, geriye ve ileriye doğru her yol noktası için yükselme ve alçalma ihtiyacına göre gereken yükselme ve alçalma oranı kontrol edilir ve bu oran, İHA ile uyumluysa irtifa, yola adım adım tanımlanır.

Böylece, eğer yol imkânsız değil ise tüm yol noktalarının irtifaları belirlenmiş olur. Bir sonraki aşama; yükselme ve alçalma ihtiyacına göre irtifaya özgü en ekonomik hızın seçilmesidir. Her yol noktası için, belirlenmiş yol irtifasında gereken yükselme ve alçalma oranını karşılayabilecek en ekonomik hız yola tanımlanır. Aynı şekilde her yol noktası için; irtifa, hız, yükselme-alçalma oranı ve faza göre (yükselme, alçalma veya seyir) yolun gerçek uzunluğuna ölçeklenmiş yakıt miktarı, uygunluk değeri toplamına eklenir. $scGA(R)$ 'de bu değer, kromozom parça uygunluğuna da eklenir. Eğer hız, önceki noktaya göre artmışsa noktanın uygunluk değerine hız artışı puanı eklenerek uygunluk değerinin hesaplanması tamamlanır.

Eğer yol imkânsız ise yol için yukarıdaki işlemlerin hiç birisi yapılmadığından ve bir uygunluk değeri hesaplanmamış olduğundan yol uzunluğu, temsili bir yakıt değeri ile çarpılarak bir uygunluk değeri hesaplanır. Buna ceza puanı eklenir. $scGA(R)$ 'de bu değer, kromozom parça uygunluğuna da atanır. Böylece, uygunluk değerleri karşılaştırması için yol uzunluğu ile ilişkili bir referans değer elde edilmiş olmaktadır.

3.3.6. Başlangıç popülasyonu

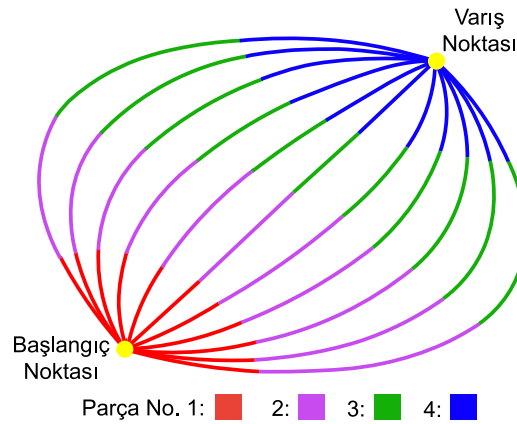
Popülasyon, terimsel olarak insan nüfusuna benzer bir anlamda kullanılmaktadır. Bir İHA için oluşturulan aday yolların tamamı popülasyon kabul edilmektedir. Başlangıç popülasyonunun oluşturulması, GA'nın önemli aşamalarından biridir ve çözüme daha erken yakınsamaya etki etmektedir. Başlangıç popülasyonunun rastgele seçilmesi kolay ve nispeten hızlıdır. Öte yandan, hesaplama süresinde artışı göze alan farklı metodolojiler bulunmaktadır. Popülasyon boyutu, çözümün optimize edilebileceği büyüklükte olmalıdır. Ancak büyüyen popülasyon miktarı, daha fazla hesaplama anlamına gelir ve çalışma süresini artırır. Bu nedenle uygulamalarda kaliteden ödün vermeyen yeterli bir popülasyon boyutu belirlenmektedir.

Başlangıç popülasyonu farklı şekillerde oluşturulabilir. Noktalar için bir alt ve üst sınır tanımlanır ve aralık içerisinde rastgele değerler üretilir. Bu yaklaşım, popülasyon çeşitliliğini artırırken çözüm alanını kapsamaya imkân vermektedir. Buna karşın kromozomların boyutu büyür ve GA'nın yakınsaması gecikir. Bu sorunu azaltmak için, başlangıç popülasyonundaki uygunluk değerlerine göre belirli bir eşiğin altına düşen bireyler atılabilir ve yerlerine yenileri oluşturulabilir. Bu işlem yakınsama süresini iyileştirebilmektedir.

Rastgele yollar üretilmesi basittir ancak yol planlaması problemi özelinde büyük oranda kullanılması uygun olmayan yollara neden olmaktadır. Uygulanabilir olmayan bu yollardan bazıları, yinelemeler ile genetik operatörler çalıştıkça uygulanabilir çözümler haline gelebilse de hesaplama süresi açısından büyük bir dezavantajdır.

GA'da başlangıç popülasyonu birçok çalışmada genellikle rastgele seçilir [117, 317, 318]. Rastgele olmayan başlangıç popülasyonu ile yapılan çalışmaların amacı, hesaplama süresini azaltmak, en iyi çözüm için amaçsız aramayı önlemek ve popülasyonun gelişim sürecini en verimli hale getirmektir. Lin ve Goodrich, (2009) çalışmalarında, başlangıç popülasyonunda çeşitli algoritmalar kullanılarak oluşturulmuş 100 yol ve 95 rastgele yol kullanmışlardır [319]. Sun vd., (2008) 3D çalışmalarında, başlangıç popülasyonu için tüm topografyayı tarayan bir ön-işleme prosedürü olarak iskeletizasyon algoritmasını kullanmış ve daha kaliteli bir başlangıç popülasyonu elde etmişlerdir [320]. Pehlivanoglu, (2012) çalışmasında başlangıç popülasyonunun 3B uzayda Voronoi diyagramı kullanılarak iyileştirildiğini göstermektedir [64]. Diğer bir çalışmada, başlangıç popülasyonunu oluşturmak için bir başlangıç tohumu kullanılır. Nominal çalışma koşullarında temel yörünge izleme yeterliliği ve kararlılığı sergilediği doğrulanan tohum, popülasyona birey olarak eklenir. Sonraki bireyler geleneksel yöntemler ile oluşturulur [321].

Önerilen yöntemde, başlangıçtan varış konumuna düz çizgi olan en kısa yol ile sağında ve solunda 20'şer adet olmak üzere yarım yay şeklinde yollar sabit başlangıç popülasyonu olarak kabul edilmektedir (Bkz. Şekil 3.12).



Şekil 3.12. cGA ve scGA'nın başlangıç popülasyonu ile scGA'daki kromozom (yol) parçaları

GA, scGAR ve cGAR algoritmalarında 55 rastgele başlangıç popülasyonu oluşturulur. cGA ve scGA için 41 sabit 14 rastgele başlangıç popülasyonu

oluşturulmaktadır. Bölüm 3.3.1’de algoritma içerisinde açıklandığı gibi 14 rastgele popülasyon, her kromozomun en veya boy değerinin sırasıyla; 1/5, 2/5, 3/5 ve 4/5’üne denk gelen alellere göre sıralanarak belirli aralıklar ile sabit popülasyonun arasına eklenir. Sıralamanın amacı kromozomların benzer çözümler ile mümkün olduğunca aynı mahallede olmalarını sağlamaktır.

3.3.7. Parametreler

Parametreler incelenen probleme özgü olduğundan genel bir tanımlama yapılamaz. Çaprazlama ve mutasyon gibi mevcut cGA’da genetik operatörleri için parametrelerin dinamik olarak güncellendiği çalışmalar [322] bulunmakla beraber birçoğu önceden belirlenmiş sabit parametreler ile çalışmaktadır [323, 324]. Parametrelerin optimizasyonu önemlidir. Örneğin, aşırı yüksek mutasyon ihtimali popülasyonu bozabilirken düşük değerler; algoritmanın yerel bir optimuma yakınsamaya, arama alanını keşfetmekte başarısızlığa ve en iyi sonuçlara ulaşamamaya neden olmaktadır.

GA performansına etki eden bir dizi parametreden bahsedilebilir. Bu parametrelerden öne çıkanları popülasyon büyüklüğü, çaprazlama ve mutasyon oranıdır. Genellikle, popülasyon büyüklüğü 50 ile 200 arasındadır. Popülasyon ne kadar büyükse çeşitlilik o kadar artmaktadır. Çaprazlama oranı, çaprazlama operatörünün ne sıklıkla uygulanacağını belirlemektedir. Yaygın olarak; çaprazlama oranı değeri 0,60 ile 1,00 arasında, mutasyon oranı değeri 0,05 ile 0,15 arasında kullanılmaktadır.

Tablo 3.14. *GA parametreleri*

Parametre	Değer
Popülasyon boyutu	55
Seçim oranı	1
Mutasyon oranı	0,1
Azami yineleme	30
Ebeveyn seçimi	Rulet tekerleği seçimi
Rekombinasyon	İki nokta çaprazlama
Rekombinasyon oranı	1,0
Mutasyon	Tek nokta
Yeniyle değiştirme	Daha iyiyse değiştir, scGA: OPRO
Durdurma kriteri	30 yineleme
cGA ızgara boyutu	55x1
cGA mahalle türü	Çizgisel
cGA mahalle boyutu	2, 3, 4, 5, 6, 7, 8, 9
scGA ızgara boyutu	55x2- <i>n</i> (kromozom uzunluğuna göre dinamik belirlenir)

Önerilen algoritmada parametreler Tablo 3.14’de gösterilmiştir. Önerilen yaklaşımda popülasyon büyüklüğü 55, çaprazlama oranı 1, mutasyon oranı ise 0,1 olarak belirlenmiştir. GA, 30 yineleme sayısına ulaşıldığında sona ermektedir. Kullanılan rekombinasyon yöntemi, çift nokta çaprazlamadır. Bir ebeveyn, yalnızca yavrunun uygunluk değeri daha iyiye değiştirilir buna ek olarak scGA’da OPRO kullanılmaktadır.

3.4. Hava Aracı Performansı

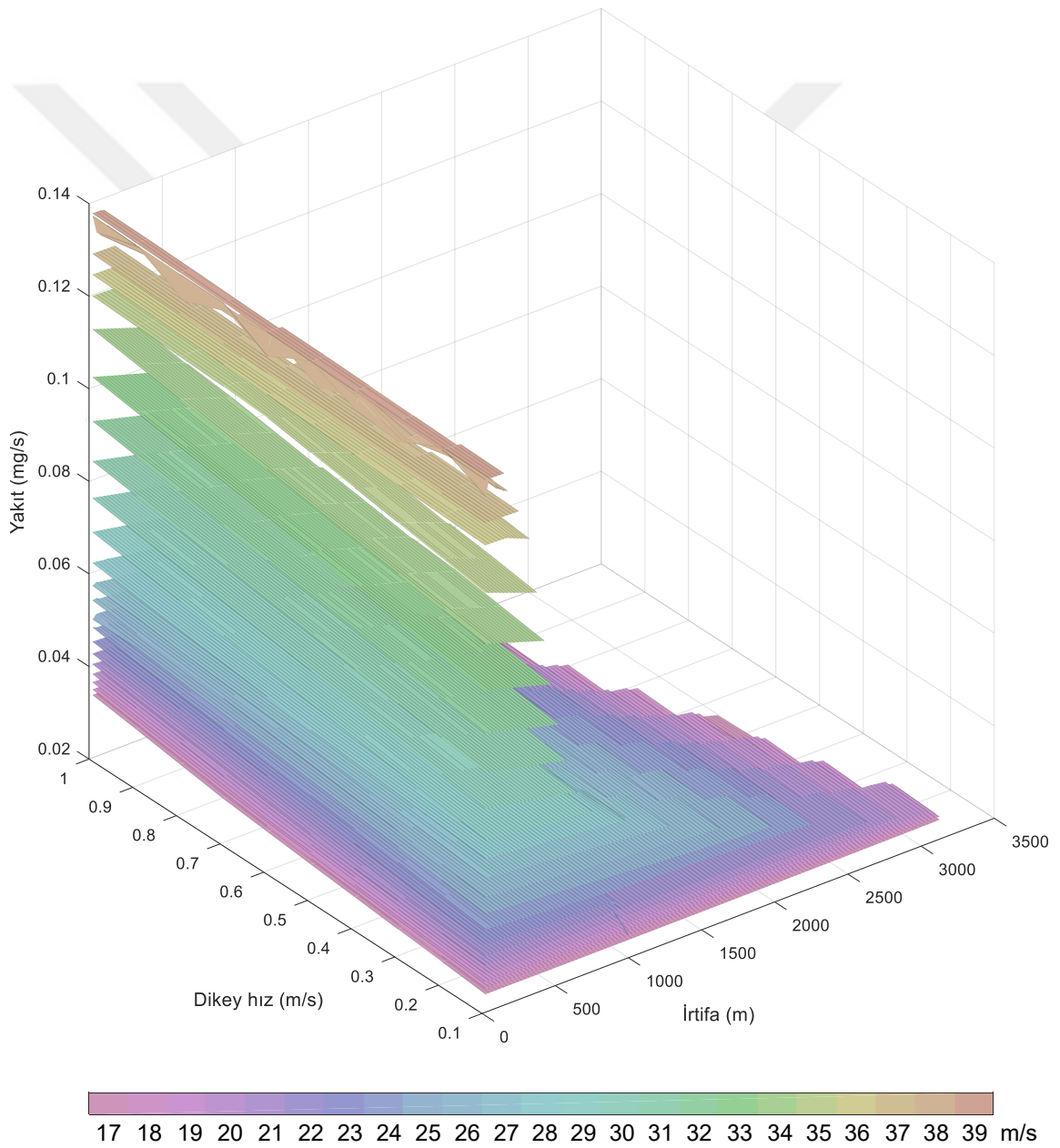
Gerçek hayatta İHA’lar 3D ortamda uçmaktadır. Hesaplama kolaylığı açısından çevre, 2B görüldüğünde yol planlayıcı, birçok zaman optimal olmayan sabit irtifa bir yol üretmektedir. 3B ortamda yol planlayıcı mesafe olarak daha kısa ve ekonomik yollar üretebilir. Sabit irtifada uçan bir İHA’nın yükselmek yerine, engellerden kaçınmak için yönünü yalnızca sağa veya sola değiştirmesi gerekmektedir; 3B ortamda yükselme fazında İHA, yerçekimine karşı itkiyi artırmak için ekstra enerji tüketmektedir. Yol planlayıcının seyir, yükselme ve alçalma fazları için hava aracı performansını doğru değerlendirmesi aday yolların karşılaştırılmasında başarıyı artırmaktadır. Bu kapsamda, tez çalışmasında önerilen yol planlama algoritmasının başarılı yollar üretebilmesi için seçilen bir İHA’ya ait uçuş verilerine ihtiyaç duyulmuştur. Bu amaçla Aerosonde İHA seçilmiştir.

İnsansız hava sistemlerinin tasarımı ve analizini kolaylaştırmak için MATLAB Simulink simülasyon ortamı kullanılmaktadır. Hava aracı modelleri, yörünge izleme kontrolörleri ve yörünge planlayıcılarının farklı konfigürasyonları için esnekliğe sahip modüler bir tasarıma sahiptir. Hava aracı modelleri, nominal koşulların yanı sıra türbülans koşulları ve sensör arızaları dâhil olmak üzere farklı koşullarda çalışacak şekilde tasarlanmıştır [325]. Aerosonde İHA’nın algoritmalarda kullanılacak kısıtları Tablo 3.15’te listelenmiştir.

Tablo 3.15. İHA fiziksel kısıtları

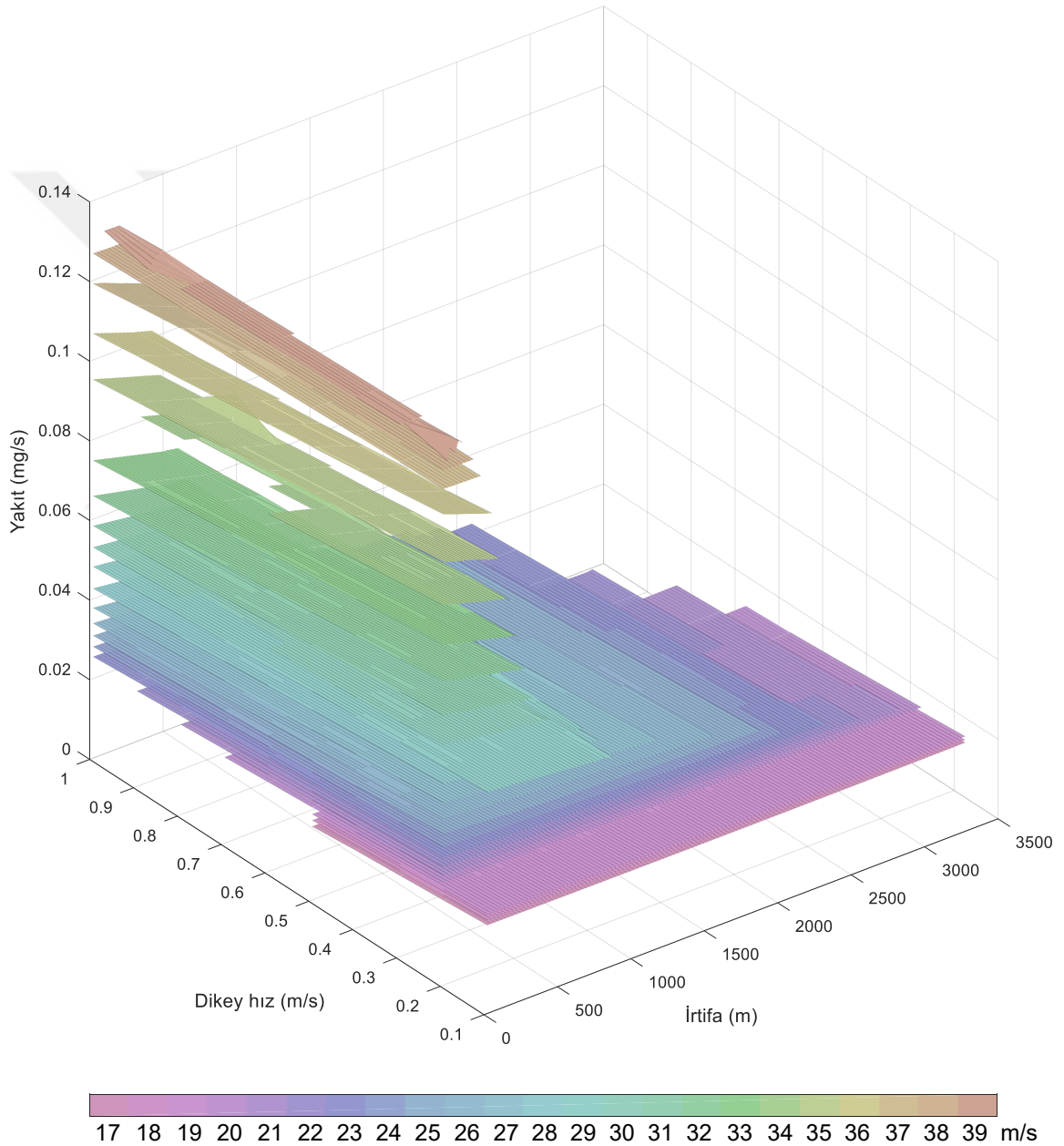
Kısıtlama	Değer
Seyir hızı	17-39 m/s
Yunuslama açısı aralığı	-15°/30°
Yatış açısı aralığı	-15°/15°
Gaz kolu pozisyonu	0,1/1,0
Yükselme ve alçalma oranı aralığı	-1/1 m/s
Asgari yerden yükseklik (AGL)	200 m
Ayrırma mesafesi	200 m

Aerosonde İHA sistemi, MATLAB'da Aerosim Havacılık Simülasyon Blok Seti kullanılarak bir dizi test uçuşunun simüle edilmesiyle modellenmiştir [326]. AeroSim® simülasyon ortamı; dünya, atmosfer, aerodinamik, atalet ve hareket denklemi alt modellerinden oluşan eksiksiz, ayrıntılı, 6 serbestlik dereceli ve doğrusal olmayan genel bir İHA modeline sahiptir. Aerosonde, Washington Üniversitesi'ndeki Uçuş Sistemleri Laboratuvarı tarafından MATLAB Simulink gerçek zamanlı atölye ortamına dayalı hızlı bir kontrol kuralları prototipleme platformunun geliştirilmesi için bir test ortamı olarak kullanılmıştır [292].

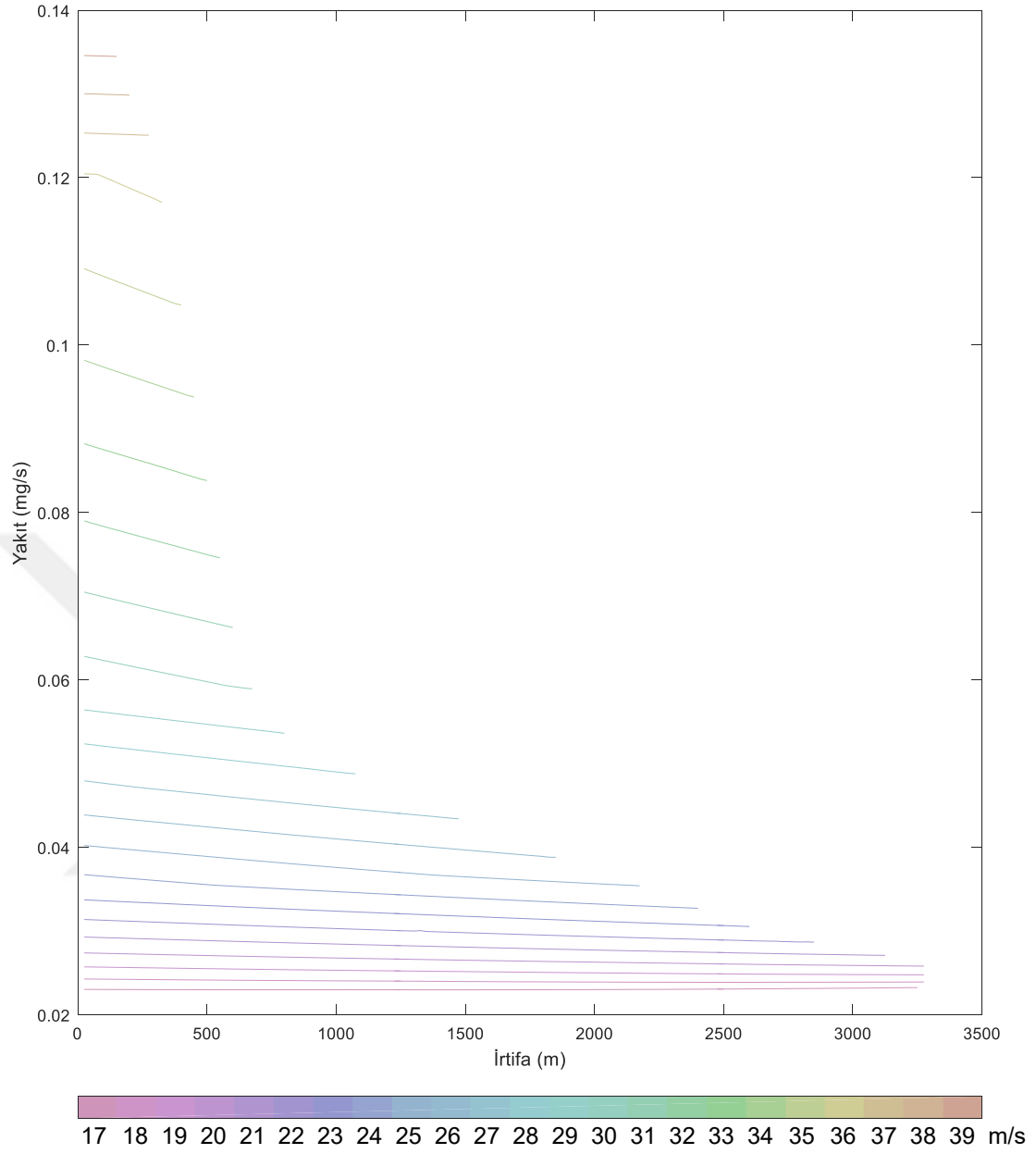


Şekil 3.13. Aerosonde İHA'ya ait turmanma fazı yakıt karakteristiği

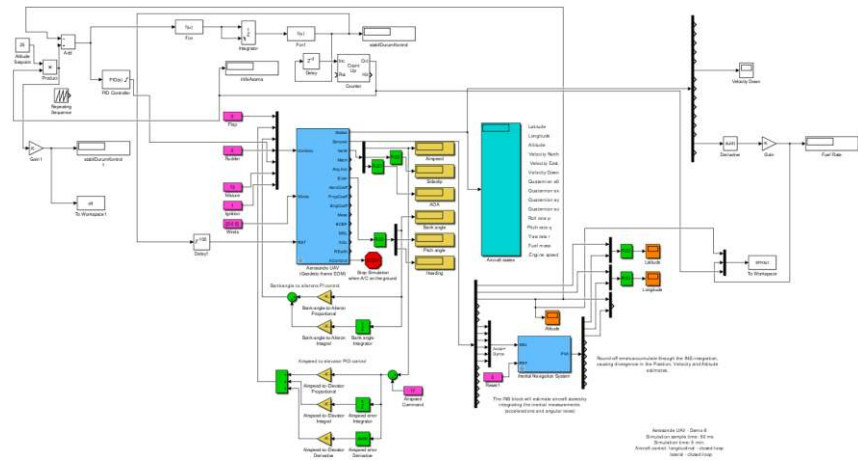
Şekil 3.16’de gösterilen Aerosonde İHA’ya ait MATLAB Simulink modeli kullanılarak hava aracının havada kalabildiği her hız ve her irtifada; tırmanma (Şekil 3.13), alçalma (Şekil 3.14) ve seyir (Şekil 3.15) fazları için yakıt verileri elde edilerek yol planlama algoritmasında kullanılmıştır. Hava aracının sağa veya sola dönüşünde oluşan yol uzunluğuna dayalı yakıt farklılığı dikkate alınmıştır ancak manevra kaynaklı yakıt tüketimi farklılığı ihmal edilmiştir.



Şekil 3.14. Aerosonde İHA'ya ait alçalma fazı yakıt karakteristiği



Şekil 3.15. Aerosonde İHA'ya ait seyir fazı yakıt karakteristiği



Şekil 3.16. Aerosonde İHA'ya ait MATLAB Simulink diyagramı

3.4.1. Kinematik kısıtlar

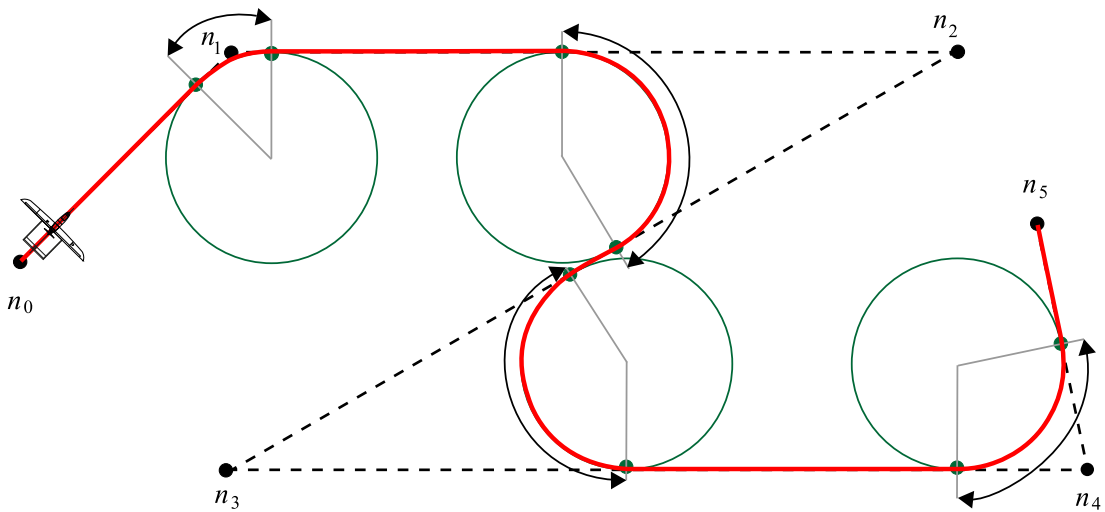
Literatürde uçuş mekaniği açısından çok sayıda kısıt dikkate alınmıştır [327-330]. Uçuş süresi, azami irtifa, azami tırmanma hızı, azami yatış açısı gibi geometrik kısıtlar olarak ifade edilebilecek kısıtlar yol planlama algoritmalarında dikkate alınmaktadır. Önerilen algoritmada da tüm bu kısıtlar değerlendirilmiştir.

Algoritma, dönüş hızına göre yörüngeyi optimize ederek aracın manevra kabiliyeti sınırlamalarını dikkate almaktadır. İHA performans parametreleri hesaplanırken dönüş performansı önemli bir faktördür. Manevra sırasındaki asgari dönüş yarıçapı, hız ve yük faktörü, azami dönüş hızı, hız, hareket sırasındaki yük faktörü hızı dikkate alınmaktadır [330]. Algoritma, yalnızca izin verilen dönüş hızları aralığında yörüngeler oluşturarak aracın manevra kabiliyetini aşmamasını sağlamaktadır. Bu parametreler için denklemler aşağıda verilmiştir.

$$R_{min} = \frac{4K(\frac{W}{S})}{g\rho(\frac{T}{W})\sqrt{1 - \frac{4KC_{D0}}{(\frac{T}{W})^2}}} \quad (3.6)$$

$$R_{min} = \frac{V^2}{19 \tan \theta} \quad (3.7)$$

Denklem (E.3.6) ile asgari dönüş yarıçapı (R_{min}) için yüksek doğrulukta bir hesaplama yapılabilir. Ancak Denklem (E.3.7) [331] kullanılarak elde edilecek hassasiyet işlem kolaylığı dikkate alındığında algoritma için yeterlidir. Burada V , m/s cinsinden gerçek hava hızı; R , metre cinsinden dönüş yarıçapı; θ , yatış açısıdır.



Şekil 3.17. İHA'nın dönüş noktalarındaki hareketi ve dönüş yarıçapı gösterimi

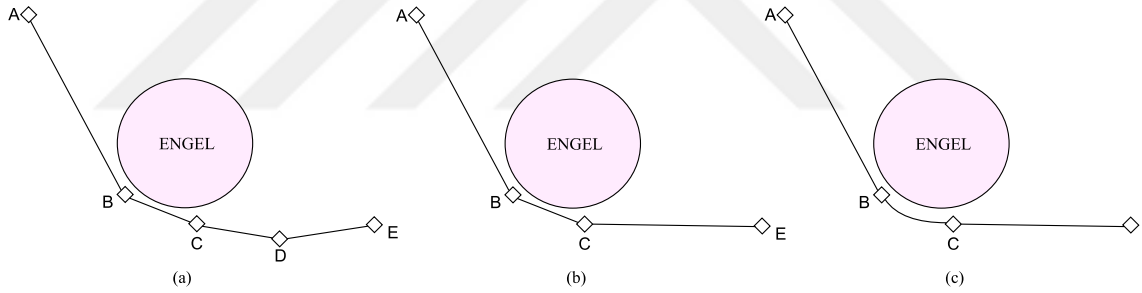
İHA'nın dönüş noktalarında başlayan hareketi, Dubins eğrilerini oluşturur. Bu durum Şekil 3.17'de gösterilmiştir.

3.5. Planlama Sonrası Yol Yumuşatma

Tez çalışmasında uygulanan yol yumuşatma algoritması iki yol noktası seçer ve bu yol noktaları arasında çarpışmasız bir ara bağlantı mümkünse, bu iki yol noktası arasındaki ara noktalar kaldırılır.

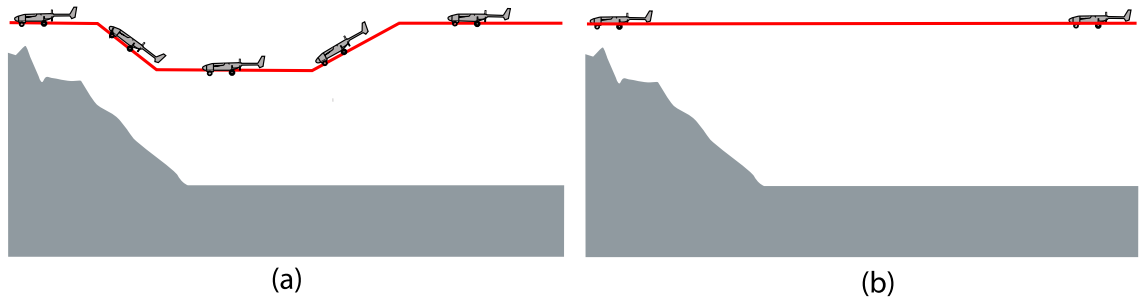
Genelde başlangıç noktası ile varış noktası arasındaki en kısa yol düz bir çizgi olmaktadır. Bu nedenle regresyon operatörü olarak da isimlendirilebilecek işlem mesafeyi kısaltmak amacıyla oluşturulan yolun en kısa yol olan düz çizgiye yakınsamasını sağlamaktadır.

Her iki yol noktasının uzunluğu taranır aralarından mevcut olandan daha kısa mesafe bulunursa aralarındaki yol noktaları silinir ve yeni bir yol oluşturmak üzere birleştirilir.



Şekil 3.18. Yol yumuşatmada silme operatörünün uygulanması

Şekil 3.18'de gösterilen örnekte üretilen yolda (a)'daki gibi gereksiz bir D noktası bulunması durumunda yol (b)'deki gibi düzeltilmekte, İHA'nın hareketi (c)'deki gibi olmaktadır.



Şekil 3.19. İrtifanın düzenlenmesi örneği

Üretilen yolda Şekil 3.19’de bir örneği gösterilen gereksiz alçalma ve yükselme durumları varsa planlama sonrası yumuşatma algoritması ile düzeltilmektedir.

3.6. Gerçek Zamanlı Uygulama

Gerçek bir uçuşta ilk yol planlama, kaba planlama şeklinde ifade edilebilir. Kaba planlayıcı, İHA’nın mevcut konumundan varış noktasına giden bir yol oluşturmaktadır [332]. Modern İHA’larda, insanların engellerden kaçınmasına benzer olarak uçuş esnasında ortaya çıkan engellerden kaçınan bir algoritma bulunmaktadır. Bu algoritma reaktif algoritma olarak da ifade edilebilir. Kaba planlayıcı tarafından oluşturulan ilk yol temel alınarak gerektiğinde reaktif algoritma tarafından ara hedefler eklenerek güncellenir. Bir yolun oluşturulması için gereken hesaplama süresi, belirlenen bir ara hedefe giden yol parçasının, İHA tarafından geçilmesi için gereken süreden küçük olması durumunda yol planlayıcı gerçek zamanlı kabul edilmektedir. Gerçek zamanlı yol planlayıcı, İHA hızına ve işlem gücüne bağlı olarak ara hedefe ulaşmak için gereken süre içinde uygun bir yol planlanabilmelidir [7].

Ani durumlarda doğru tepki verebilmek sadece sistemin zekâsına değil, aynı zamanda bir acil durum manevrasının ne sürede uygulanabileceğine de bağlıdır. Bu süre her İHA için farklıdır. İHA’nın hızı ve kinematik kısıtlarının yanı sıra platformunun ne uzunlukta bir menzili tarayabildiği etkin faktörlerdir.

Yol planlamada, uçuş platformu üzerinde gerçek zamanlı çalışan algoritmaların hesaplama açısından verimli olması oldukça önemlidir. Gerçek zamanlı algoritmalarda çoğu durumda ideal hesaplama süresinin milisaniyeler cinsinden olması gerekmektedir. Buna karşın daha uzun sürelerde hesaplama süresine sahip algoritmaların da gerçek zamanlı kullanılabileceğinin gösterildiği çalışmalar bulunmaktadır [9].

Gerçek zamanlı bir yol planlayıcı, varış noktasına uygun bir yol oluşturmak için sınırlı bir süreye sahiptir. Döner kanatlı İHA’larda olduğu gibi havada asılı kalma imkânı olsa bile böyle bir durum tercih edilmez. Gerçek zamanlı bir yol planlayıcısı için havada durmak bir seçenek değildir.

Gerçek zamanlı yol planlamasında zamansal darboğaz durumunda yineleme sayısı sınırlandırılabilir. Yineleme sayısının azaltılması optimum çözüm olmasa bile geçerli çözümlerin üretilmesi ile gerçek zamanlılığın korunmasını sağlamaktadır. Buna karşın yineleme süresinin sınırlandırılması da bir yöntemdir [94]. Süre sınırlandırılması ile

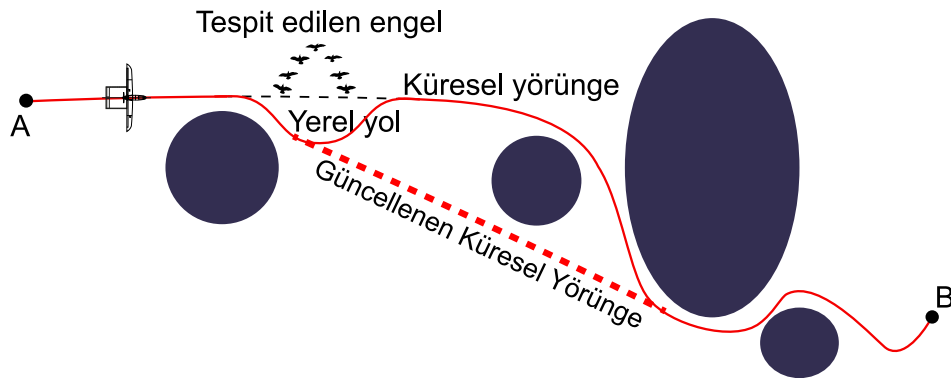
yinelemede bir çözüm bulunamasa bile sonraki yinelemeler bu durumu telafi edebilmektedir.

Gerçek zamanlı yol planlama, çevrenin yüksek doğrulukta modellenmesini gerektirir. Gerçek zamanlı yol planlamayı kolaylaştırmak için gerçek ortamlarda haritalar oluşturan Eşzamanlı Yerelleştirme ve Haritalama (SLAM) algoritmaları kullanılır. Bu tür algoritmalar, GPS bilgisinin kısmen veya tamamen yokluğunda veya sensör bilgilerinin süreksizliğinde işlevseldir [333].

Yol planlama algoritmasının çalışması ve optimum yolu bulması nispeten uzun sürebileceğinden gerçek zamanlı yol planlamasının uygulama aşamaları şu şekilde gösterilebilir [334]:

- 1) İHA bir çarpışma olmaksızın başlangıç noktasından varış noktasına daha önce belirlenmiş yol planlaması doğrultusunda planlanan şekilde hareket etmektedir.
- 2) Bir engel tespit edildiğinde gerektiğinde ani yol değişikliği yapılır ve İHA'nın t saniye kadar hareketi yol yönünde planlanır.
- 3) Daha sonra t saniye sonra ulaşılacak koordinat yeni bir başlangıç noktası olarak kabul edilir ve yeni bir yol bulmak için algoritma çalıştırılır.
- 4) Eğer t saniyenin bitiminden önce algoritma başarılı bir şekilde tamamlanırsa, İHA güzergâhı üretilen bu yol ile değiştirilir. Aksi takdirde 3. aşamaya dönlür.

Bu çalışma varış noktasına ulaşana, algoritmanın çalışması için yeterli süreye sahip olana ya da başarılı bir yol bulunana kadar devam etmektedir.



Şekil 3.20. İHA'nın karşılaşılan bir engel durumunda hareket örneği

10 saniyelik bir hesaplama süresi için 20 m/s seyir hızında bir İHA, bir engelin algılanmasından yeni yörünge bulunmasına kadar geçen sürede 200 metre yol kat etmektedir. Böyle bir durumda hesaplanacak olan yol başlangıcı, refleks gösterilen yolun

en az 200 metre ilerisinde olmalıdır. Şekil 3.20’de İHA’nın karşılaşılan bir engel durumunda hareket örneği gösterilmiştir.

Genel bir değerlendirme olarak, İHA yeterli derecede bir görüş mesafesindeki engelleri tespit edebildiği sürece, tamamen yeni bir yörüngeyi yeniden hesaplayarak engellerden kaçınabilecektir.

3.7. Çoklu İşlem Kullanımı

EA, çoklu kısıtlarla karmaşık optimizasyon problemlerini çözmek için yaygın bir yöntemdir. Ancak, yinelemeli yapısı nedeniyle en iyi sonuca ulaşmada çözüm süresi uzayabilmektedir. İHA yol planlaması, genel olarak sınırlı bir sürede çözülmesi gereken gerçek zamanlı bir problemdir. EA’da hesaplama süresini azaltmak için kullanılan etkili yöntemlerden birisi de paralelleştirmedir. Verimliliği artırmak için çoklu işlem kullanılsa da PGA’nın temel kuralları seri bir GA ile aynıdır.

Hesaplama karmaşıklığı yüksek problemlerde sayısal verimlilik konusu önemlidir. Bilgisayar donanımı alanındaki son gelişmeler, paralel hesaplama süreçlerinin cazibesini artırmıştır. Çoklu hesaplama mimarilerinden yararlanmak için algoritmaların paralel çalışmaya uyarlanması gerekmektedir. Dağıtılmış işlemci yükünde bilgiyi koordine etme ihtiyacından dolayı paralelliğin sağladığı getiri sınırlı olmaktadır. GA tabanlı algoritmaların popülasyon temelli olmaları nedeniyle paralel uygulamaya yatkın olduğu söylenebilir. PGA, her sürece tek popülasyon atanarak [242] veya birden fazla popülasyonu paralel olarak geliştirip bireylerin aralıklarla göç etmesine izin vererek [335] tasarlanabilir. Yapılandırılmış popülasyon yapısı, GA’yı paralel uygulamalar için ideal hale getirmektedir. Mahalle yapısı cGA’ya paralel kullanımda bir avantaj sağlamaktadır. cGA’nın, paralel donanım üzerinde uygulama kolaylığı popülerliğini artırmıştır [245, 336, 337]. cGA, yalnızca aynı mahalledeki bireyler arasında iletişim gerektirdiğinden, paralel bir uygulama için oldukça uygundur.

Roberge, Tarbouchi ve Labonté, (2018) GPU üzerinde çalışan bir GA ile ortalama irtifayı en aza indirerek yakıt tüketimini azaltmayı amaçlamışlar ve yarı optimal çözümler üretmişlerdir [290].

Hossain, Magierowski ve Messier, (2014) belirli noktaları ziyaret etmek için yarı optimal yollar hesaplayabilen, GPU üzerinde çalışan, tek çekirdekli bir CPU’ya kıyasla 4.8x’lik bir hızlanma ölçtükleri GA tabanlı bir planlayıcı tanıtmışlardır [338]. Cekmez,

Ozsiginan ve Sahingoz, (2014) aynı problem için GPU üzerinde paralel bir ACO ile 13x hızlanma elde etmişlerdir [339].

Bu tez çalışmasında önerilen yaklaşım, mahalle yapısı ile çok çekirdekli CPU'nun hesaplama gücünden tam olarak yararlanmak için süreçler arasındaki iletişimi en aza indirir ve hesaplama süresini önemli ölçüde azaltmaktadır. Böylece, seri algoritmaya kıyasla daha iyi sonuçlar vermektedir. Çoklu işlem kullanımda cGA(R) ve scGA(R)'de mahalleler arasında bilgi geçişinden kaynaklanan iş yükü, hız artışı, geleneksel bir GA'ya kıyasla bir derece daha düşük kılınmaktadır ancak hızdan ödün vererek elde edilen başarı çok daha yüksektir.

Tez çalışması kapsamında üretilen yazılımın PHP dili ile yazılmış olması nedeniyle algoritmalar da aynı dil ile geliştirilmiştir. Tez çalışmasının ilk aşamasında tek süreç için hazırlanan algoritma, sonraki aşamada çoklu işlem için yeniden tasarlanmıştır. Bu aşamada farklı bir dil kullanılmamış ve aynı dil ile geliştirme sürecine devam edilmiştir. PHP dili çoklu işlem desteğine sahip olmadığından bağımsız geliştiriciler tarafından üretilen modüller de istenen performansı sağlayamadığından süreçler arası iletişimde MySQL veritabanı kullanılarak Bölüm 3.3.1'de açıklandığı gibi bir çoklu işlem mimarisi oluşturulmuştur.

3.8. Ortama Ait Haritanın Oluşturulması

Yol planlayıcılar genellikle çevrenin harita ile temsiline bağımlıdır [340]. Yol planlamadaki ilk adım, algoritma tarafından kullanılabilir bir ortam haritası oluşturmaktır. Zeminin temsil etmek için koordinata göre yüksekliği gösteren 2D bir matris kullanılabilir [9].

Yol planlama için kullanılan en yaygın teknik olan yapılandırma alanı, sistemdeki tüm noktaların ayrıntılı konum bilgilerini sağlamaktadır. Yapılandırma alanı, İHA'nın hareketi için boş olan gerçek alanı temsil eder ve böylece engel ile çarpışmayan yollar mümkün olmaktadır. Yol planlamada yapılandırma alanı yönteminin popüleritesi, çeşitli algoritmaları karşılaştırmak ve değerlendirmek için tek tip çerçeve kullanımından kaynaklanmaktadır.

Web üzerinden erişime sunulmuş farklı harita uygulamaları, koordinatlardaki yükselti bilgilerine erişmek için farklı tekniklere imkân vermektedir. Tekil koordinat veya bir koordinatlar dizisinin yükselti değerlerine ulaşılabilmektedir [341].

Planlama yaparken zemin ile çarpışmayı önlemek için alandaki en yüksek noktanın yükselti bilgisine ihtiyaç duyulur. Tekil yükselti bilgisinde seçilen noktanın yakınında yükselti olmasının dikkate alınamaması bir risktir. Bölgenin taramadan geçirilmesi olarak da ifade edilebilecek bir şekilde, dikdörtgensel bölgeler içerisinde alınacak çok sayıda nokta yükseltilerinin azami değeri, alanın yükselti değeri kabul edilerek uygulamadaki sorunların büyük oranda önüne geçilebilmektedir.

Dikdörtgensel bir bölge içerisindeki çok sayıda noktanın yükseltilerine erişim, geniş bir alanı doğru tarayabilmek için uygun bir yöntem olarak değerlendirilmiştir. Microsoft Bing haritalarında dikdörtgensel alan yöntemi desteklenmektedir [342].

Microsoft Bing haritası üzerinden verileri alacak bir algoritma ile her birim alan içerisinde 64 noktanın azami değeri o birim alanın yükseltisi kabul edilmiştir. Sistem, istenilen koordinat aralığı için yüksek çözünürlükte bu işlemleri otomatik yapabilecek şekilde tasarlanmıştır.

Hazırlanan program ile dünya genelinde istenilen her bölge için yükselti haritası oluşturulabilmektedir. Boylamların kutuplara doğru birbirine yaklaşarak daralması değerlendirilerek belirlenen genişlik için Haversine yöntemi ile boylam bilgileri hesaplanabilmektedir.

Harita üzerinde tüm engel çeşitleri çokgenler şeklinde tanımlanabilmektedir. Bunlara bir örnek olarak radar tehditleri gösterilebilir. Her radarın parametreleri kendine özgüdür. Her radar, 3B silindirik bir tehdit alanına sahip olduğu varsayılarak basitleştirilmiştir [212].

Tez çalışmasında statik engeller olarak yeryüzü yükseltileri Microsoft Bing harita verileri kullanılarak elde edilmektedir. Radar alanları, uçuşa yasak alanlar ve meteorolojik kısıtlar yapılandırma alanına üretilen yol planlama yazılımı kullanılarak yansıtılabilir. Mobil tehditlerin tehlike oluşturduğu alanlar, diğer kısıtlarda olduğu gibi operatör tarafından belirlenebilir.

3.9. Üretilen Yol Planlama Yazılım Paketi

Oluşturulan web uygulaması ile algoritmanın çeşitli parametreleri değiştirilebilir, yeni haritalar oluşturulabilir, rota ve yöntem seçilebilir ve sonuçlar grafiklerle görüntülenebilir. Web arayüzünün hazırlanmasında kullanım kolaylığına dikkat edilmiştir. Her planlama sonrasında grafiklerin incelenmesi ve üretilen yolun simülatöre uygulanması mümkündür.

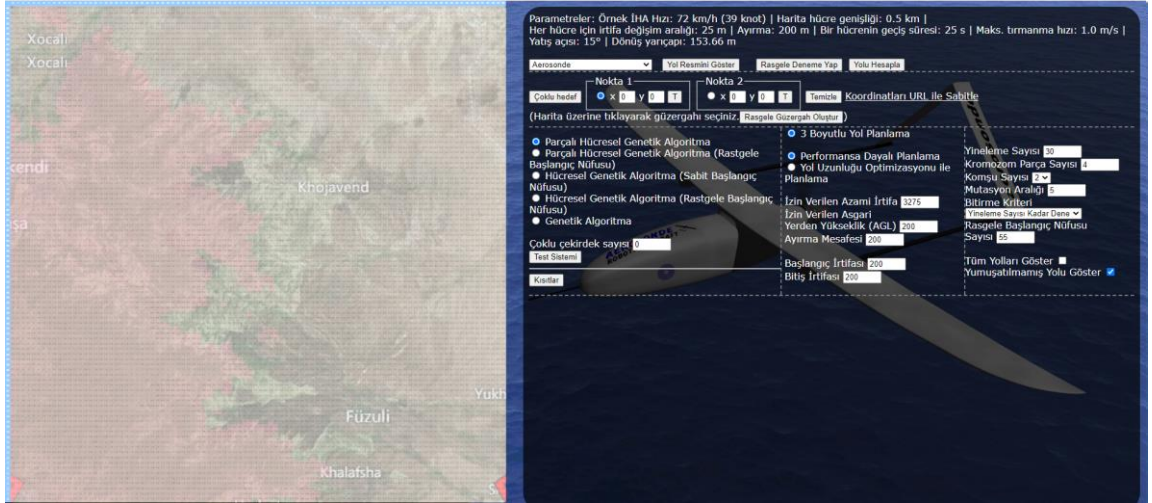
3.9.1. Yazılım paketinin grafik kullanıcı arayüzü

Bir yol planlama algoritmasının çalışabilmesi için engellerin sayısı, boyutları ve konumu, başlangıç noktası, varış noktası, aracın hızı ve kinematik kısıtları gibi çeşitli bilgilere sahip olması gerekmektedir. Öte yandan, bilgilerde bir eksiklik yol planlamayı sekteye uğratmaktadır. Geliştirilen yazılım paketi, yol planlama sürecinin sistematik ve kullanıcı dostu bir şekilde yönetilebilmesi için tasarlanmıştır.

Grafiksel kullanıcı arayüzü (GUI) üç bölüme ayrılmıştır:

- 1) Başlangıç verilerinin girilmesi için bölüm.
- 2) Çözüm algoritmasını seçme bölümü.
- 3) Sonuçlar ile ilgili bilgilerin görüntülediği bölüm.

Yazılımın grafik çıktıları aynı pencerede görüntülenmektedir. Bulunan en iyi yol, bir noktalar dizisi olarak uçuş hız ve irtifasını gösterir bir yörünge şeklinde listelenmektedir. Hesaplama süresi görüntülenmektedir. GUI'nin genel görünümü Görsel 3.2.'de sunulmuştur.

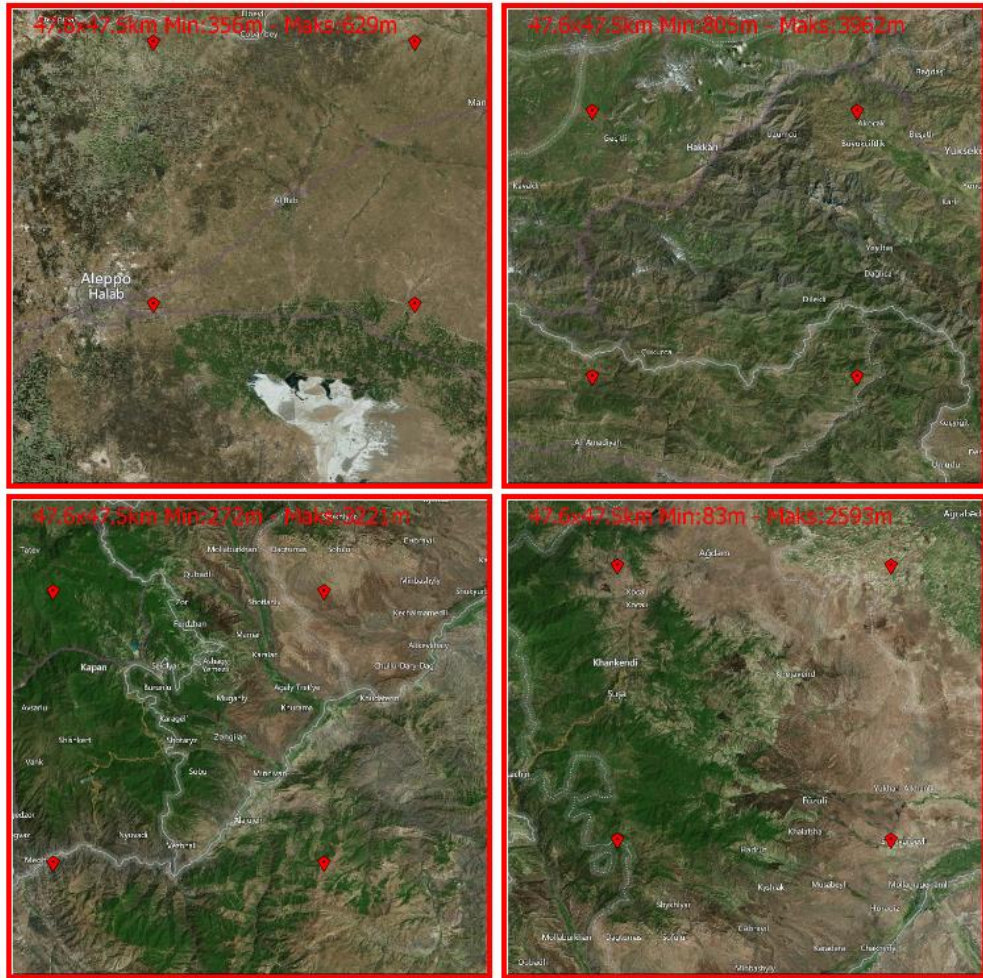


Görsel 3.2. Yol planlama arayüzü

GUI'ye aşağıdaki özellikler dâhil edilmiştir:

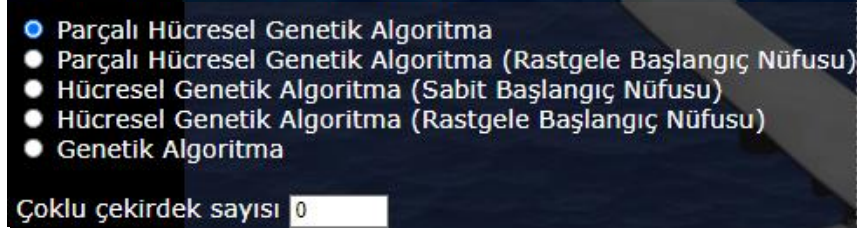
- Düzenlenebilir kutular, basmalı düğmeler vb. tüm nesneler kategorisel olarak düzenlenmiştir.
- Nesneler, çerçeveler içinde gruplandırılmıştır. İlişkili nesnelerin aynı alanda olmaları görülmelerini ve kullanılmalarını kolaylaştırmaktadır.

- İHA'nın hareketi, harita üzerinde çözüm üretildiği anda görsel olarak yer almaktadır.
- İstenildiğinde son popülasyondaki alternatif tüm yollar görüntülenebilir.
- Yol uygunluk değeri ve işlem süresi gibi planlanan yol hakkında bilgiler görüntülenmektedir.
- Simülasyon butonuna tıklandığında FlightGear simülatörü üzerinde planlanan yol üzerinde uçuş gösterilmektedir.
- Kullanıcının yol planlama sonucu oluşturulan yolu görüntülemesini sağlayan bir seçenek bulunmaktadır.
- Haritalar ve seçilen noktalar gelecekte kullanılmak üzere kaydedilebilmektedir.



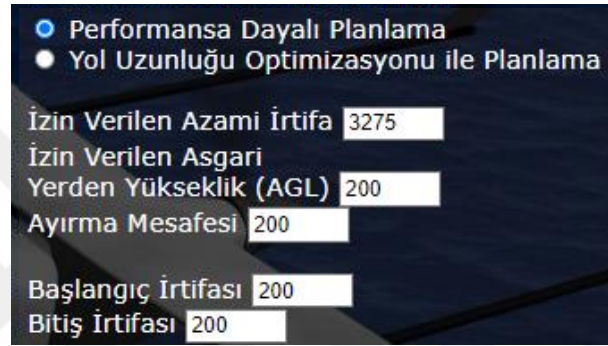
Görsel 3.3. Yol planlama yazılımı harita seçme ekranı

Görsel 3.3'te üzerinde yol planlaması yapılmak istenen uçuş bölgesinin seçimi için oluşturulan harita seçim ekranından bir kesit sunulmuştur. Haritalar istenen koordinat aralığı için oluşturulabilmektedir.



Görsel 3.4. Yol planlama yazılımı algoritma ve çoklu işlem seçenekleri

Yol planlamanın hangi algoritma ile yapılacağını seçildiği bir ekran bulunmaktadır (Bkz. Görsel 3.4). Aynı ekranda çoklu işlem kullanılacaksa süreç sayısı belirtilebilir. “0” girişi yapılan durumlarda tek süreç kullanılmaktadır.



Görsel 3.5. Yol planlama yazılımı algoritma parametreleri

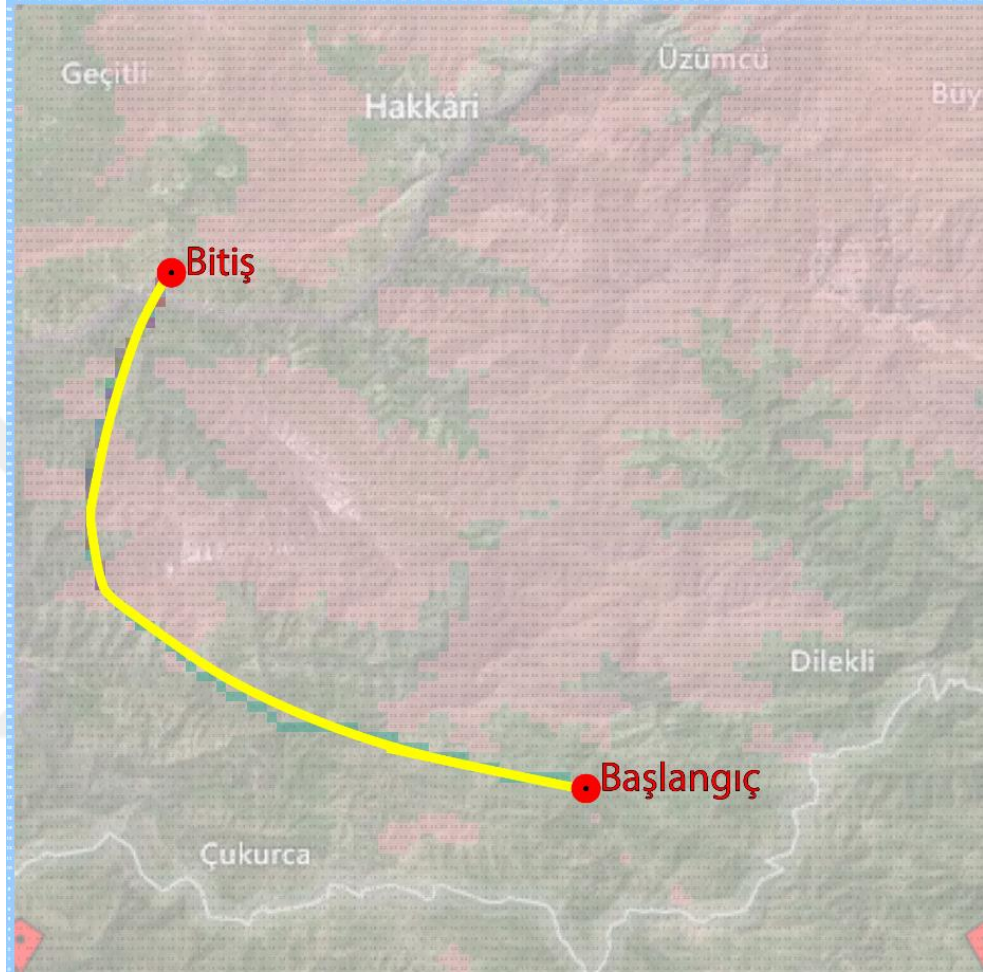
Karşılaştırmaya imkân vermek amacıyla hava aracı performansına dayalı ya da yol uzunluğuna dayalı planlama, GUI üzerinden seçilebilir. İzin verilen azami irtifa, izin verilen asgari yerden yükseklik (AGL), ayırma mesafesi, başlangıç irtifası ve bitiş irtifası bilgileri aynı ekran üzerinde tanımlanabilir. Başlangıç ve bitiş irtifası bilgileri harita üzerine tıklandığında yer seviyesine göre güncellenmektedir (Bkz. Görsel 3.5)



Görsel 3.6. Yol planlama yazılımı algoritma parametreleri

Yineleme sayısı, kromozom parçası sayısı, mahalle içerisindeki komşu sayısı, mutasyon oranı, bitirme kriteri (yineleme sayısı kadar dene ya da geçerli sonucu bulunca dur) ve rastgele başlangıç popülasyon büyüklüğü parametreleri Görsel 3.6’da gösterildiği

gibi tanımlanabilir. Aynı ekran üzerinde, algoritma tamamlandığında bulunan alternatif tüm yolların gösterilmesini sağlayan “tüm yolları göster” seçeneği işaretlenebilir. Yol yumuşatma seçeneği de bu ekranda bulunmaktadır.

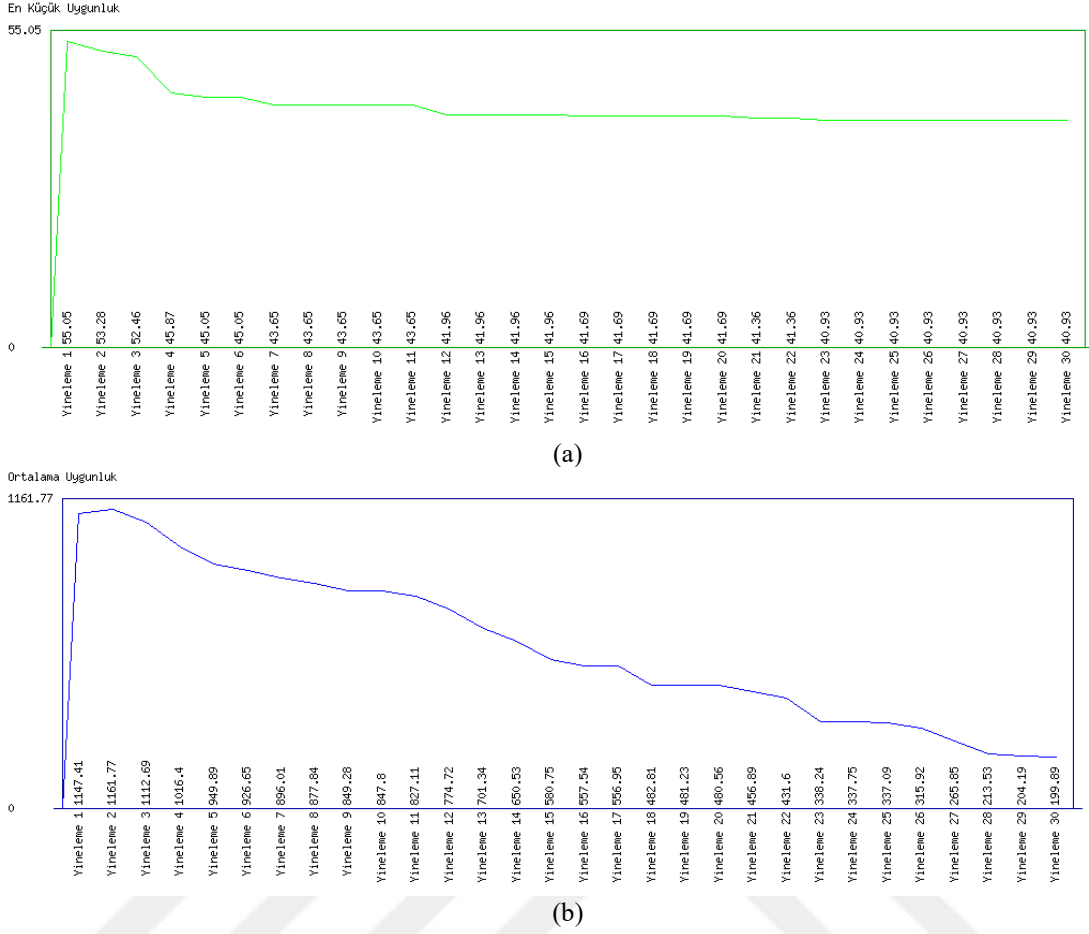


Şekil 3.21. Harita ve üretilen yol

3.9.2. Yazılım paketinin ana işlevleri

Bu bölüm, yazılım paketinin tüm işlevlerini, amaçlarını, giriş ve çıkış parametrelerini açıklamaktadır.

İHA yönetim paneli olarak adlandırılan bölüm, uçuş görevini yapılandırmak için geliştirilmiştir. Bu yazılım aynı zamanda, çalışılan harita üzerinde 2D uçuş yolu görüntüleme işlevi görmektedir. Bununla birlikte, ara noktalar ve engeller gibi görev parametrelerinin belirlenmesine izin verilmektedir. Algoritmalar, farklı haritalar ve seçenekler ile otomatik test edilebilir. Harita örneği ve üretilen çözüm Şekil 3.21'de gösterilmiştir. Asgari ve ortalama uygunluk değer çizelgeleri algoritma tamamlandığı anda hazırlanmaktadır (Bkz. Şekil 3.22).



Şekil 3.22. (a) Asgari uygunluk değeri grafiği (b) Ortalama uygunluk değeri grafiği

Adil bir karşılaştırma yapabilmek için her bir yol planlama algoritmasında aynı parametreler kullanılmaktadır. Yol yumuşatma algoritmasının etkisi uygunluk değerine dâhil edilmez. Tüm algoritmalar aynı bilgisayar üzerinde ve aynı harita senaryolarında test edilmektedir. Yol uygunluk değeri, yol geçerliliği ve hesaplama süresi karşılaştırmada dikkate alınan performans ölçütleridir.

3.10. Simülasyon Ortamı

Bugüne kadar, birçok teorik algoritma, araştırmacılar tarafından geniş çapta tartışılmıştır. Sunulan yöntemlerden bazıları gerçek ortamda gösterilme imkânı bulmuştur. Teorik hesaplamaların pratikte uygulanabilir olduğunun en azından simülasyon ortamında sınanması önemlidir.

İHA simülasyon araçları, gerçeğe yakın durumları modellemek için İHA uçuşlarını sanal ortamda taklit etmektedirler. İHA'ların gerçek senaryolarda kullanımından önce çok daha düşük maliyet ve zahmetle sanal ortamda test etme kolaylığı sağlamaktadırlar. Simülatörün seçimi hem test amacına hem de her simülatör tarafından sunulan

özelliklerin listesine bağlıdır. Bazı simülatörler, İHA'ların doğal hareketlerini simüle etmeyi sağlayan hareket yakalama (MOCAP) özelliğine sahiptir [343].

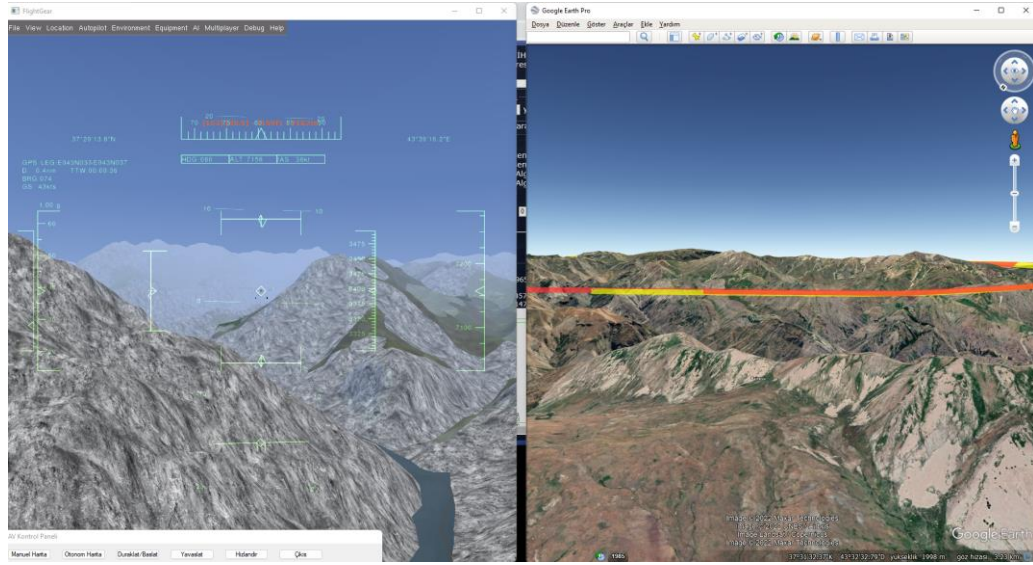
İHA simülasyonu için kullanılan çok sayıda araç bulunmaktadır. Microsoft AirSim [344], FlightGear [345], JMavSim [346], Gazebo [347], X-Plane [348] ve UE4Sim [349] dâhil olmak üzere çok sayıda simülatör literatürde yer almıştır. Hem yapay zekâ araştırmaları hem de eğitim için bir platform olarak kabul edilen simülatörler İHA hareketlerini, fizik yasalarına dayalı bir yörünge oluşturma motoru kullanarak simüle etmektedir [350]. AirSim gibi bazı simülatörlerde derin öğrenme, bilgisayarla görmeye yönelik veri kümeleri oluşturma gibi özellikler kullanılabilir [351]. Simülatörler araştırmacıların, algoritmalarını yapay zekâ desteği ile gerçeğe yakın bir ortamda simüle etmelerine ve geliştirmelerine olanak tanımaktadır.

Açık kaynaklı bir uçuş simülasyon uygulaması olan FlightGear yaygın bir simülasyon ortamıdır. FlightGear; mühendislik, araştırma ve akademik amaçlı, aynı zamanda pilot eğitiminde kullanılabilen sofistike ve açık kaynaklı bir uçuş simülatörüdür [345, 352]. Birçok akademik çalışmada kullanılmıştır [353-356]. FlightGear aracılığıyla uçağın durumu görselleştirilebilir. MATLAB Simulink üzerinden kontrol arayüzüne sahiptir [357].

Tez çalışması kapsamında algoritmaların ürettiği yollar, geçerliliği kanıtlanmak için FlightGear simülasyon programına aktarılmış ve GoogleEarth üzerinde eş zamanlı olarak izlenebilmesi sağlanmıştır.

Yol planlama sistemi ile FlightGear programı arasında bağlantı ve senkronizasyon telnet protokolü üzerinden sağlanmıştır. Uçuş için İHA; yolun başlangıç noktasında başlangıç koordinatlarında, hesaplanan baş açısı istikametinde ve tanımlanan hızda havada başlangıç yapmakta ve belirlenen yolu takip ederek varış noktasına ulaşmaktadır.

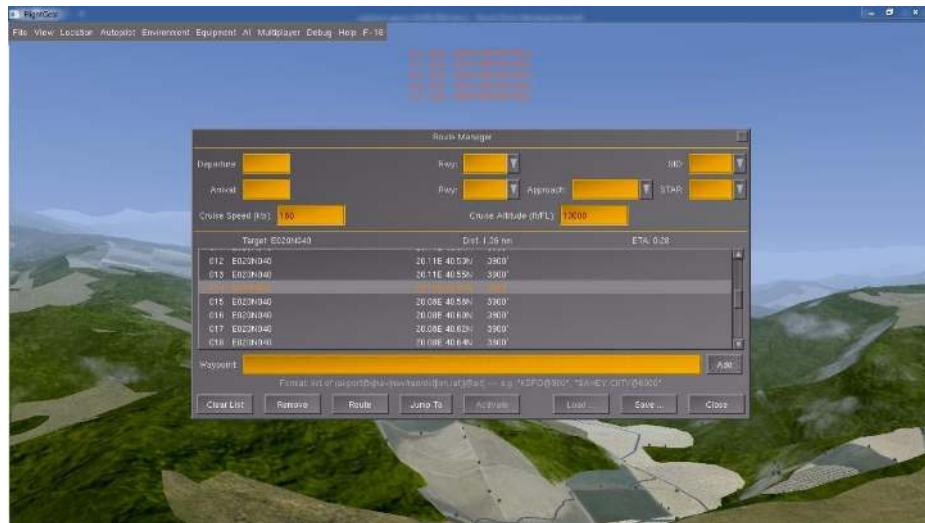
Koordinatlar ve irtifa değerlerinin Google haritalarına aktarıldığı ve izlendiği çalışmalar bulunmaktadır [358]. Bu çalışmada da hazırlanan yol ve manevralar koordinat tabanlı yörüngeler olarak iki uçuş izi şeklinde GoogleEarth üzerinde görüntülenmektedir. Görsel 3.7 ve 3.8'de FlightGear, planlanan yol ve İHA hareket izinin görüntülediği GoogleEarth ekran görüntüsüne örnek verilmiştir.



Görsel 3.7. *FlightGear ve GoogleEarth eşzamanlı görüntüsü*



Görsel 3.8. *Planlanan yolun (kırmızı) ve İHA hareket izinin (sarı) görüntülediği GoogleEarth ekran görüntüsü*



Görsel 3.9. *FlightGear Route Manager*

Yollara ait noktalar “waypoint” olarak “routemanager”e (Bkz. Görsel 3.9) telnet üzerinden yüklenir ve otomatik pilot bu “waypoint”ler üzerinden çalışmaktadır. Otomatik pilotun kullandığı hız bilgileri telnet üzerinden kontrol edilmiştir.

Yol planlaması ile üretilen yolların uygulanabilir olduğunun gösterilmesi önemlidir. Bu kapsamda İHA, oluşturulan yollar üzerinde FlightGear kullanılarak simüle edilmiştir. Yapılan simülasyon denemeleri, İHA'nın çarpışma olmadan, önceden tanımlanmış yörüngeleri takip edebildiğini doğrulamaktadır.



4. ARAŞTIRMA VE BULGULAR

Bu bölümde, tez çalışması kapsamında önerilen algoritmalar ve bunların varyantlarının karşılaştırmalı bir analizinin sunulması amaçlanmaktadır. cGA ve scGA algoritmalarının, 3B İHA yol planlamasında uygulanabilirliği değerlendirilmektedir. Bulgular, algoritmaların kullanılabilirliğini ana hatlarıyla belirtirken en iyi algoritma ve en iyi parametre konfigürasyonunun seçimine yardımcı olabilir. Tüm algoritmalar aynı senaryolarda test edilmekte ve aynı performans ölçütleri kullanılarak analiz edilmektedir. Önerilen algoritmalarda, Bölüm 3.3.7'de belirtilen tüm parametreler uygulanarak ayrı ayrı analiz yapılmaktadır. Böylece, yol uygunluğu, hesaplama süresi ve geçerli yollar açısından algoritmaların başarısı bağımsız olarak değerlendirilmektedir. Değişkenlerin etkisi vurgulanmaktadır.

3B yol planlamasının uygulanması için en iyi konfigürasyonların belirlenmesine çalışılmıştır. Bu bölümde algoritmaların ve varyantlarının yol planlama performansı analiz edilmekte, tartışılmakta ve karşılaştırılmaktadır. 3B yol planlamasında tez kapsamında değerlendirilen her bir algoritmanın güçlü ve zayıf yönlerinin belirlenmesine çalışılmaktadır.

Önerilen scGA, cGA ve varyantlarının etkinliği standart bir GA ile birlikte karşılaştırılmıştır. Algoritmaları karşılaştırmak için 30 yinelemede durdurma koşulu kullanılmıştır. Popülasyon büyüklüğü 55 olarak sabit tutulmuştur. cGA ve scGA için ikişer varyant oluşturulmuştur. Birinci tip varyantta 41 sabit ile 14 rastgele başlangıç popülasyonu kullanılmaktadır (cGA ve scGA). İkinci tip varyant tamamen rastgele başlangıç popülasyonu kullanılmaktadır (cGAR ve scGAR). cGA(R) ve scGA(R) için farklı mahalle yapıları ile testler gerçekleştirilmiştir.

Mahalle türü, analiz için dikkate alınan değişkenlerden biridir. Her testte tanımlanan diğer tüm parametrelerin, standart değerlerinde olduğu durumda mahalle yapısının etkisi incelenmiştir. Tüm senaryolarda test sonuçlarının karşılaştırılmasını ve geçerli yol oluşturulmadığı durumların dağılımını gösteren tablo ve grafikler sunulmaktadır.

Bu araştırmayı netleştirebilmek için mümkün olan en iyi komşuluğun uzunluğu belirlenmeye çalışılmıştır. cGA'da yapılandırma ızgarası (popülasyon büyüklüğü ile ilişkili olarak) 55x1 birey boyutundadır. scGA'daki yerleşim kromozomlara göre

olduğundan 55x3, 55x4 gibi ızgara yapılandırmaları kromozom uzunluğuna göre otomatik belirlenmektedir. Izgara yapılandırmasındaki satır sayısı, ortadaki (sabit başlangıç popülasyonunda en düz yola sahip) kromozoma göre; düğüm sayısı sırasıyla 4'ten az ise 2, 12'den az ise 3, 16'dan az ise 4 olarak; 16'dan büyük düğüm sayısında, düğüm sayısının 0.2 çarpımından elde edilen tam sayı şeklinde tanımlanmaktadır. Belirlenen ızgara yapılandırması üzerinde ayrı ayrı 2, 3, 5, 6, 7, 8 ve 9 bireylik mahalle uzunluklarında testler gerçekleştirilmiştir.

Tablolarda her bir algoritma için 30 yinleme üzerinden ortalama sonuçlar özetlenmiştir. Geçersiz sonuçların yüzdesi, çeşitlilik oranı ve ortalama değerler olarak; ortalama uygunluk değeri, ortalama hesaplama süresi, ortalama geçerli sonuç bulma süresi, ortalama geçerli sonuç bulma yinleme sayısı, ortalama en iyi geçerli sonuç bulma süresi, ortalama en kötü geçerli sonuç bulma süresi ve son popülasyon ortalama uyum değeri sütunları gösterilmiştir. Yol uzunluğu optimizasyonunu esas alan algoritmalara ait testlerde yol uzunluğu optimizasyonunu esas alan uygunluk değerinin yanında, İHA performansına dayalı olarak hesaplanması durumunda uygunluk değerinin ne olacağı da bir sütun şeklinde sunulmuştur.

Tüm deneyler, Windows 10 Pro, 64 bit işletim sisteminde, 3,90 GHz hızında çalışan AMD Ryzen 5 2400G işlemci ve 16 GB RAM ile yapılandırılmış bir PC üzerinde test edilmiştir (Bkz. Tablo 4.1). Test sürecinde işletim sistemi ve diğer programlarla ilişkili arka plan görevleri az da olsa test sonuçlarına etki edebilmektedir.

PHP 5.6.39 dili, MySQL 10.1.37-MariaDB veritabanı ile Apache 2.4 web sunucusu üzerinde çalışan bir web uygulaması olarak programlanan algoritmaların her biri için bağımsız çalışma gerçekleştirilmiştir. Toplam 19800 deneye tekabül eden 99 küme ve 4 farklı harita kullanılmıştır. Kategorize edilen tüm testlerde rastgele 10 farklı rotada, her rotada 5'er defa olmak üzere toplam 50 test gerçekleştirilmiştir. Geçersiz yollar ortalamalara dâhil edilmez. Ancak rotadaki 5 deneyin tamamı geçersiz ise uygunluk değeri olarak bir ceza puanı değeri, çözüm sürelerinde hesaplama süresi değeri kullanılmaktadır. Bu nedenle, geçerli sonuç üretemeyen bir algoritmanın hız açısından daha başarılı görünme algısına karşın analizler, geçerli sonuç üretebilme perspektifinden değerlendirilmelidir.

Test sonuçları değerlendirilirken, PHP'nin derlenerek çalışan bir programlama dilinden bir derece daha yavaş çalışan betik bir dil olduğu dikkate alınmalıdır. Bununla birlikte; cGA, scGA ve geleneksel GA'nın sonuçlarını karşılaştırmak için kullanılan dilin seçimi bir dezavantaj değildir. Her yol planlama algoritmasını test etmek için aynı işlemci ve yazılım dili kullanılmakta, bu da algoritmalar arasında adil bir karşılaştırma yapılabileceği anlamına gelmektedir.

Tablo 4.1. *Test platformu özellikleri*

Özellikler	Değerler
İşlemci	AMD Ryzen 5 2400G with Radeon Vega Graphics, 3900 Mhz 4 Çekirdek, 8 Mantıksal İşlemci
Anakart	Gigabyte B450M S2H
Bellek	16 GB DDR4-3200 / PC4-25600 DDR4
Disk	Samsung SSD 960 EVO 250GB
İşletim Sistemi	Microsoft Windows 10 Pro, 64 bit

Algoritmaların parametre farklılıklarının performans üzerindeki etkileri, yol ekonomisi, planlama süresi ve algoritma performansı açısından değerlendirilmektedir. Bu nedenle, bu tez çalışması çerçevesinde, algoritmalarındaki çeşitli değişkenlerin yol planlama performansı üzerindeki etkisini görmek için İHA parametreleri, bağımsız olarak incelenmektedir. Algoritma isimlerine eklenen L2, L3 vb. ekler mahalle boyutunu; “Ç” ekli ise çoklu işlem kullanıldığını göstermektedir.

4.1. Yol Uzunluğu Optimizasyona Dayalı Yol Planlama

Tez çalışmasının ilk aşamasında, hava aracı performansına dayalı yaklaşım geliştirilmeden önce 2D ve 3D ortamda yol uzunluğu optimizasyona dayalı yol planlama çalışılmıştır [306]. Testler tek süreç üzerinde farklı bir bilgisayar konfigürasyonunda gerçekleştirilmiştir.

Tablo 4.2. *2D ortamda L3 mahalle yapısında yol uzunluğu optimizasyonunu esas alan algoritmaların tek süreçli test sonuçları*

Algoritma	Geçersiz sonuç yüzdesi	Uygunluk değeri	Geçerli sonuç bulma süresi (s)	Hesaplama süresi (s)
cGA	%1,28	32,97	1,28	13,13
cGAR	%2,87	34,62	3,08	20,94
GA	%7,23	36,87	2,43	11,65
scGA	%0,96	34,14	0,26	5,92
scGAR	%0,32	38,26	1,67	15,63

2D yol planlamada, scGA neredeyse her başlıkta en iyi sonuçlara ulaşmıştır (Bkz. Tablo 4.2). scGA, GA'dan iki kat daha hızlı sonuç üretebilmiştir. İlk geçerli çözümde hız farkının 11 kat daha iyi olduğu not edilmiştir. 3D yol planlamada algoritmanın başarısı 2D'ye kıyasla azalsa da scGA simülasyonu 1,6 kat daha hızlı tamamlamış ve geçerli yolları 2,4 kat daha hızlı üretebilmiştir (Bkz. Tablo 4.3) [306]. cGA, ortalama uygunluk değerinin ve asgari uygunluk değerinin, scGA'dan küçük bir fark ile daha iyi sonuçlar elde etmiştir. Bu durumun, kromozom parçalarının sabit olarak ilk ve son noktalarından çaprazlanmasından kaynaklandığı değerlendirilmektedir. Algoritmadaki bu dezavantaj sonraki çalışmalarda düzeltilmiştir.

Tablo 4.3. 3D ortamda L3 mahalle yapısında yol uzunluğu optimizasyonunu esas alan algoritmaların tek süreçli test sonuçları

Algoritma	Geçersiz sonuç yüzdesi	Uygunluk değeri	Geçerli sonuç bulma süresi (s)	Hesaplama süresi (s)
cGA	%2,25	33,49	3,16	32,58
cGAR	%1,61	33,41	8,41	57,36
GA	%2,58	35,11	5,14	33,82
scGA	%0,00	33,61	1,68	21,64
scGAR	%0,00	34,28	3,71	41,16

Tez çalışmasının ilk aşamasında kullanılan algoritma çeşitli yönlerden iyileştirilmiş ve hava aracı performansına dayalı bir planlamaya uygun hâle getirilmiştir. Yukarıdaki testlerde elde edilen sonuçlar ile aşağıda açıklanacak test sonuçları arasında farklılıkların nedeni yazılımın büyük oranda yeniden yazılmış olmasıdır. Örnek olarak önceki çalışmada [306], scGA(R)'de kromozom parçaları, dizi değişkenin farklı boyutlarında yer alırken algoritmada yapılan değişiklikler ile, tek boyutlu dizide ve parçalar ile ilgili uygunluk değerlerini saklayan ikinci bir dizinin yardımıyla kullanılmıştır.

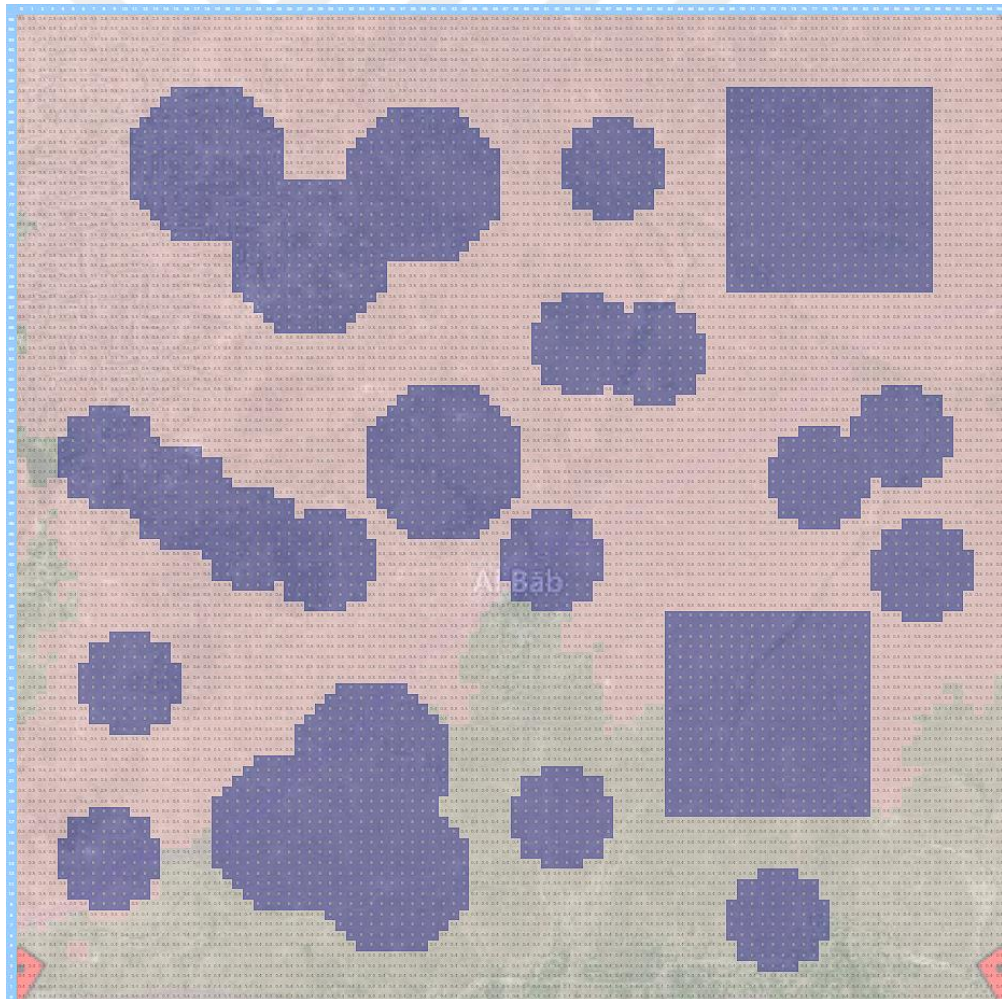
4.2. Hava Aracı Performansına Dayalı Yol Planlama

Yol uzunluğu optimizasyonunu esas alan algoritmalar ile hava aracı performansına dayalı algoritmaların karşılaştırılabilmesi için aynı konfigürasyonda testler gerçekleştirilmiştir. Testlere tabi tutulan standart GA algoritması da önerilen algoritmalar gibi aynı yol uzunluğu optimizasyonunu esas alan ya da hava aracı performansına dayalı amaç fonksiyonu kullanmaktadır. Testler tek süreçli ve çok süreçli olarak iki şekilde uygulanmıştır. Testlerde dört farklı harita kullanılmıştır. Haritalar, 47,5 m²

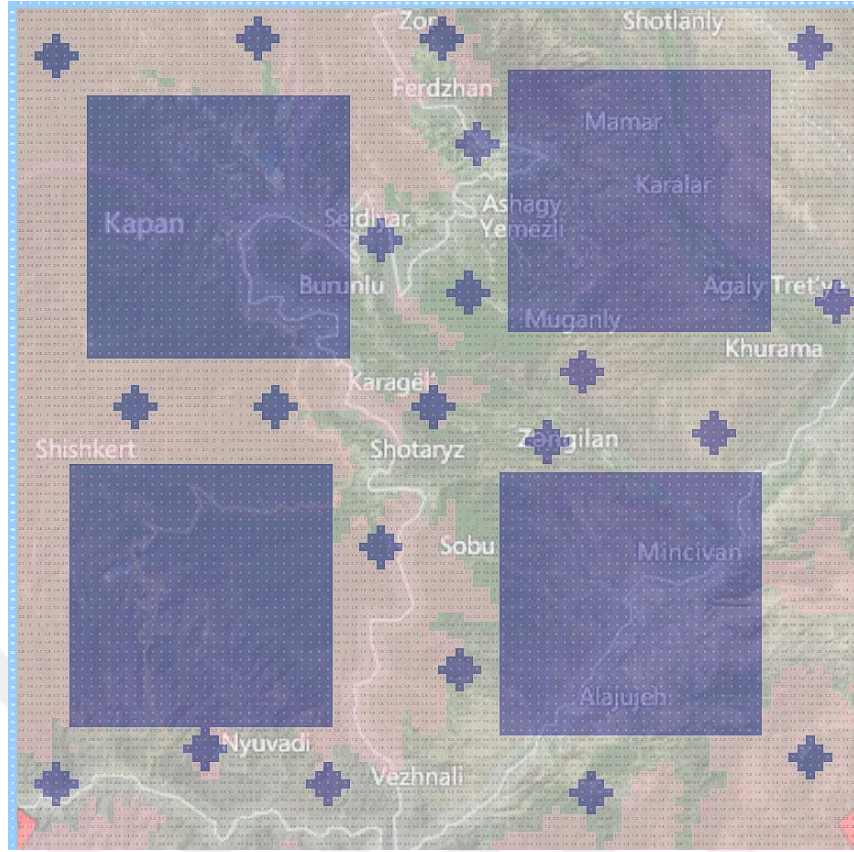
genişliğindedir. Görsel 4.1’de gösterilen birinci harita, daha yumuşak yeryüzü şekillerine sahiptir ancak çeşitli engeller tanımlanmıştır. Görsel 4.2’de gösterilen ikinci harita, engeller içeren engebeli ve yükseltili bir arazi yapısındadır. Görsel 4.3 ve Görsel 4.4’te verilen üçüncü ve dördüncü haritalar, engel içermeyen farklı yeryüzü şekillerinde yükseltili arazi yapısına sahiptir (Bkz. Tablo 4.4).

Tablo 4.4. Haritalar

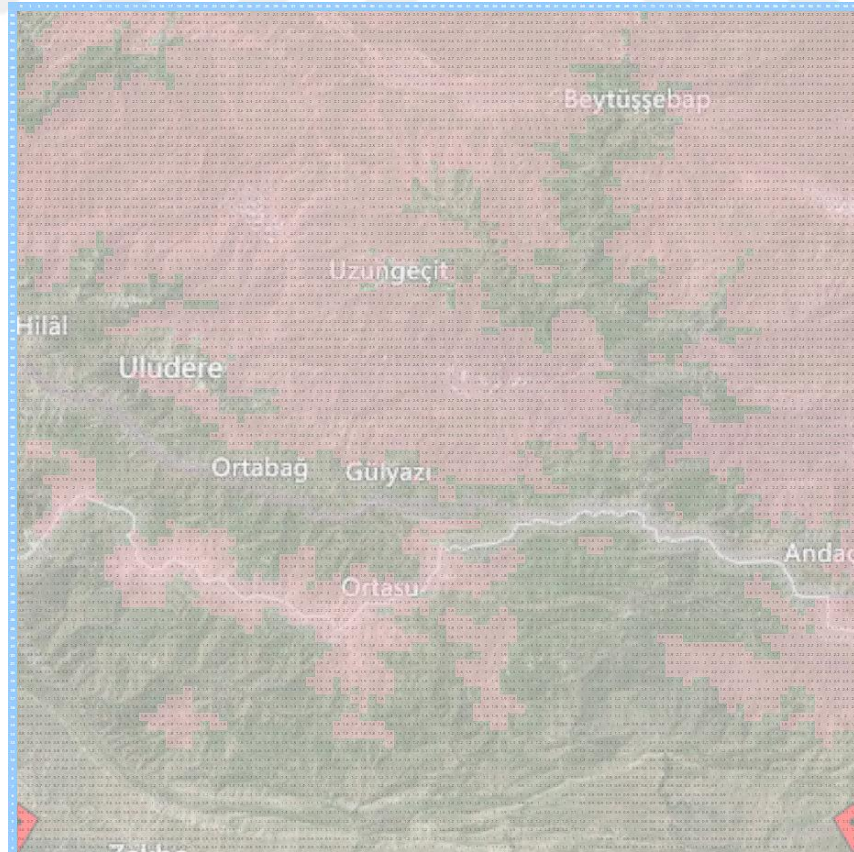
Harita	Koordinat aralığı	Asgari yükseklik	Azami yükseklik	Engel durumu
Harita-1	36.18511383-37.26516926, 36.61304176-37.79441901	356	629	Var
Harita-2	38.87373875-46.28945710, 39.30166668-46.83813435	272	3221	Var
Harita-3	37.19608466-42.75250172, 37.62401259-43.28875772	555	3286	Yok
Harita-4	37.19324349-43.51862462, 37.62117142-44.05485687	805	3962	Yok



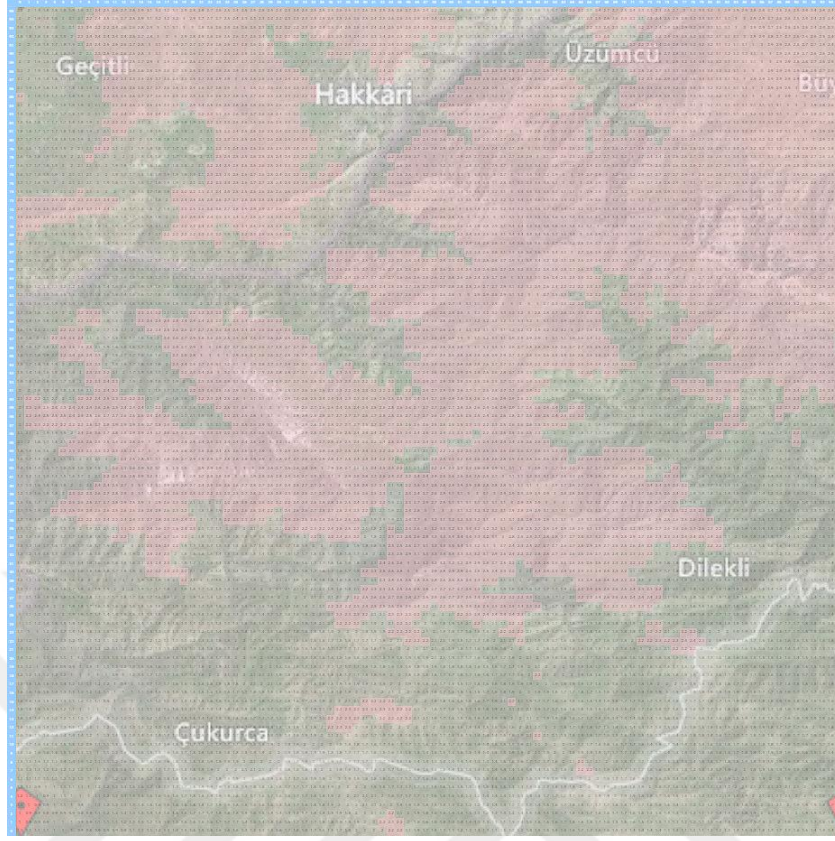
Görsel 4.1. Harita-1



Görsel 4.2. Harita-2



Görsel 4.3. Harita-3



Görsel 4.4. Harita-4

Yol uzunluğu optimizasyonunu esas alan algoritmalar karşılaştırıldığında Harita-1’de, (Bkz. Tablo 4.5) scGA-L4 algoritması en iyi, cGAR-L5 algoritması ise en kötü sonuçları elde etmiştir.

Tablo 4.5. Yol uzunluğu optimizasyonunu esas alan algoritmaların Harita-1 üzerindeki test sonuçları

Algoritma	Ort. yol uzunluğuna dayalı uygunluk değeri	Ort. performansa dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yineleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
cGA-L2	38,47	49,89	3,29	0,81	3,18	0,24	3,19	%18	3,859	827
cGA-L3	134,14	145,21	3,11	0,71	1,16	0,21	3,08	%26	2,785	789
cGA-L4	225,03	232,93	3,16	1,32	1,27	0,23	6,13	%24	2,75	712
cGA-L5	36,71	46,44	3,15	1,02	4,15	0,22	5,12	%18	3,744	695
cGA-L6	36,56	46,26	3,23	0,97	3,73	0,23	4,35	%20	3,306	661
cGA-L7	131,99	139,79	3,22	1,12	2,86	0,22	3,89	%26	2,559	662
cGA-L8	37,94	49,1	3,24	0,99	4,08	0,22	3,93	%16	3,759	629
cGA-L9	131,07	138,88	3,27	0,99	2,43	0,23	3,19	%22	2,563	629

Tablo 4.5. (Devam) *Yol uzunluğu optimizasyonunu esas alan algoritmaların Harita-1 üzerindeki test sonuçları*

Algoritma	Ort. yol uzunluğuna dayalı uygunluk değeri	Ort. performans dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yineleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
cGAR-L2	131,57	139,56	3,1	1,56	5,43	0,43	5,05	%34	4,879	936
cGAR-L3	226,25	232,51	2,94	1,51	2,63	0,36	4,7	%38	2,83	869
cGAR-L4	320,88	327,34	2,99	1,77	1,54	0,47	5,75	%44	2,987	861
cGAR-L5	323,08	328,13	2,96	1,79	3,64	0,4	4,99	%46	2,146	829
cGAR-L6	225,59	231,57	3,03	1,53	2,31	0,43	4,75	%38	2,372	788
cGAR-L7	131,89	139,91	3,01	1,53	5,79	0,4	4,44	%40	3,221	741
cGAR-L8	226,19	232,32	3,03	1,64	3,62	0,46	4,15	%42	2,223	745
cGAR-L9	226,99	233,48	3,06	1,47	3,68	0,43	5	%40	2,177	741
GA	131,72	139,83	3,39	1,55	5,06	0,45	4,43	%34	2,396	920
scGA-L2	34,39	43,43	5,09	0,69	1,38	0,33	1,78	%0	4,845	375
scGA-L3	34	42,9	4,91	0,73	1,58	0,31	2,38	%0	5,601	276
scGA-L4	33,89	42,81	4,91	0,63	1,24	0,3	1,28	%0	6,891	216
scGA-L5	34,23	43,16	4,94	0,74	1,56	0,31	2,37	%0	8,259	171
scGA-L6	34,29	43,29	5,01	0,71	1,5	0,3	1,86	%0	8,74	168
scGA-L7	34,23	43,27	5,07	0,72	1,46	0,33	2,2	%0	10,242	144
scGA-L8	33,99	42,92	5,04	0,7	1,44	0,3	2,06	%0	8,918	165
scGA-L9	33,94	42,84	5,06	0,81	1,78	0,28	2,92	%0	10,58	144
scGAR-L2	34,21	43,25	5,13	1,45	3,56	0,61	2,83	%0	4,241	404
scGAR-L3	34,15	43,11	4,78	1,29	3,12	0,59	2,71	%0	6,472	224
scGAR-L4	33,97	42,9	4,77	1,32	3,12	0,58	2,88	%0	6,443	225
scGAR-L5	33,91	42,88	4,85	1,26	2,78	0,57	2,6	%0	8,193	173
scGAR-L6	34,02	42,96	4,96	1,27	2,74	0,61	2,53	%0	8,442	165
scGAR-L7	34,11	43,07	4,88	1,18	2,46	0,63	2,54	%0	9,625	136
scGAR-L8	34	42,92	4,95	1,25	2,7	0,57	2,6	%0	9,036	156
scGAR-L9	33,94	42,86	4,96	1,4	3,38	0,6	3,22	%0	8,625	154

Yol uzunluğu optimizasyonunu esas alan algoritmalar karşılaştırıldığında Harita-2’de, (Bkz. Tablo 4.6) scGAR-L6 algoritması en iyi, cGAR-L2 algoritması ise en kötü sonuçları elde etmiştir.

Tablo 4.6. *Yol uzunluğu optimizasyonunu esas alan algoritmaların Harita-2 üzerindeki test sonuçları*

Algoritma	Ort. yol uzunluğuna dayalı uygunluk değeri	Ort. performans dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yineleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
cGA-L2	19,55	221,63	1,53	0,28	1,66	0,08	0,61	%2	13,823	500
cGA-L3	19,39	221,43	1,47	0,26	1,68	0,09	0,52	%4	11,342	469
cGA-L4	19,37	221,34	1,49	0,27	1,58	0,08	0,65	%4	12,145	400
cGA-L5	19,35	221,34	1,53	0,26	1,64	0,08	0,64	%0	13,28	343
cGA-L6	19,39	221,34	1,54	0,27	1,6	0,08	0,66	%6	12,912	337
cGA-L7	19,35	221,33	1,54	0,3	2,44	0,07	0,76	%2	14,653	313
cGA-L8	19,29	221,23	1,53	0,25	1,58	0,08	0,53	%2	14,635	286
cGA-L9	19,3	221,22	1,57	0,25	1,44	0,08	0,45	%0	15,948	251
cGAR-L2	19,71	221,87	1,65	0,45	1,78	0,13	1,08	%2	15,636	510
cGAR-L3	19,48	221,51	1,59	0,4	1,44	0,14	0,86	%2	13,858	469
cGAR-L4	19,34	221,48	1,63	0,37	1,17	0,13	0,62	%4	13,297	390
cGAR-L5	19,32	221,39	1,63	0,39	1,26	0,13	0,78	%4	13,423	369
cGAR-L6	19,3	221,31	1,71	0,4	1,3	0,14	0,87	%4	15,104	305
cGAR-L7	19,25	221,26	1,65	0,43	2,2	0,14	1,01	%4	14,139	299
cGAR-L8	19,35	221,35	1,65	0,45	2,17	0,14	1,21	%8	14,446	301
cGAR-L9	19,29	221,25	1,64	0,4	1,66	0,13	0,83	%8	14,539	273
GA	19,31	123,21	1,9	0,47	2,32	0,15	1,48	%0	9,926	495
scGA-L2	19,36	221,37	2,42	0,35	1,2	0,1	0,72	%0	32,405	159
scGA-L3	19,29	221,25	2,24	0,32	1,06	0,1	0,66	%0	44,319	102
scGA-L4	19,26	221,23	2,24	0,34	1,14	0,11	0,69	%0	41,268	106
scGA-L5	19,3	221,29	2,22	0,32	1,08	0,1	0,66	%0	51,61	86
scGA-L6	19,23	221,25	2,24	0,33	1,02	0,1	0,65	%0	58,064	75
scGA-L7	19,25	221,24	2,25	0,34	1,16	0,11	0,69	%0	55,243	77
scGA-L8	19,25	221,27	2,24	0,34	1,1	0,1	0,69	%0	51,273	81
scGA-L9	19,25	221,21	2,21	0,33	1,1	0,1	0,65	%0	54,473	76
scGAR-L2	19,29	221,34	2,82	0,62	1,3	0,23	1,34	%0	36,282	127
scGAR-L3	19,23	221,24	2,65	0,58	1,14	0,21	0,99	%0	44,989	96
scGAR-L4	19,23	221,25	2,62	0,57	1,14	0,2	1,04	%0	45,161	90
scGAR-L5	19,22	221,2	2,61	0,58	1,2	0,22	1,01	%0	44,546	88
scGAR-L6	19,2	221,19	2,52	0,56	1,08	0,21	0,96	%0	53,648	78
scGAR-L7	19,25	221,28	2,64	0,56	1,1	0,19	0,97	%0	43,01	87
scGAR-L8	19,29	221,26	2,59	0,58	1,22	0,23	1,08	%0	52,707	81
scGAR-L9	19,25	221,23	2,55	0,55	1,2	0,2	0,95	%0	51,999	75

Yol uzunluğu optimizasyonunu esas alan algoritmalar karşılaştırıldığında Harita-3'te, (Bkz. Tablo 4.7) scGAR-L6 algoritması en iyi, cGAR-L2 algoritması ise en kötü sonuçları elde etmiştir.

Tablo 4.7. *Yol uzunluğu optimizasyonunu esas alan algoritmaların Harita-3 üzerindeki test sonuçları*

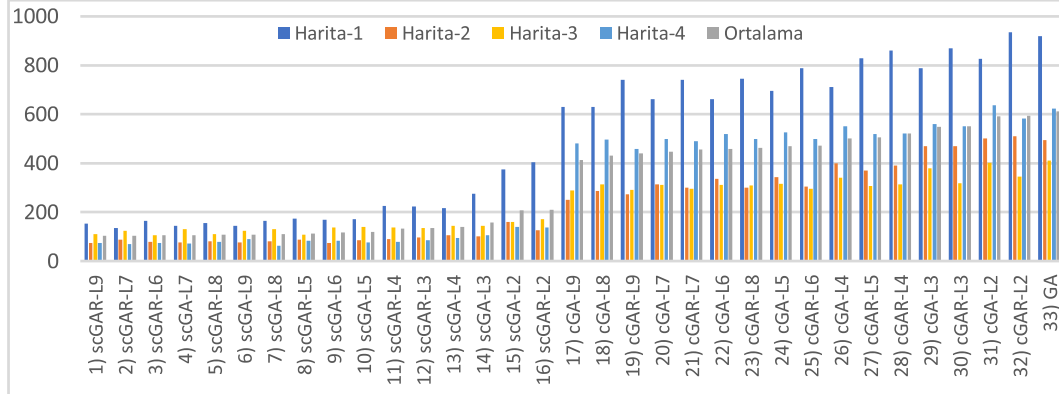
Algoritma	Ort. yol uzunluğuna dayalı uygunluk değeri	Ort. performans dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yinleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
cGA-L2	30,36	612,24	2,96	0,47	1,26	0,08	1,29	%0	12,625	401
cGA-L3	30,43	705,27	2,83	0,45	1,18	0,08	1,44	%2	11,029	378
cGA-L4	30,1	705,27	2,91	0,45	1,18	0,08	1,23	%0	12,027	340
cGA-L5	30,54	612,27	2,95	0,46	1,3	0,09	1,23	%2	12,134	316
cGA-L6	29,82	611,85	2,98	0,46	1,18	0,09	1,73	%0	12,277	312
cGA-L7	30,21	705,26	3,01	0,43	1,04	0,1	1,32	%2	11,643	311
cGA-L8	30,04	705,25	2,99	0,51	1,35	0,1	1,83	%4	12,221	313
cGA-L9	30,02	705,25	3,05	0,43	1,12	0,09	1,31	%0	12,877	288
cGAR-L2	30,93	612,33	2,95	0,58	1,13	0,18	1,18	%2	12,117	346
cGAR-L3	29,79	705,17	2,79	0,53	1	0,19	1,03	%4	11,933	318
cGAR-L4	30,15	705,23	2,89	0,56	1,05	0,2	1,05	%4	11,695	314
cGAR-L5	30,73	612	2,84	0,53	1	0,2	1,11	%4	11,824	306
cGAR-L6	30,05	612,17	2,92	0,62	1,4	0,19	1,89	%2	12,103	296
cGAR-L7	29,47	705,18	2,94	0,53	1	0,2	1,05	%6	12,05	296
cGAR-L8	30,92	612,3	2,91	0,65	1,45	0,19	1,74	%4	11,593	308
cGAR-L9	29,57	705,17	2,98	0,56	1,03	0,19	1,19	%6	11,539	290
GA	30,07	612,25	3,21	0,63	1,41	0,2	1,42	%4	10,261	410
scGA-L2	29,6	612,37	3,88	0,41	1	0,08	1,05	%0	34,523	159
scGA-L3	29,49	611,65	3,64	0,42	1,02	0,09	1,09	%0	34,03	144
scGA-L4	29,53	705,36	3,67	0,43	1,04	0,08	1,13	%0	33,508	144
scGA-L5	29,5	612,36	3,66	0,42	1	0,08	1,17	%0	35,648	139
scGA-L6	29,46	705,29	3,67	0,42	1	0,09	1,17	%0	32,116	138
scGA-L7	29,43	614,92	3,68	0,42	1	0,09	1,16	%0	36,477	131
scGA-L8	29,54	705,25	3,69	0,42	1,02	0,09	1,13	%0	32,771	131
scGA-L9	29,57	705,28	3,69	0,43	1	0,1	1,16	%0	34,078	124
scGAR-L2	29,72	612,04	4,3	0,63	1,08	0,19	1,42	%0	23,436	170
scGAR-L3	29,52	612,22	3,94	0,61	1,08	0,18	1,35	%0	28,361	134
scGAR-L4	29,53	705,26	4,02	0,6	1,06	0,18	1,36	%0	27,136	137
scGAR-L5	29,4	705,2	4,01	0,6	1,02	0,2	1,31	%0	33,339	108
scGAR-L6	29,37	705,2	4,11	0,62	1,04	0,2	1,41	%0	34,068	106
scGAR-L7	29,48	611,65	4,08	0,61	1,02	0,19	1,25	%0	28,979	123
scGAR-L8	29,5	611,94	4,11	0,63	1,08	0,18	1,48	%0	31,394	111
scGAR-L9	29,61	521,54	4,07	0,59	1	0,19	1,19	%0	32,701	110

Yol uzunluğu optimizasyonunu esas alan algoritmalar karşılaştırıldığında Harita-4'te, (Bkz. Tablo 4.8) scGA-L9 algoritması en iyi, cGAR-L7 algoritması ise en kötü sonuçları elde etmiştir.

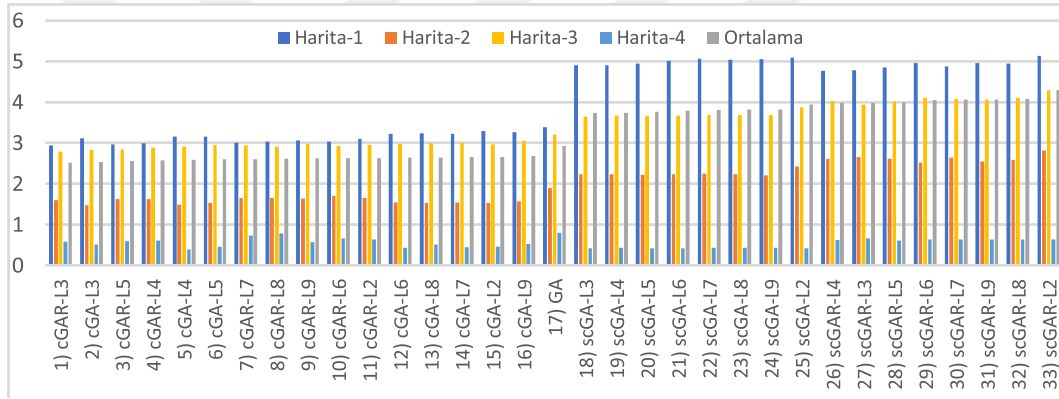
Tablo 4.8. *Yol uzunluğu optimizasyonunu esas alan algoritmaların Harita-4 üzerindeki test sonuçları*

Algoritma	Ort. yol uzunluğuna dayalı uygunluk değeri	Ort. performans dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yinleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
cGA-L2	34,69	1000	2,85	0,46	1,5	0,1	0,96	%4	7,55	637
cGA-L3	34,12	1000	2,73	0,51	1,91	0,1	1,51	%6	6,901	560
cGA-L4	34,55	1000	2,78	0,39	1,05	0,1	0,81	%10	6,257	550
cGA-L5	34,37	1000	2,77	0,46	1,59	0,1	1,28	%6	7,01	527
cGA-L6	34,69	1000	2,81	0,43	1,37	0,09	1,09	%4	6,457	520
cGA-L7	33,81	1000	2,86	0,44	1,29	0,11	1,07	%6	6,195	498
cGA-L8	34,2	1000	2,82	0,51	1,78	0,1	1,97	%6	5,876	496
cGA-L9	34,47	1000	2,85	0,52	1,87	0,1	1,88	%10	5,931	481
cGAR-L2	35,06	1000	2,83	0,64	1,97	0,22	1,22	%14	7,472	582
cGAR-L3	34,56	1000	2,76	0,58	1,3	0,17	1,21	%8	6,531	550
cGAR-L4	37,19	1000	2,79	0,6	1,66	0,2	1,27	%8	6,58	522
cGAR-L5	34,71	1000	2,78	0,59	1,59	0,2	1,46	%10	6,07	520
cGAR-L6	34,48	1000	2,85	0,66	2,42	0,2	1,15	%8	6,602	498
cGAR-L7	129,72	1000	2,81	0,73	1,03	0,19	2,27	%18	4,76	490
cGAR-L8	129,02	1000	2,84	0,78	1,4	0,19	2,33	%14	5,204	499
cGAR-L9	35,42	1000	2,83	0,57	1,34	0,19	1,19	%16	6,901	458
GA	128,61	1000	3,19	0,8	1,32	0,2	2,52	%12	3,633	624
scGA-L2	33,31	1000	4,37	0,42	1,02	0,12	0,85	%0	20,109	139
scGA-L3	32,86	1000	4,14	0,42	1,08	0,12	0,92	%0	22,33	105
scGA-L4	32,89	1000	4,16	0,43	1,1	0,13	0,89	%0	23,946	94
scGA-L5	32,74	1000	4,22	0,41	1	0,12	0,76	%0	30,204	77
scGA-L6	33,04	1000	4,24	0,41	1	0,12	0,78	%0	25,73	84
scGA-L7	32,91	1000	4,25	0,43	1,06	0,13	0,84	%0	30,3	72
scGA-L8	33,03	1000	4,3	0,43	1,02	0,12	0,8	%0	35,576	62
scGA-L9	32,57	1000	4,33	0,43	1,04	0,13	0,85	%0	25,682	89
scGAR-L2	33,47	1000	4,94	0,64	1,18	0,21	1	%0	18,165	138
scGAR-L3	32,91	1000	4,58	0,66	1,32	0,23	1,14	%0	25,905	86
scGAR-L4	33,04	1000	4,54	0,62	1,14	0,22	0,89	%0	27,555	78
scGAR-L5	32,9	1000	4,49	0,61	1,1	0,23	0,89	%0	24,973	83
scGAR-L6	32,97	1000	4,64	0,64	1,24	0,22	1,1	%0	28,672	74
scGAR-L7	32,94	1000	4,67	0,63	1,1	0,21	0,9	%0	29,789	70
scGAR-L8	33,1	1000	4,66	0,64	1,16	0,23	0,98	%0	25,994	79
scGAR-L9	32,87	1000	4,7	0,64	1,18	0,22	0,95	%0	27,735	75

Yol uzunluğu optimizasyonunu esas alan algoritmalar uygunluk değerlerine göre (Bkz. Şekil 4.1) ve hesaplama sürelerine göre (Bkz. Şekil 4.2) karşılaştırılmıştır.



Şekil 4.1. Yol uzunluğu optimizasyonunu esas alan algoritmaların uygunluk değerlerine göre karşılaştırması



Şekil 4.2. Yol uzunluğu optimizasyonunu esas alan algoritmaların hesaplama sürelerine göre karşılaştırması

Yol uzunluğu optimizasyonunu esas alan algoritmaların, Harita-1 üzerindeki testlerinde, amaç fonksiyonuna göre geçerli ancak pratikte uygulanamaz yollar ürettiği görülmektedir (Bkz. Tablo 4.5). Üretilen yolların performansa dayalı uygunluk değerlerinin ceza puanından yüksek olması nedeniyle yolların kullanılamaz olduğu anlaşılmaktadır. Bunun nedeni, İHA'nın azami tırmanma oranı ve alçalma oranının her irtifada aynı olmamasıdır. Bu kontrol sadece performansa dayalı planlamada yapılmaktadır. Bu testlerin bir amacı da aslında performansa dayalı yol planlamanın standart bir planlamadan çok daha fazlası olduğunun gösterilmesidir. Harita-4'te (Bkz. Tablo 4.8) üretilen yolların tamamen geçersiz olduğu; Harita-3'te (Bkz. Tablo 4.7) ve Harita-2'de (Bkz. Tablo 4.6) her rotada yapılan 5 testten bazılarının geçerli yol üretebildiği; Harita-1'de (Bkz. Tablo 4.5) scGA(R)'nin tüm varyantlarının ve cGA (L2, L5, L6, L8 mahallelerinin) ürettiği yolların geçerli olduğu ancak diğer algoritmalar

tarafından üretilen yollardan bazılarının geçerliği olduğu görülmektedir. Yukarıdaki tablolarda gösterilen yol uzunluğu optimizasyonunu esas alan algoritmaların oluşturduğu yollar performansa dayalı uygunluk değeri ile kıyaslandığında, aşağıda sunulacak tablolardaki hava aracı performansına dayalı üretilen yolların uygunluk değerinin, beklendiği gibi daha düşük olduğu açıkça fark edilmektedir.

Hava aracı performansına dayalı yaklaşımda algoritmalar karşılaştırıldığında Harita-1’de, (Bkz. Tablo 4.9) scGAR-L9 algoritması en iyi, cGAR-L8 algoritması ise en kötü sonuçları elde etmiştir. scGA-L7, geleneksel GA’ya göre yaklaşık 5,4 kat daha yüksek uygunluk değerinde yollar üretebilmiştir.

Tablo 4.9. Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-1 üzerindeki test sonuçları

Algoritma	Ort. performansa dayalı uygunluk	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yinleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
cGA-L2	233,18	3,37	1,32	0,94	0,23	6,3	%26,0	2,389	844
cGA-L2-Ç	47,07	1,5	0,53	5,32	0,16	1,75	%24,0	3,306	879
cGA-L3	49,22	3,19	0,7	2,54	0,24	1,99	%22,0	2,675	792
cGA-L3-Ç	138,84	1,47	0,56	2,16	0,16	2,7	%24,0	2,423	829
cGA-L4	232,35	3,3	1,32	1,23	0,22	6,12	%24,0	1,913	735
cGA-L4-Ç	138,67	1,51	0,55	2,93	0,16	1,87	%18,0	2,179	789
cGA-L5	46,74	3,4	1,02	3,89	0,24	5,33	%18,0	2,576	702
cGA-L5-Ç	139,39	1,53	0,66	5,3	0,16	1,85	%26,0	2,511	798
cGA-L2	233,18	3,37	1,32	0,94	0,23	6,3	%26,0	2,389	844
cGA-L2-Ç	47,07	1,5	0,53	5,32	0,16	1,75	%24,0	3,306	879
cGA-L3	49,22	3,19	0,7	2,54	0,24	1,99	%22,0	2,675	792
cGA-L3-Ç	138,84	1,47	0,56	2,16	0,16	2,7	%24,0	2,423	829
cGA-L4	232,35	3,3	1,32	1,23	0,22	6,12	%24,0	1,913	735
cGA-L4-Ç	138,67	1,51	0,55	2,93	0,16	1,87	%18,0	2,179	789
cGA-L5	46,74	3,4	1,02	3,89	0,24	5,33	%18,0	2,576	702
cGA-L5-Ç	139,39	1,53	0,66	5,3	0,16	1,85	%26,0	2,511	798
cGA-L2	233,18	3,37	1,32	0,94	0,23	6,3	%26,0	2,389	844
cGA-L2-Ç	47,07	1,5	0,53	5,32	0,16	1,75	%24,0	3,306	879
cGA-L3	49,22	3,19	0,7	2,54	0,24	1,99	%22,0	2,675	792
cGA-L3-Ç	138,84	1,47	0,56	2,16	0,16	2,7	%24,0	2,423	829
cGA-L4	232,35	3,3	1,32	1,23	0,22	6,12	%24,0	1,913	735
cGA-L4-Ç	138,67	1,51	0,55	2,93	0,16	1,87	%18,0	2,179	789
cGA-L5	46,74	3,4	1,02	3,89	0,24	5,33	%18,0	2,576	702
cGA-L5-Ç	139,39	1,53	0,66	5,3	0,16	1,85	%26,0	2,511	798
cGA-L6	139,17	3,43	1,16	1,7	0,24	6,14	%26,0	2,108	674
cGA-L6-Ç	139,3	1,57	0,61	4,09	0,17	1,62	%20,0	2,316	760
cGA-L7	138,78	3,31	1,4	4,55	0,24	4,74	%22,0	2,2	683
cGA-L7-Ç	138,03	1,62	0,52	2,55	0,17	1,57	%24,0	2,052	752
cGA-L8	138,53	3,28	1,11	3,11	0,24	4,47	%22,0	2,124	629
cGA-L8-Ç	231,12	1,6	0,65	1,62	0,17	2,32	%28,0	1,808	739
cGA-L9	139,03	3,29	1,23	3,57	0,23	5,05	%22,0	2,44	614
cGA-L9-Ç	139,35	1,7	0,59	3,38	0,17	1,91	%26,0	2,377	719

Tablo 4.9. (Devam) *Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-1 üzerindeki test sonuçları*

Algoritma	Ort. performans dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yinleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
cGAR-L2	140,27	3,23	1,52	4,61	0,5	4,87	%36,0	3,747	945
cGAR-L2-Ç	234	1,54	0,82	2,7	0,31	2,13	%38,0	4,092	983
cGAR-L3	232,45	3,08	1,51	2,34	0,42	5,39	%40,0	2,473	885
cGAR-L3-Ç	235,13	1,52	0,87	3,43	0,32	2,66	%42,0	3,264	922
cGAR-L4	138,88	3,14	1,26	3,37	0,4	3,49	%38,0	2,423	835
cGAR-L4-Ç	232,53	1,55	0,79	2,25	0,3	2,15	%40,0	3,161	883
cGAR-L5	325,99	3,16	1,85	1,21	0,4	5,71	%42,0	1,755	837
cGAR-L5-Ç	328,54	1,57	0,88	2,42	0,35	2,24	%38,0	2,507	877
cGAR-L6	43,78	3,23	1,11	4,13	0,43	3,75	%30,0	2,512	784
cGAR-L6-Ç	141,38	1,61	0,74	4	0,32	1,8	%48,0	3,147	863
cGAR-L7	231,84	3,13	1,51	2,17	0,48	5,68	%42,0	1,495	747
cGAR-L7-Ç	138,66	1,66	0,7	3,46	0,29	1,65	%48,0	3,022	836
cGAR-L8	330,56	3,19	1,92	3,62	0,43	5,68	%44,0	1,378	777
cGAR-L8-Ç	136,52	1,68	0,76	2,85	0,32	2,46	%44,0	3,453	858
cGAR-L9	139,99	3,1	1,29	4,03	0,34	3,76	%38,0	1,782	730
cGAR-L9-Ç	326,57	1,78	1,06	2,54	0,31	2,53	%42,0	2,882	806
GA	230,9	3,45	1,71	2,77	0,41	5,27	%38,0	1,312	922
GA-Ç	233,88	1,14	0,69	3,79	0,32	1,67	%38,0	3,214	1048
scGA-L2	43,7	5,24	0,78	1,64	0,32	2,03	%0,0	3,587	389
scGA-L2-Ç	43,37	2,02	0,38	1,68	0,18	0,94	%0,0	3,02	508
scGA-L3	43,02	4,98	0,75	1,64	0,31	1,97	%0,0	4,808	254
scGA-L3-Ç	43,43	2,12	0,37	1,4	0,18	0,94	%0,0	3,188	398
scGA-L4	43,1	5,12	0,71	1,5	0,29	1,55	%0,0	5,622	212
scGA-L4-Ç	43,16	2,11	0,41	1,82	0,19	1,01	%0,0	3,737	350
scGA-L5	42,97	5,2	0,73	1,5	0,32	1,91	%0,0	6,292	180
scGA-L5-Ç	43,24	2,09	0,35	1,3	0,19	0,64	%0,0	3,595	354
scGA-L6	42,89	5,38	0,74	1,42	0,33	1,89	%0,0	6,973	167
scGA-L6-Ç	43,14	2,1	0,36	1,26	0,18	0,66	%0,0	3,526	349
scGA-L7	42,69	5,36	0,72	1,38	0,3	1,91	%0,0	6,286	188
scGA-L7-Ç	43,04	2,17	0,4	1,66	0,19	1,05	%0,0	3,28	371
scGA-L8	42,96	5,35	0,71	1,34	0,33	1,36	%0,0	6,445	172
scGA-L8-Ç	43,24	2,21	0,38	1,56	0,2	0,69	%0,0	3,611	334
scGA-L9	42,96	5,19	0,68	1,4	0,32	1,16	%0,0	7,869	153
scGA-L9-Ç	43,12	2,27	0,39	1,5	0,2	0,89	%0,0	3,452	354
scGAR-L2	43,58	5,35	1,53	3,54	0,65	3,23	%0,0	3,17	444
scGAR-L2-Ç	43,77	2,35	0,74	3,46	0,39	1,69	%2,0	2,997	558
scGAR-L3	42,71	4,95	1,26	2,82	0,61	2,51	%0,0	4,32	250
scGAR-L3-Ç	43,47	2,27	0,7	3,14	0,36	1,42	%8,0	3,335	461
scGAR-L4	42,8	5,09	1,45	3,34	0,57	2,85	%0,0	5,179	205
scGAR-L4-Ç	43,08	2,27	0,72	3,36	0,37	1,52	%2,0	3,747	407
scGAR-L5	42,7	5,19	1,37	2,92	0,59	2,76	%0,0	5,889	182
scGAR-L5-Ç	43,41	2,34	0,72	3,24	0,38	1,38	%2,0	3,385	429
scGAR-L6	42,74	5,27	1,48	3,18	0,6	2,92	%0,0	5,775	184
scGAR-L6-Ç	43,24	2,36	0,71	3,16	0,37	1,27	%8,0	3,867	383
scGAR-L7	42,71	5,11	1,3	2,8	0,59	2,52	%0,0	7,611	138
scGAR-L7-Ç	42,89	2,36	0,7	3,16	0,39	1,35	%8,0	4,178	366
scGAR-L8	42,75	5,16	1,27	2,54	0,6	2,5	%0,0	6,977	151
scGAR-L8-Ç	43,35	2,41	0,77	3,58	0,39	1,5	%4,0	3,549	407
scGAR-L9	42,45	5	1,3	2,96	0,6	2,76	%0,0	6,165	168
scGAR-L9-Ç	43,06	2,51	0,78	3,54	0,42	1,36	%2,0	3,827	352

Hava aracı performansına dayalı yaklaşımda algoritmalar karşılaştırıldığında Harita-2’de, (Bkz. Tablo 4.10) scGAR-L8 algoritması en iyi, cGAR-L5-Ç algoritması ise en kötü sonuçları elde etmiştir.

Tablo 4.10. Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-2 üzerindeki test sonuçları

Algoritma	Ort. performansa dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yinleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
cGA-L2	24,37	1,51	0,41	3,92	0,08	0,81	%12,0	11,697	620
cGA-L2-Ç	24,27	0,86	0,2	2,53	0,08	0,33	%4,0	12,694	608
cGA-L3	24,12	1,45	0,35	3,39	0,07	0,74	%2,0	10,979	536
cGA-L3-Ç	24,37	0,84	0,21	3,02	0,09	0,38	%4,0	10,347	562
cGA-L4	24,38	1,48	0,34	3,26	0,07	0,7	%8,0	11,366	531
cGA-L4-Ç	24,22	0,85	0,23	3,79	0,09	0,4	%6,0	11,294	513
cGA-L5	24	1,48	0,39	3,65	0,08	1,13	%12,0	9,946	472
cGA-L5-Ç	24,11	0,84	0,19	2,35	0,08	0,33	%8,0	9,73	482
cGA-L6	23,93	1,5	0,3	2,46	0,08	0,54	%10,0	9,807	433
cGA-L6-Ç	24,04	0,89	0,18	1,93	0,09	0,3	%6,0	9,33	472
cGA-L7	23,97	1,51	0,34	3,23	0,07	0,72	%2,0	10,353	394
cGA-L7-Ç	23,95	0,89	0,21	2,68	0,08	0,45	%2,0	10,671	414
cGA-L8	24,06	1,51	0,26	1,61	0,07	0,52	%8,0	9,871	415
cGA-L8-Ç	24,02	0,92	0,23	3,21	0,09	0,43	%4,0	10,03	450
cGA-L9	23,73	1,53	0,28	1,84	0,08	0,65	%4,0	11,848	325
cGA-L9-Ç	23,81	0,97	0,21	2,37	0,09	0,34	%6,0	9,887	387
cGAR-L2	24,26	1,64	0,5	2,69	0,14	1,07	%2,0	12,664	621
cGAR-L2-Ç	24,69	0,97	0,33	3,77	0,12	0,56	%4,0	12,164	613
cGAR-L3	24,09	1,56	0,41	1,74	0,13	0,85	%10,0	9,571	558
cGAR-L3-Ç	24,23	0,94	0,3	2,57	0,12	0,56	%8,0	10,218	565
cGAR-L4	24,32	1,57	0,47	2,62	0,13	1,11	%8,0	9,266	537
cGAR-L4-Ç	24,05	0,96	0,29	1,91	0,11	0,5	%8,0	9,538	517
cGAR-L5	24,06	1,6	0,5	3,09	0,13	1,54	%14,0	9,727	425
cGAR-L5-Ç	119,27	0,94	0,36	1,7	0,12	1,23	%10,0	9,61	443
cGAR-L6	24,02	1,6	0,43	2,04	0,14	0,7	%14,0	10,419	450
cGAR-L6-Ç	24,04	1	0,29	1,74	0,13	0,6	%10,0	9,875	432
cGAR-L7	23,93	1,62	0,4	2,11	0,13	0,6	%10,0	9,995	400
cGAR-L7-Ç	23,91	1	0,33	3,51	0,13	0,54	%10,0	10,281	433
cGAR-L8	23,71	1,63	0,41	1,75	0,14	0,68	%14,0	9,705	371
cGAR-L8-Ç	25,23	1,04	0,31	2,49	0,12	0,45	%6,0	10,867	389
cGAR-L9	24,13	1,68	0,47	2,79	0,13	0,94	%4,0	11,528	372
cGAR-L9-Ç	24,13	1,09	0,32	2,41	0,13	0,51	%10,0	10,46	409
GA	23,7	1,86	0,5	3,13	0,13	0,98	%8,0	7,335	564

Tablo 4.10. (Devam) *Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-2 üzerindeki test sonuçları*

Algoritma	Ort. performans dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yinleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
GA-Ç	24,02	0,67	0,28	2,95	0,11	0,63	%6,0	9,769	716
scGA-L2	24,07	2,44	0,4	1,58	0,1	0,72	%0,0	13,809	371
scGA-L2-Ç	23,98	1,03	0,2	1,4	0,08	0,36	%0,0	14,571	360
scGA-L3	23,88	2,27	0,35	1,34	0,1	0,68	%0,0	15,23	291
scGA-L3-Ç	23,81	1,03	0,2	1,22	0,09	0,33	%0,0	14,707	295
scGA-L4	23,87	2,28	0,34	1,1	0,1	0,68	%0,0	15,923	266
scGA-L4-Ç	23,89	1,01	0,2	1,22	0,09	0,36	%0,0	14,988	282
scGA-L5	23,93	2,26	0,33	1,12	0,1	0,7	%0,0	15,352	268
scGA-L5-Ç	23,73	1,05	0,2	1,28	0,1	0,35	%0,0	16,08	247
scGA-L6	23,89	2,27	0,34	1,28	0,1	0,72	%0,0	15,953	259
scGA-L6-Ç	23,91	1,04	0,21	1,5	0,1	0,37	%0,0	16,436	253
scGA-L7	23,83	2,3	0,34	1,14	0,1	0,65	%0,0	15,08	264
scGA-L7-Ç	23,87	1,08	0,21	1,18	0,09	0,35	%0,0	17,818	216
scGA-L8	23,93	2,26	0,34	1,16	0,1	0,71	%0,0	13,295	292
scGA-L8-Ç	23,85	1,06	0,22	1,56	0,09	0,35	%0,0	16,745	250
scGA-L9	23,85	2,3	0,36	1,4	0,1	0,67	%0,0	14,741	280
scGA-L9-Ç	23,72	1,11	0,21	1,36	0,09	0,36	%0,0	16,692	237
scGAR-L2	23,88	2,93	0,67	1,58	0,22	1,44	%0,0	12,654	382
scGAR-L2-Ç	23,94	1,39	0,37	1,58	0,15	0,58	%0,0	13,523	366
scGAR-L3	23,84	2,62	0,57	1,32	0,19	1,06	%0,0	12,853	307
scGAR-L3-Ç	23,72	1,31	0,36	1,52	0,14	0,65	%0,0	14,82	280
scGAR-L4	23,86	2,63	0,62	1,64	0,2	1,18	%0,0	13,304	312
scGAR-L4-Ç	23,72	1,32	0,36	1,58	0,15	0,64	%0,0	16,464	262
scGAR-L5	23,85	2,7	0,59	1,3	0,21	1,16	%0,0	12,857	303
scGAR-L5-Ç	23,69	1,34	0,36	1,36	0,17	0,63	%0,0	15,7	258
scGAR-L6	23,91	2,64	0,61	1,52	0,19	1,2	%0,0	12,883	304
scGAR-L6-Ç	23,7	1,35	0,37	1,5	0,16	0,65	%0,0	16,818	234
scGAR-L7	23,81	2,71	0,57	1,14	0,22	0,87	%0,0	11,883	303
scGAR-L7-Ç	23,78	1,37	0,37	1,56	0,16	0,65	%0,0	15,128	254
scGAR-L8	23,67	2,69	0,59	1,38	0,21	0,96	%0,0	12,162	318
scGAR-L8-Ç	23,71	1,38	0,37	1,36	0,16	0,6	%0,0	16,219	237
scGAR-L9	23,98	2,73	0,59	1,3	0,18	1,1	%0,0	13,11	303
scGAR-L9-Ç	23,69	1,44	0,4	1,72	0,16	0,75	%0,0	16,579	237

Hava aracı performansına dayalı yaklaşımda algoritmalar karşılaştırıldığında Harita-3'te, (Bkz. Tablo 4.11) scGA-L5-Ç algoritması en iyi, cGAR-L4-Ç algoritması ise en kötü sonuçları elde etmiştir.

Tablo 4.11. Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-3 üzerindeki test sonuçları

Algoritma	Ort. performans dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yineleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
cGA-L2	219,15	2,82	1,51	0,8	0,07	8,1	%20,0	8,942	593
cGA-L2-Ç	36,72	1,25	0,37	2,4	0,07	1,73	%16,0	9,397	592
cGA-L3	37,88	2,71	0,84	3,74	0,07	4,27	%16,0	7,904	529
cGA-L3-Ç	219,12	1,34	0,66	0,8	0,08	3,09	%20,0	7,443	539
cGA-L4	219,01	2,78	1,45	0,8	0,07	7,54	%20,0	6,835	497
cGA-L4-Ç	129,51	1,29	0,44	1,4	0,07	1,89	%18,0	7,719	549
cGA-L5	219,03	2,85	1,45	0,8	0,07	7,52	%20,0	6,593	472
cGA-L5-Ç	132,42	1,29	0,42	1,16	0,08	1,97	%18,0	7,193	485
cGA-L6	131,15	2,85	1,03	0,92	0,07	7,62	%18,0	6,922	454
cGA-L6-Ç	218,99	1,31	0,63	0,8	0,08	2,95	%20,0	6,795	465
cGA-L7	218,94	2,89	1,45	0,8	0,07	7,55	%20,0	6,718	430
cGA-L7-Ç	218,93	1,35	0,65	0,8	0,08	3,03	%20,0	6,645	452
cGA-L8	125,69	2,88	1,05	1,1	0,08	7,49	%18,0	7,594	426
cGA-L8-Ç	37,93	1,42	0,28	1,45	0,09	1,01	%14,0	8,272	426
cGA-L9	218,95	2,86	1,45	0,8	0,08	7,7	%22,0	6,949	426
cGA-L9-Ç	126,09	1,43	0,51	1,1	0,09	3,11	%18,0	7,711	441
cGAR-L2	127,81	2,78	1,09	1	0,15	6,5	%20,0	8,076	551
cGAR-L2-Ç	219,57	1,32	0,7	1,08	0,13	2,68	%20,0	7,988	536
cGAR-L3	219,51	2,64	1,41	1,05	0,15	5,94	%22,0	6,465	502
cGAR-L3-Ç	219,05	1,42	0,73	0,86	0,12	2,8	%20,0	6,463	517
cGAR-L4	127,17	2,72	1,03	1,08	0,15	6,05	%20,0	7,192	480
cGAR-L4-Ç	219,64	1,38	0,76	1,82	0,12	2,86	%20,0	6,512	480
cGAR-L5	219,29	2,74	1,43	1,16	0,15	5,96	%20,0	5,973	448
cGAR-L5-Ç	126,62	1,39	0,56	1,26	0,12	2,56	%18,0	7,172	463
cGAR-L6	126,77	2,82	1,02	1,03	0,14	5,93	%20,0	7,276	453
cGAR-L6-Ç	219,44	1,41	0,74	1,65	0,13	2,74	%22,0	6,4	446
cGAR-L7	126,05	2,81	1,05	1,38	0,15	5,99	%20,0	6,858	446
cGAR-L7-Ç	219,04	1,45	0,73	0,97	0,14	2,68	%24,0	6,315	443
cGAR-L8	127,13	2,84	1,15	1,9	0,15	5,93	%18,0	7,381	414
cGAR-L8-Ç	219,17	1,48	0,75	1,23	0,13	2,75	%22,0	6,516	418
cGAR-L9	37,15	2,86	0,56	1,13	0,14	1,31	%18,0	7,575	420
cGAR-L9-Ç	219,06	1,51	0,75	0,83	0,13	2,85	%22,0	6,338	419
GA	127,44	3,05	1,16	1,65	0,16	6,46	%20,0	7,13	556
GA-Ç	126,55	1	0,47	0,95	0,11	2,1	%20,0	8,76	609
scGA-L2	37,42	4,36	0,76	2,06	0,08	3,47	%2,0	8,308	633
scGA-L2-Ç	37,61	1,64	0,4	2,52	0,08	1,89	%0,0	9,336	632
scGA-L3	36,64	4,08	0,63	1,74	0,08	2,51	%2,0	7,92	609
scGA-L3-Ç	36,22	1,64	0,34	2,04	0,09	1,18	%0,0	8,652	562

Tablo 4.11. (Devam) *Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-3 üzerindeki test sonuçları*

Algoritma	Ort. performansa dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yineleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
scGA-L4	36,51	4,17	0,7	1,82	0,07	3,44	%2,0	8,467	597
scGA-L4-Ç	36,66	1,7	0,39	2,27	0,08	1,69	%2,0	9,189	540
scGA-L5	36,44	4,25	0,66	1,72	0,09	2,97	%0,0	7,38	576
scGA-L5-Ç	35,8	1,68	0,35	1,96	0,08	1,23	%0,0	9,661	529
scGA-L6	35,93	4,35	0,66	1,7	0,09	3,05	%0,0	7,54	577
scGA-L6-Ç	36,32	1,69	0,39	2,24	0,08	1,89	%0,0	10,195	522
scGA-L7	36,5	4,35	0,63	1,66	0,08	2,49	%0,0	7,726	570
scGA-L7-Ç	36	1,71	0,36	2,02	0,08	1,42	%0,0	8,611	499
scGA-L8	36,04	4,44	0,76	1,94	0,09	3,88	%0,0	7,542	569
scGA-L8-Ç	36,6	1,72	0,4	2,28	0,09	1,82	%2,0	9,188	514
scGA-L9	36,27	4,37	0,68	1,78	0,09	3,21	%0,0	7,343	595
scGA-L9-Ç	35,97	1,79	0,4	2,4	0,1	1,6	%0,0	9,566	484
scGAR-L2	38,81	5,01	1,21	2,72	0,18	7	%2,0	5,351	691
scGAR-L2-Ç	38,22	1,9	0,54	2,58	0,13	2,32	%0,0	6,894	658
scGAR-L3	36,07	4,56	0,89	1,96	0,17	3,97	%0,0	5,683	624
scGAR-L3-Ç	36,52	1,89	0,5	2,23	0,14	1,71	%2,0	6,535	592
scGAR-L4	36,68	4,65	1,07	2,36	0,17	5,82	%0,0	5,616	620
scGAR-L4-Ç	35,89	1,9	0,49	2,08	0,14	1,56	%0,0	6,35	564
scGAR-L5	36,98	4,71	1,03	2,28	0,18	5,06	%0,0	5,178	603
scGAR-L5-Ç	36,48	1,87	0,5	2,11	0,13	1,7	%4,0	6,45	545
scGAR-L6	36,86	4,91	0,97	2,06	0,18	4,2	%0,0	5,378	602
scGAR-L6-Ç	36,93	1,94	0,49	2,02	0,15	1,59	%0,0	6,569	525
scGAR-L7	36,87	4,89	1,05	2,3	0,18	5,31	%0,0	5,048	623
scGAR-L7-Ç	37,45	1,96	0,52	2,32	0,14	1,7	%0,0	6,28	531
scGAR-L8	36,45	4,86	0,94	1,98	0,18	4,32	%0,0	5,18	589
scGAR-L8-Ç	37,81	1,98	0,58	2,88	0,13	2,28	%2,0	6,114	558
scGAR-L9	37,1	4,8	1,04	2,3	0,17	5,11	%0,0	5,466	599
scGAR-L9-Ç	36,99	2,03	0,58	2,64	0,15	2,25	%0,0	7,242	524

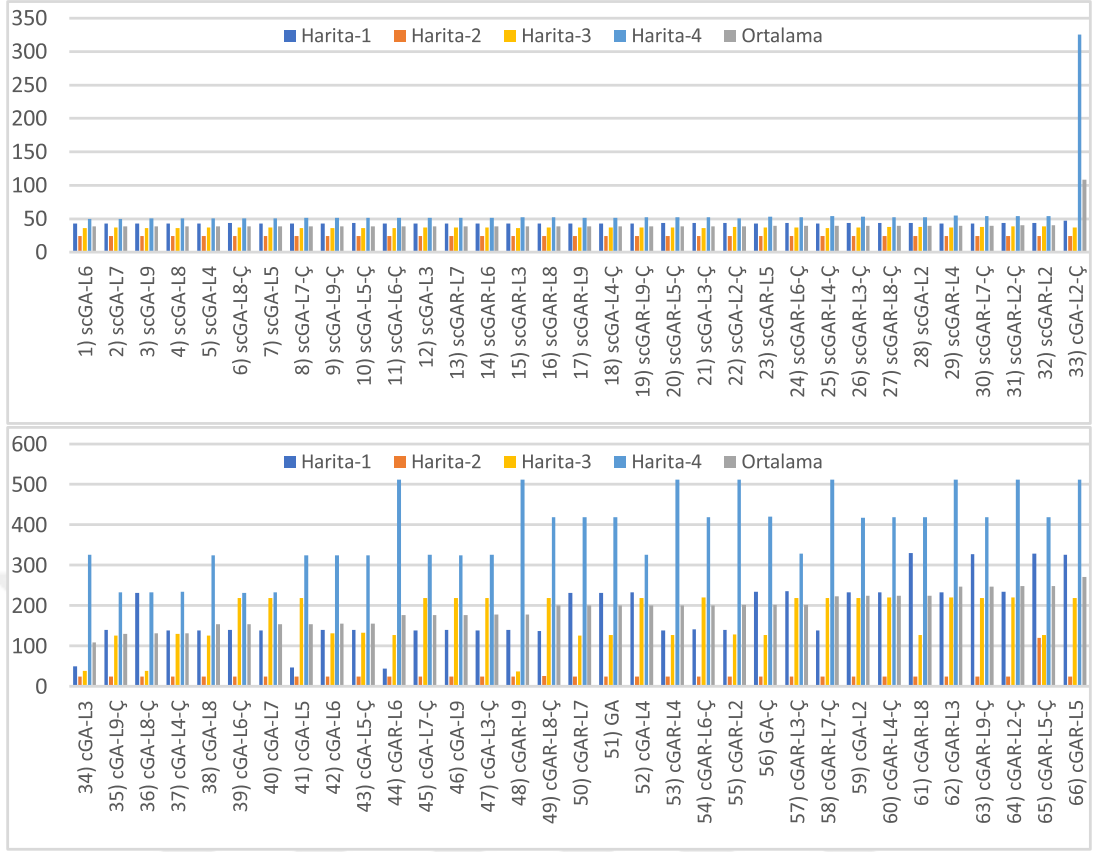
Hava aracı performansına dayalı yaklaşımda algoritmalar karşılaştırıldığında Harita-4'te, (Bkz. Tablo 4.12) scGA-L5-Ç algoritması en iyi, cGAR-L4-Ç algoritması ise en kötü sonuçları elde etmiştir.

Tablo 4.12. Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-4 üzerindeki test sonuçları

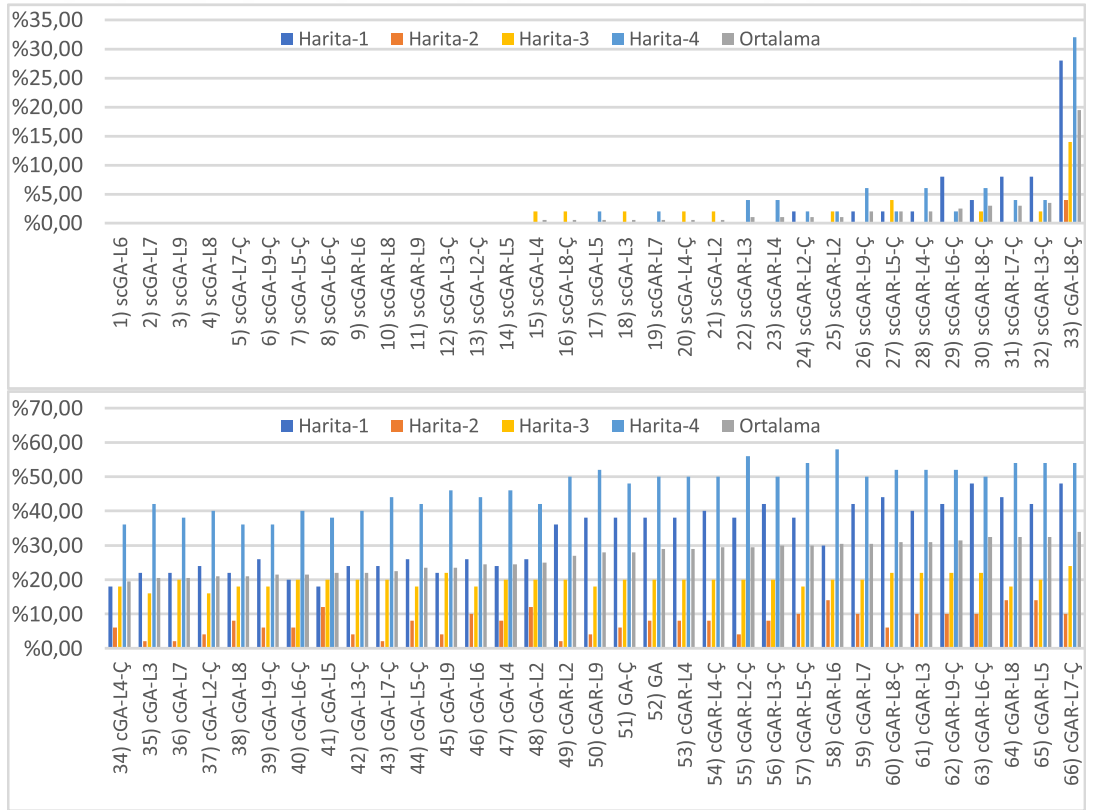
Algoritma	Ort. performansa dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yinleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
cGA-L2	418,05	2,62	1,55	0,72	0,1	5,13	%42,0	4,649	937
cGA-L2-Ç	325,19	1,25	0,69	1,17	0,09	2,6	%40,0	4,746	940
cGA-L3	325,64	2,48	1,14	0,95	0,09	4,83	%42,0	4,468	951
cGA-L3-Ç	325,9	1,26	0,62	1,28	0,08	2,2	%40,0	4,06	935
cGA-L4	325,18	2,54	1,19	1,14	0,08	4,88	%46,0	3,993	919
cGA-L4-Ç	234,11	1,22	0,6	2,88	0,08	2,2	%36,0	4,185	911
cGA-L5	325,03	2,54	1,33	0,96	0,08	4,84	%38,0	3,801	893
cGA-L5-Ç	323,99	1,24	0,66	1,11	0,08	2,32	%42,0	3,89	880
cGA-L6	324,52	2,59	1,36	1,07	0,08	4,83	%44,0	3,554	856
cGA-L6-Ç	231,14	1,36	0,56	1,43	0,08	2,6	%40,0	3,665	875
cGA-L7	232,03	2,59	1,04	1,56	0,08	4,78	%38,0	3,895	856
cGA-L7-Ç	325,66	1,45	0,74	1,03	0,09	2,71	%44,0	4,018	869
cGA-L8	324,25	2,67	1,36	0,87	0,08	5,1	%36,0	3,653	822
cGA-L8-Ç	232,51	1,35	0,54	1,25	0,09	2,44	%32,0	3,504	830
cGA-L9	324,92	2,64	1,29	0,78	0,08	5,25	%46,0	3,062	813
cGA-L9-Ç	232,15	1,45	0,61	1,09	0,09	3,06	%36,0	3,67	842
cGAR-L2	511,87	2,63	1,94	0,92	0,18	4,57	%50,0	4,476	956
cGAR-L2-Ç	511,88	1,34	1	1,99	0,14	2,59	%56,0	4,436	966
cGAR-L3	511,78	2,49	1,85	1	0,17	4,29	%52,0	3,506	951
cGAR-L3-Ç	328,69	1,35	0,77	2,63	0,14	2,14	%50,0	3,652	929
cGAR-L4	511,9	2,53	1,93	1,84	0,16	4,33	%50,0	3,027	909
cGAR-L4-Ç	419,2	1,31	0,81	0,79	0,13	2,15	%50,0	3,149	908
cGAR-L5	511,89	2,56	1,91	1,38	0,16	4,36	%54,0	3,101	892
cGAR-L5-Ç	418,76	1,33	0,84	1,21	0,13	2,22	%54,0	3,462	909
cGAR-L6	511,9	2,6	1,89	1,13	0,15	4,35	%58,0	2,625	885
cGAR-L6-Ç	418,94	1,45	0,93	2,01	0,13	2,46	%50,0	2,92	885
cGAR-L7	418,2	2,63	1,62	1,46	0,19	4,4	%50,0	2,918	865
cGAR-L7-Ç	511,48	1,54	1,09	0,95	0,14	2,6	%54,0	2,469	855
cGAR-L8	418,26	2,65	1,68	1,82	0,16	4,64	%54,0	2,6	857
cGAR-L8-Ç	418,28	1,46	0,96	0,91	0,15	2,57	%52,0	2,79	845
cGAR-L9	511,38	2,71	2	0,96	0,16	5,23	%52,0	2,261	825
cGAR-L9-Ç	418,81	1,56	0,98	1,08	0,14	2,94	%52,0	3,028	863
GA	418,4	2,89	1,85	0,79	0,18	5,32	%50,0	3,081	945
GA-Ç	420,08	1	0,71	1,02	0,11	1,83	%48,0	4,199	998

Tablo 4.12. (Devam) *Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların Harita-4 üzerindeki test sonuçları*

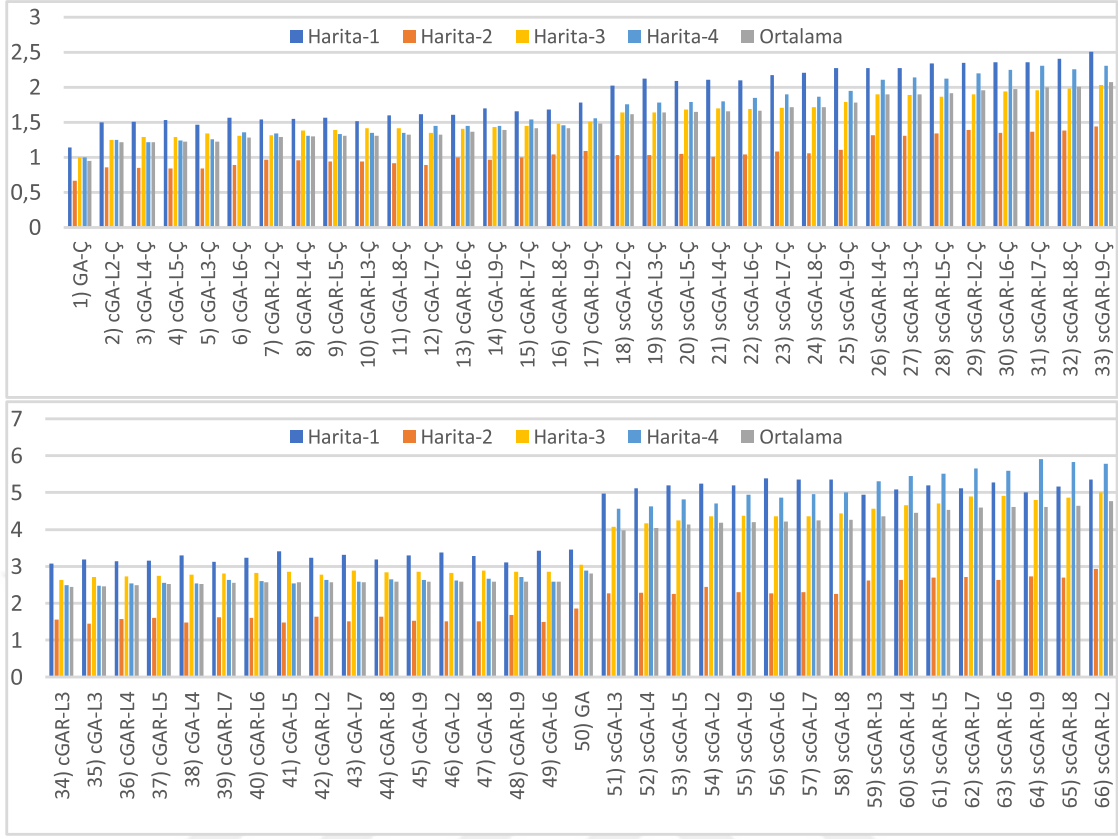
Algoritma	Ort. performansa dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yineleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
scGA-L2	52,19	4,71	0,91	3,66	0,11	2,74	%0,0	3,524	882
scGA-L2-Ç	50,79	1,76	0,35	2,96	0,09	0,84	%0,0	4,037	888
scGA-L3	50,92	4,56	0,83	3,24	0,12	2,08	%0,0	3,437	786
scGA-L3-Ç	52,27	1,78	0,45	4,16	0,09	1,39	%0,0	3,786	792
scGA-L4	50,15	4,62	0,83	2,94	0,13	2,93	%0,0	3,582	740
scGA-L4-Ç	51,45	1,8	0,44	3,88	0,09	1,1	%0,0	3,944	740
scGA-L5	50,48	4,81	0,79	2,98	0,13	1,97	%2,0	3,619	712
scGA-L5-Ç	51,11	1,79	0,38	2,92	0,09	0,92	%0,0	3,687	733
scGA-L6	49,96	4,86	0,9	3,4	0,12	2,57	%0,0	3,777	699
scGA-L6-Ç	50,9	1,85	0,39	3,26	0,09	0,91	%0,0	3,854	720
scGA-L7	49,99	4,96	0,8	2,82	0,12	1,96	%0,0	3,811	668
scGA-L7-Ç	50,93	1,9	0,41	3,42	0,09	0,97	%0,0	4,009	686
scGA-L8	50,67	5	0,81	2,96	0,11	2,47	%0,0	3,654	686
scGA-L8-Ç	50,1	1,87	0,36	2,66	0,1	0,75	%0,0	4,024	671
scGA-L9	50,16	4,94	0,74	2,74	0,12	1,92	%0,0	3,821	669
scGA-L9-Ç	51,06	1,95	0,46	4	0,09	1,01	%0,0	3,911	686
scGAR-L2	53,99	5,78	1,7	5,55	0,25	3,93	%2,0	2,808	957
scGAR-L2-Ç	53,64	2,2	0,68	4,58	0,15	1,89	%2,0	3,334	923
scGAR-L3	52,17	5,3	1,36	4,28	0,24	3,92	%4,0	2,732	835
scGAR-L3-Ç	53,31	2,14	0,78	6,13	0,14	2,21	%4,0	2,763	882
scGAR-L4	54,82	5,45	1,67	5,77	0,22	3,91	%4,0	2,761	818
scGAR-L4-Ç	53,71	2,11	0,76	6,29	0,16	1,69	%6,0	3,097	845
scGAR-L5	52,63	5,51	1,48	4,5	0,22	4,44	%0,0	2,96	747
scGAR-L5-Ç	52,13	2,12	0,79	6,64	0,15	1,53	%2,0	3,142	842
scGAR-L6	51,22	5,59	1,5	4,8	0,21	3,47	%0,0	2,871	731
scGAR-L6-Ç	52,46	2,25	0,8	5,77	0,16	2,46	%2,0	3,186	789
scGAR-L7	51,27	5,66	1,58	4,88	0,21	4,53	%2,0	3,287	713
scGAR-L7-Ç	54,22	2,31	0,85	6,33	0,15	2,46	%4,0	3,064	824
scGAR-L8	51,95	5,83	1,57	4,8	0,22	3,28	%0,0	3,16	712
scGAR-L8-Ç	52,15	2,26	0,78	5,94	0,16	1,67	%6,0	3,326	808
scGAR-L9	51,44	5,9	1,61	4,8	0,22	3,76	%0,0	3,135	692
scGAR-L9-Ç	51,86	2,31	0,87	6,6	0,16	2,45	%6,0	3,249	789



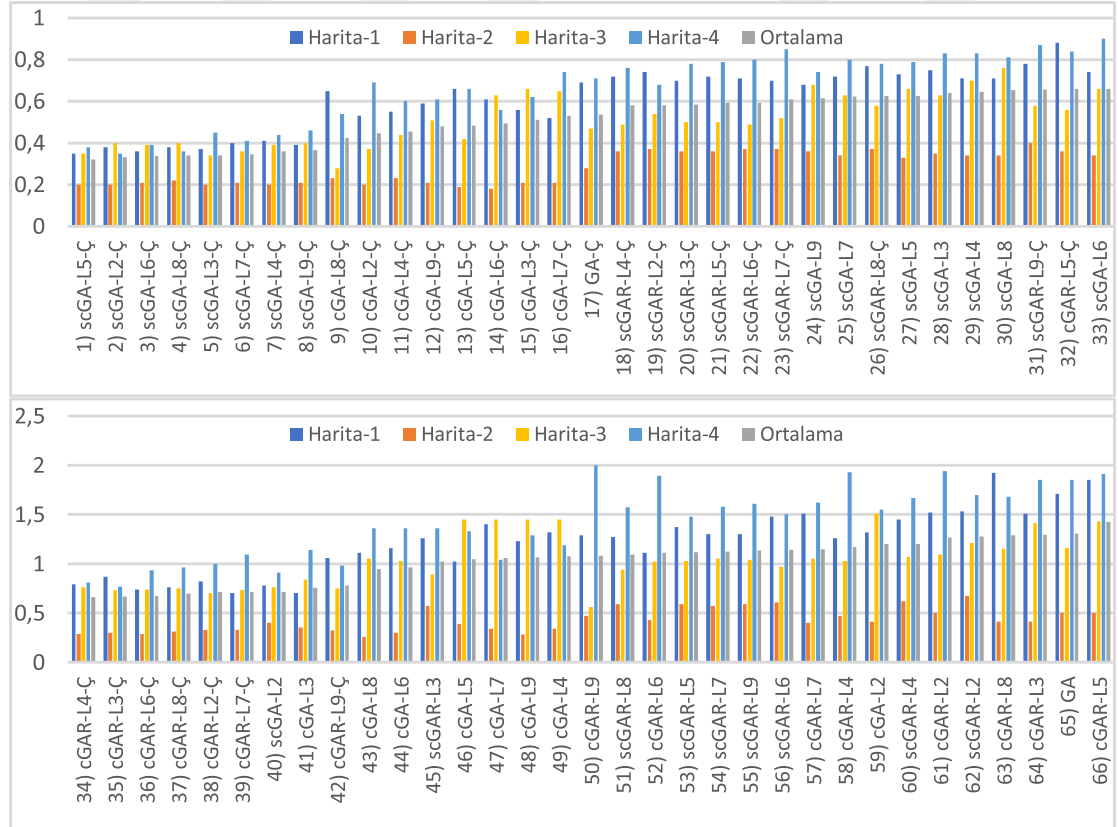
Şekil 4.3. Hava aracı performansına dayalı algoritmaların ortalama uygunluk değerlerine göre karşılaştırması



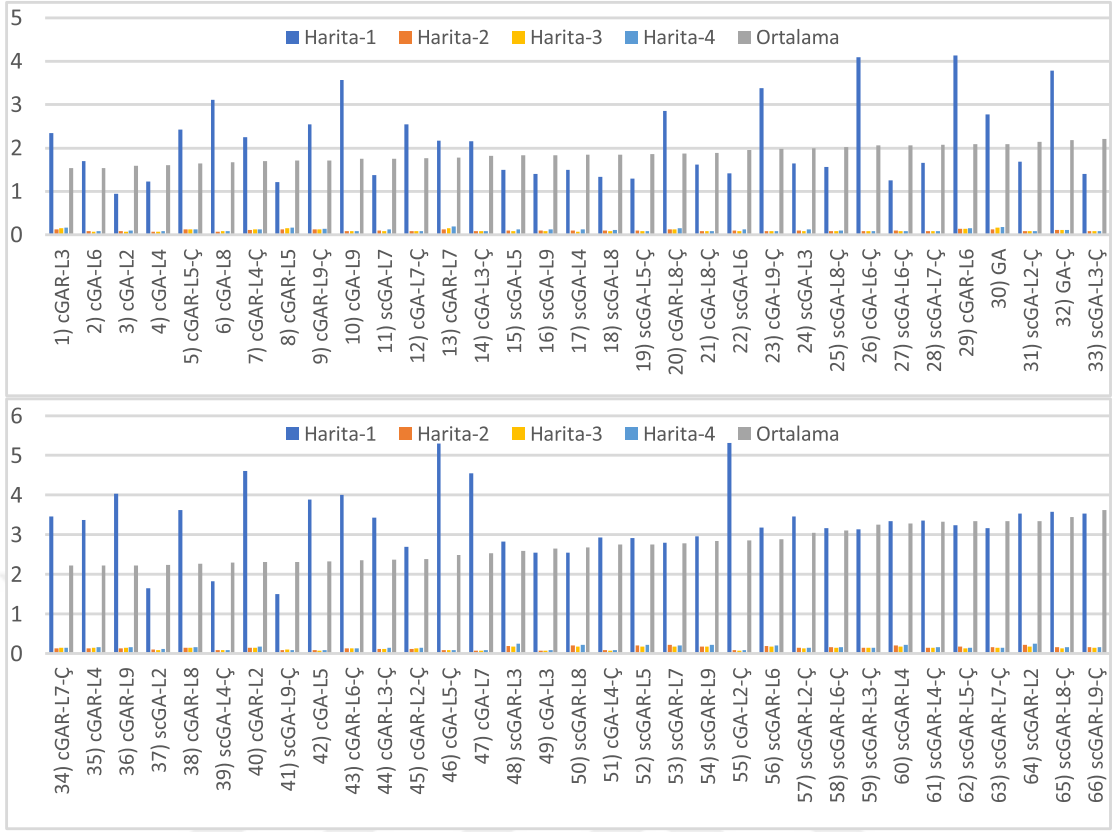
Şekil 4.4. Hava aracı performansına dayalı algoritmaların geçersiz sonuç yüzdelerine göre karşılaştırması



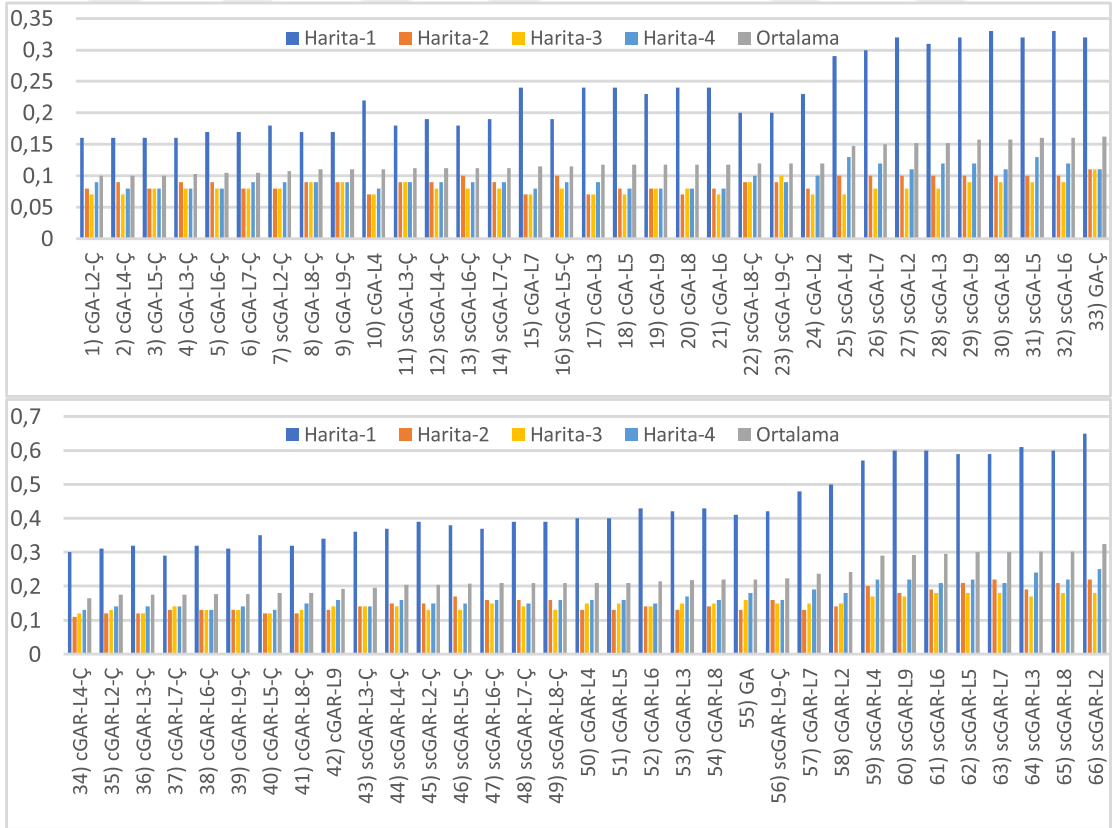
Şekil 4.5. Hava aracı performansına dayalı algoritmaların ortalama hesaplama sürelerine göre karşılaştırması



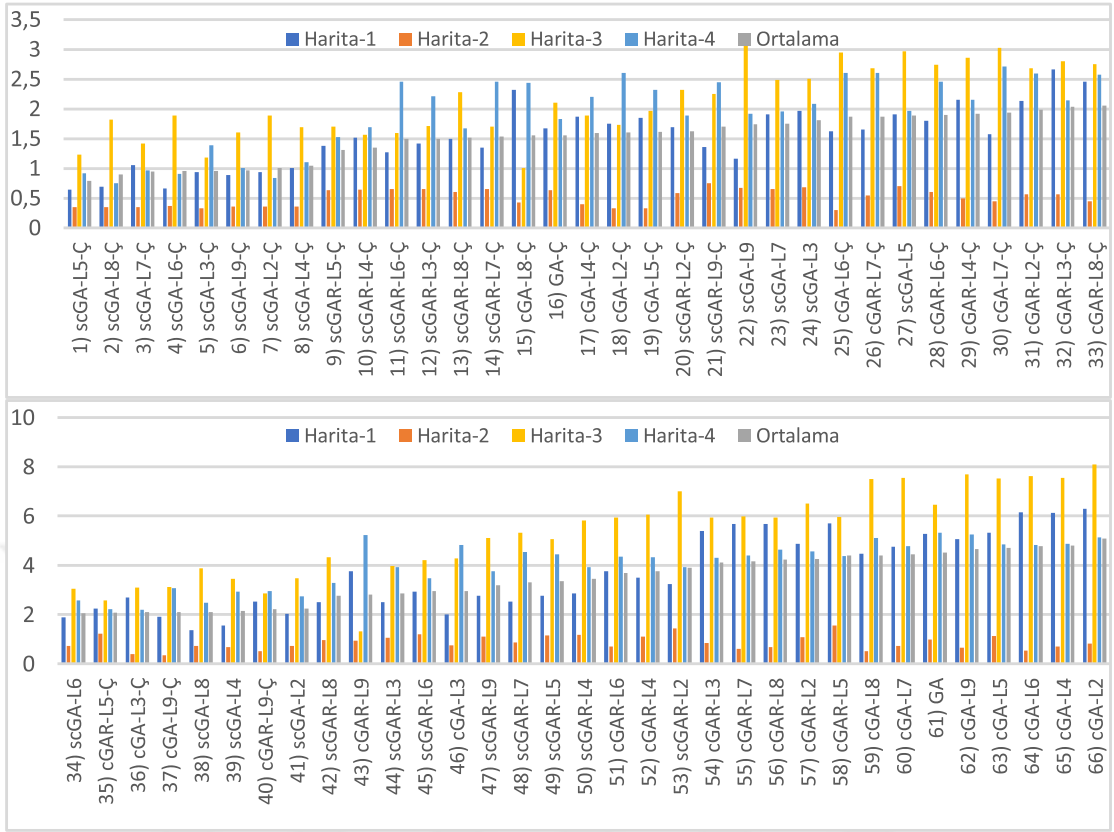
Şekil 4.6. Hava aracı performansına dayalı algoritmaların ortalama ilk geçerli yol bulma sürelerine göre karşılaştırması



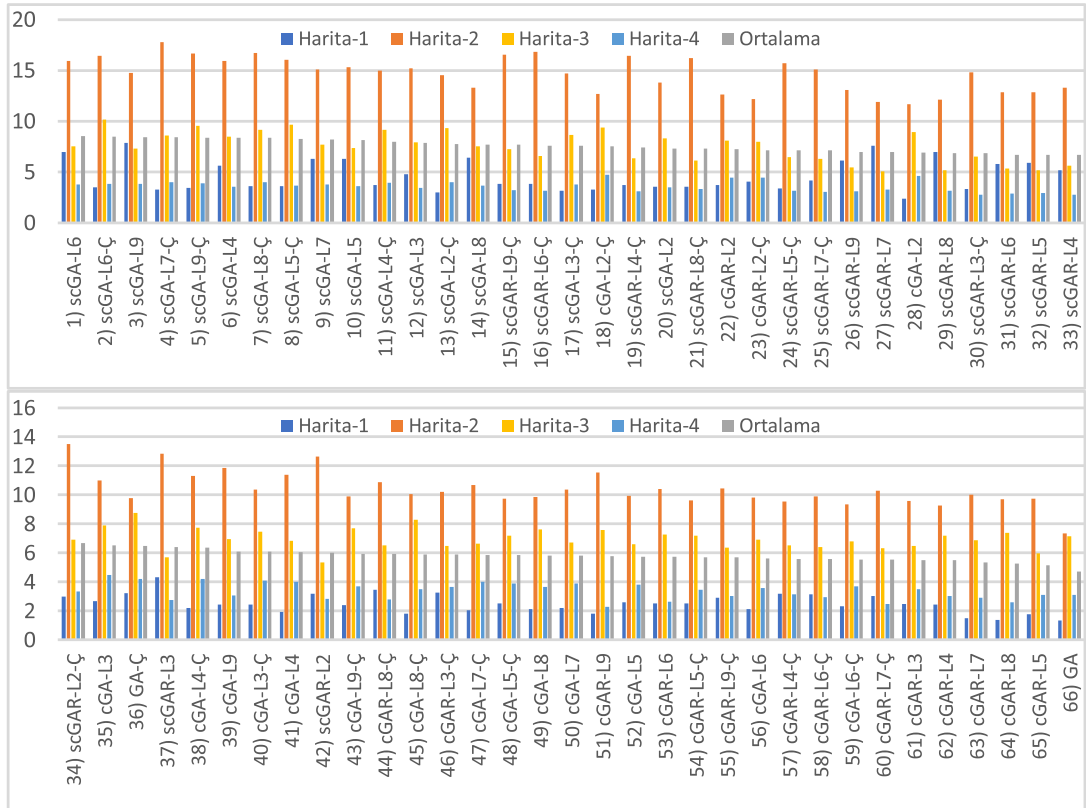
Şekil 4.7. Hava aracı performansına dayalı algoritmaların ortalama ilk geçerli yol bulma yineleme sayısına göre karşılaştırması



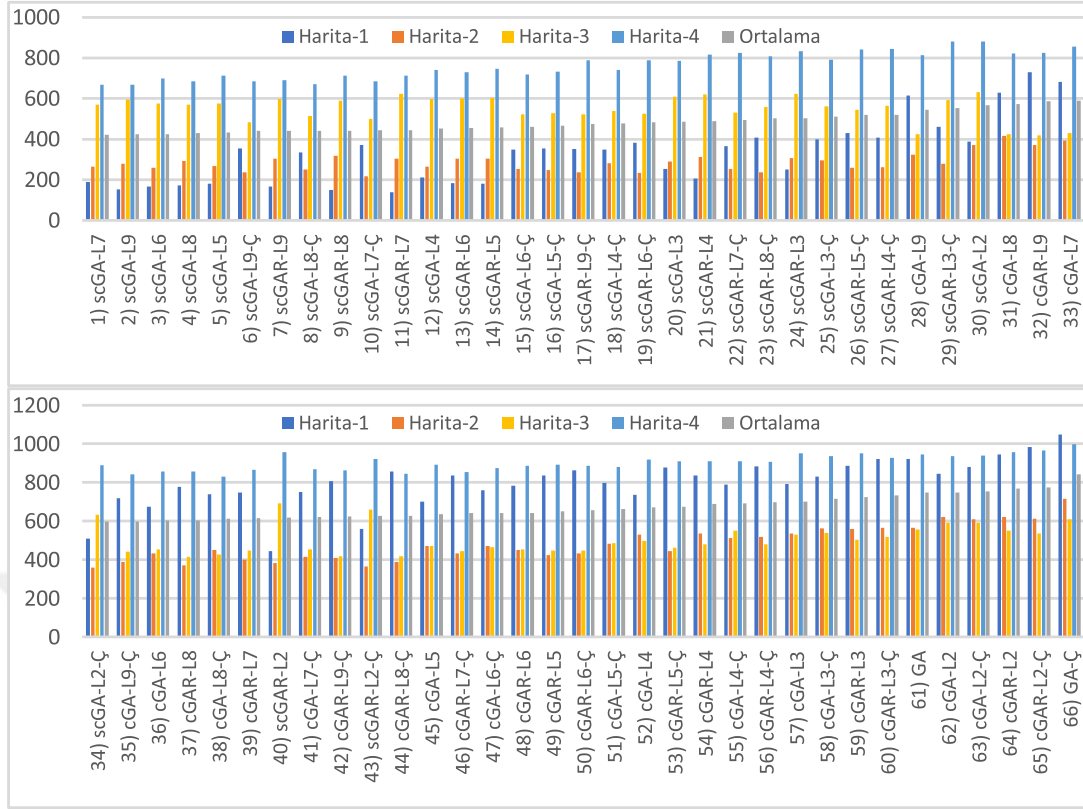
Şekil 4.8. Hava aracı performansına dayalı algoritmaların ortalama en iyi ilk geçerli yol bulma sürelerine göre karşılaştırması



Şekil 4.9. Hava aracı performansına dayalı algoritmaların ortalama en kötü ilk geçerli yol bulma sürelerine göre karşılaştırması

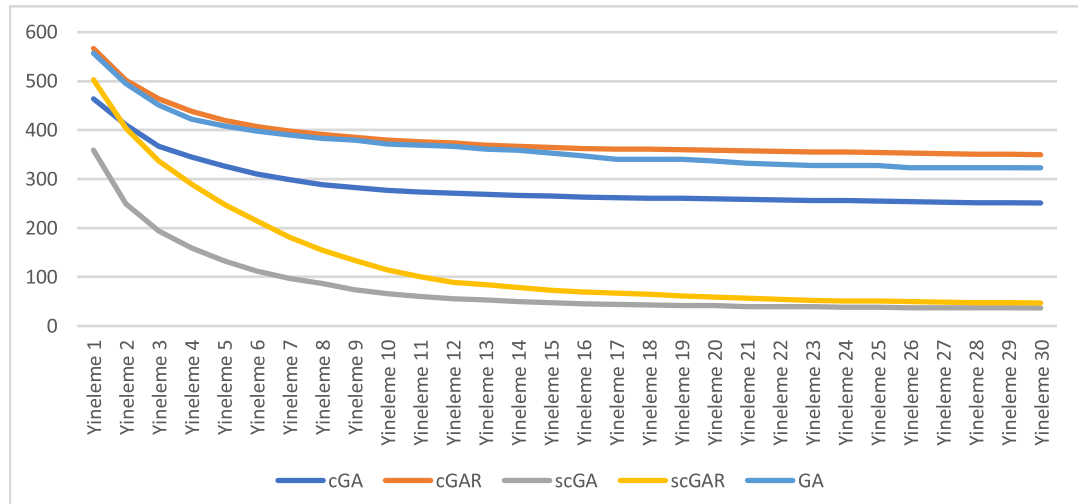


Şekil 4.10. Hava aracı performansına dayalı algoritmaların çeşitlilik oranlarına göre karşılaştırması

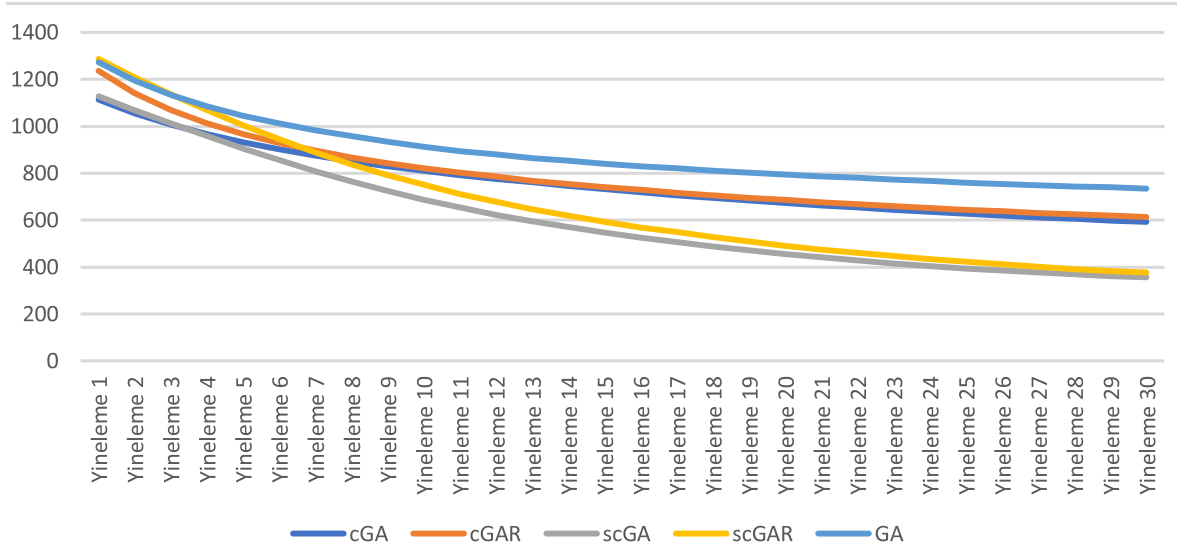


Şekil 4.11. Hava aracı performansına dayalı algoritmaların son popülasyon ortalama uygunluk değerlerine göre karşılaştırması

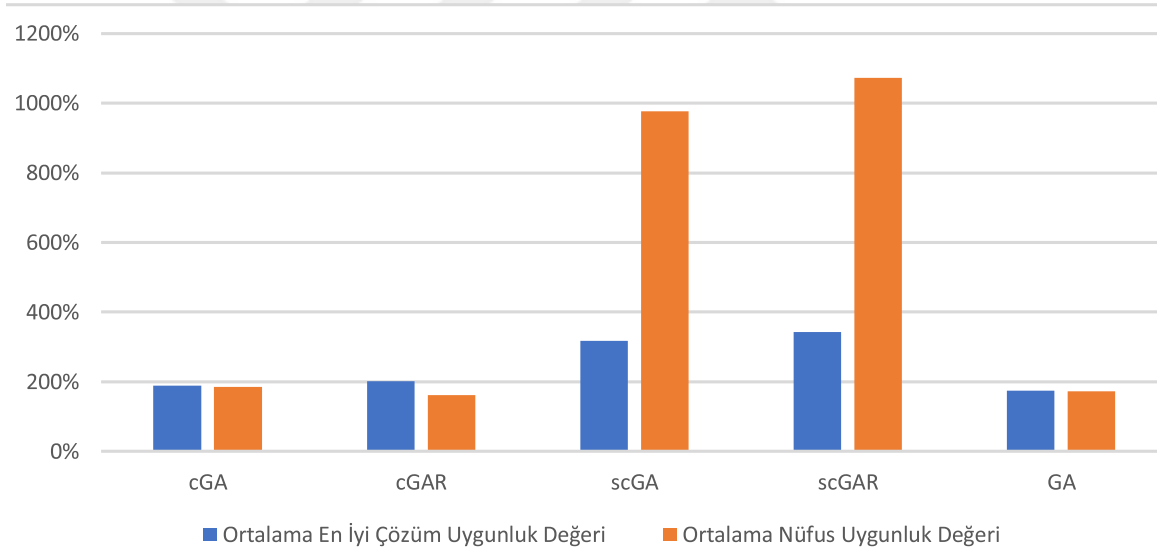
Son popülasyona göre test sonuçları yukarıda sunulmuştur. Algoritma başarısının bir diğer ölçütü popülasyonun iyileşme hızıdır. Yapılan 19800 testin ortalamaları dikkate alınarak hazırlanan tablolarda, yinelemelere göre ortalama en iyi çözüm uygunluk değeri değişimi (Bkz. Şekil 4.12), yinelemelere göre ortalama popülasyon uygunluk değeri değişimi (Bkz. Şekil 4.13) ve tüm yinelemeler sonunda başlangıç popülasyonuna göre iyileşme oranları (Bkz. Şekil 4.14) verilmiştir.



Şekil 4.12. Yinelemelere göre ortalama en iyi çözüm uygunluk değerinin değişimi

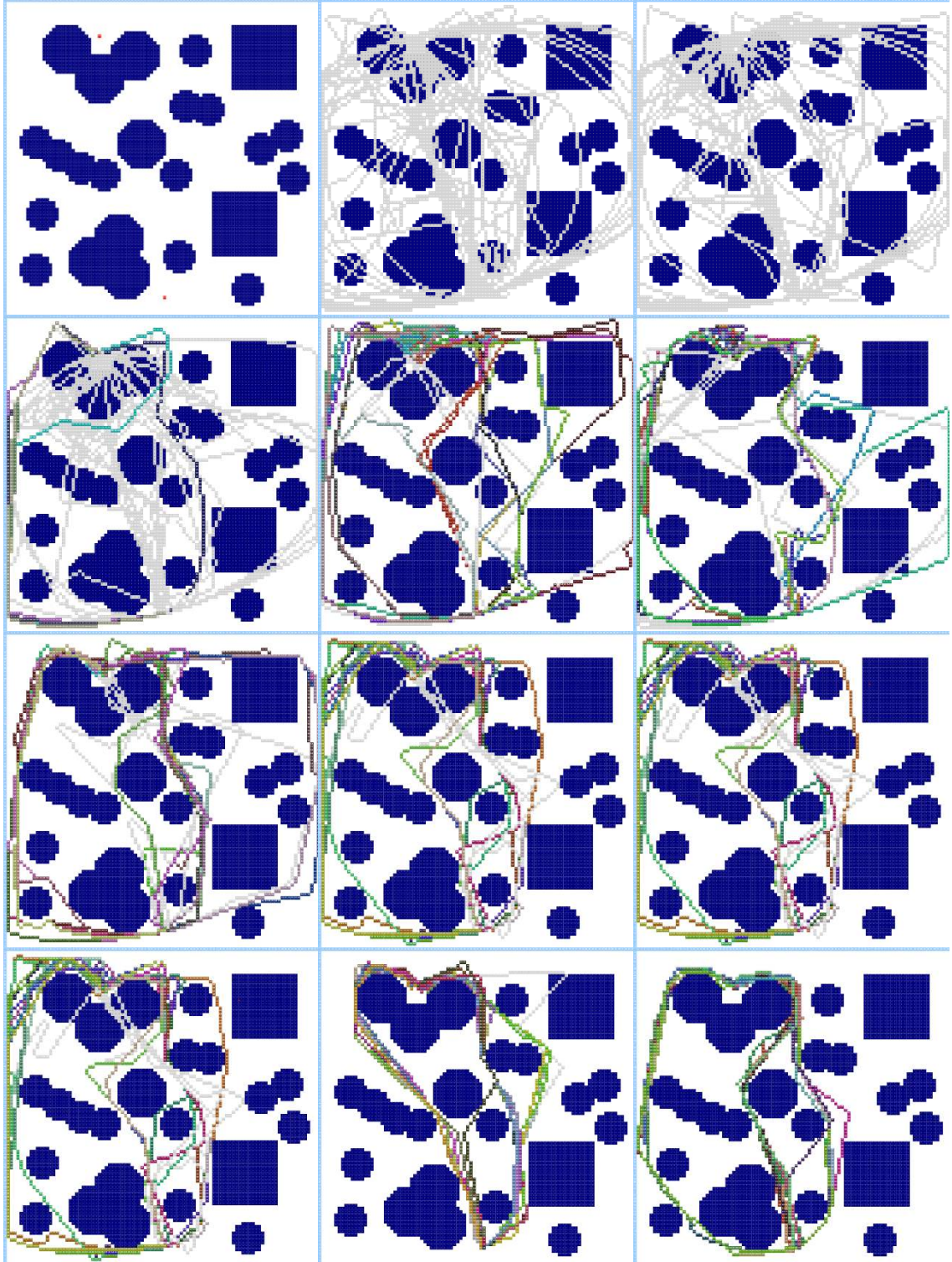


Şekil 4.13. Yinelemelere göre ortalama popülasyon uygunluk değerinin değişimi

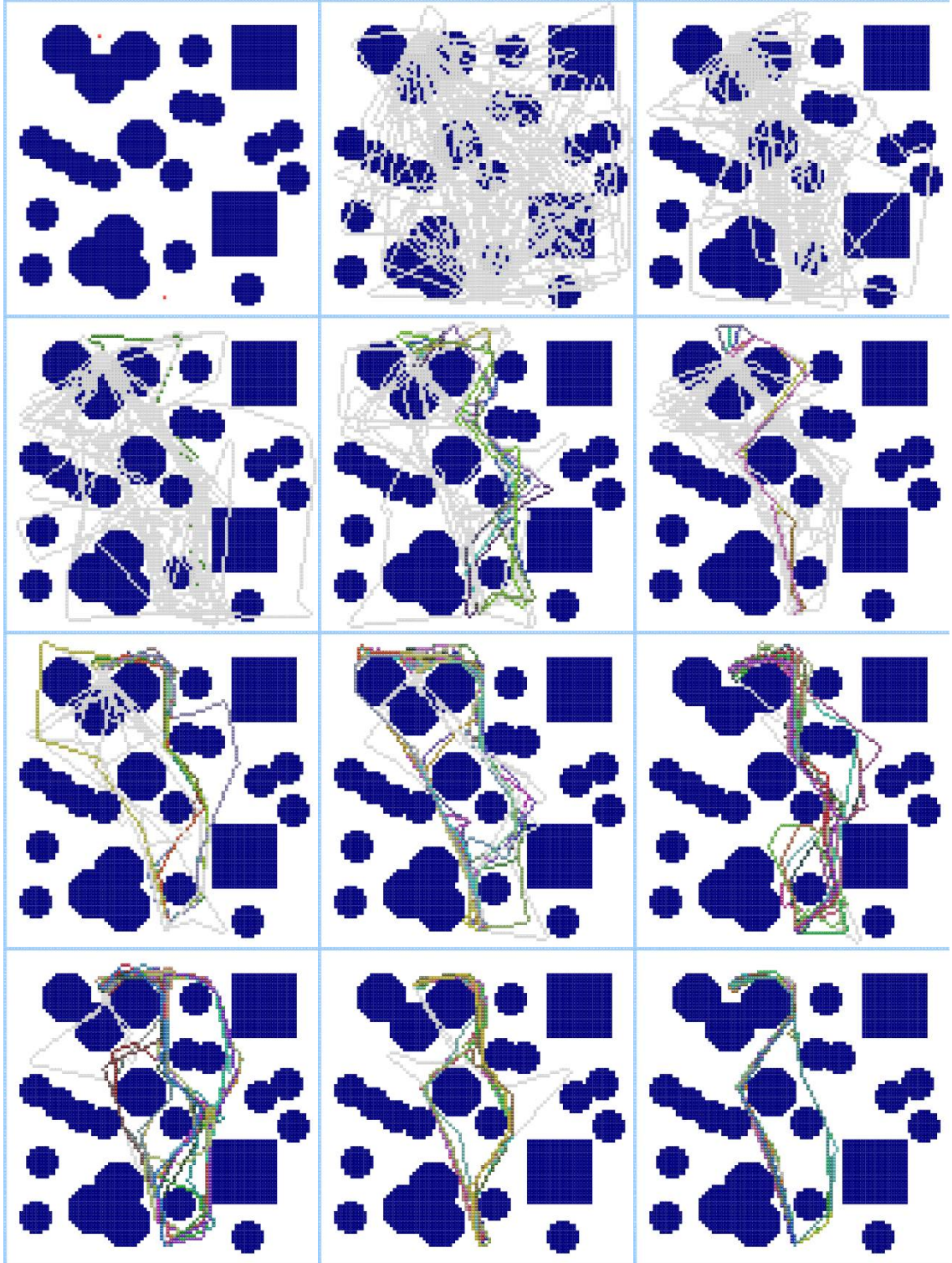


Şekil 4.14. 30 yineleme sonunda başlangıç popülasyonuna göre iyileşme oranları

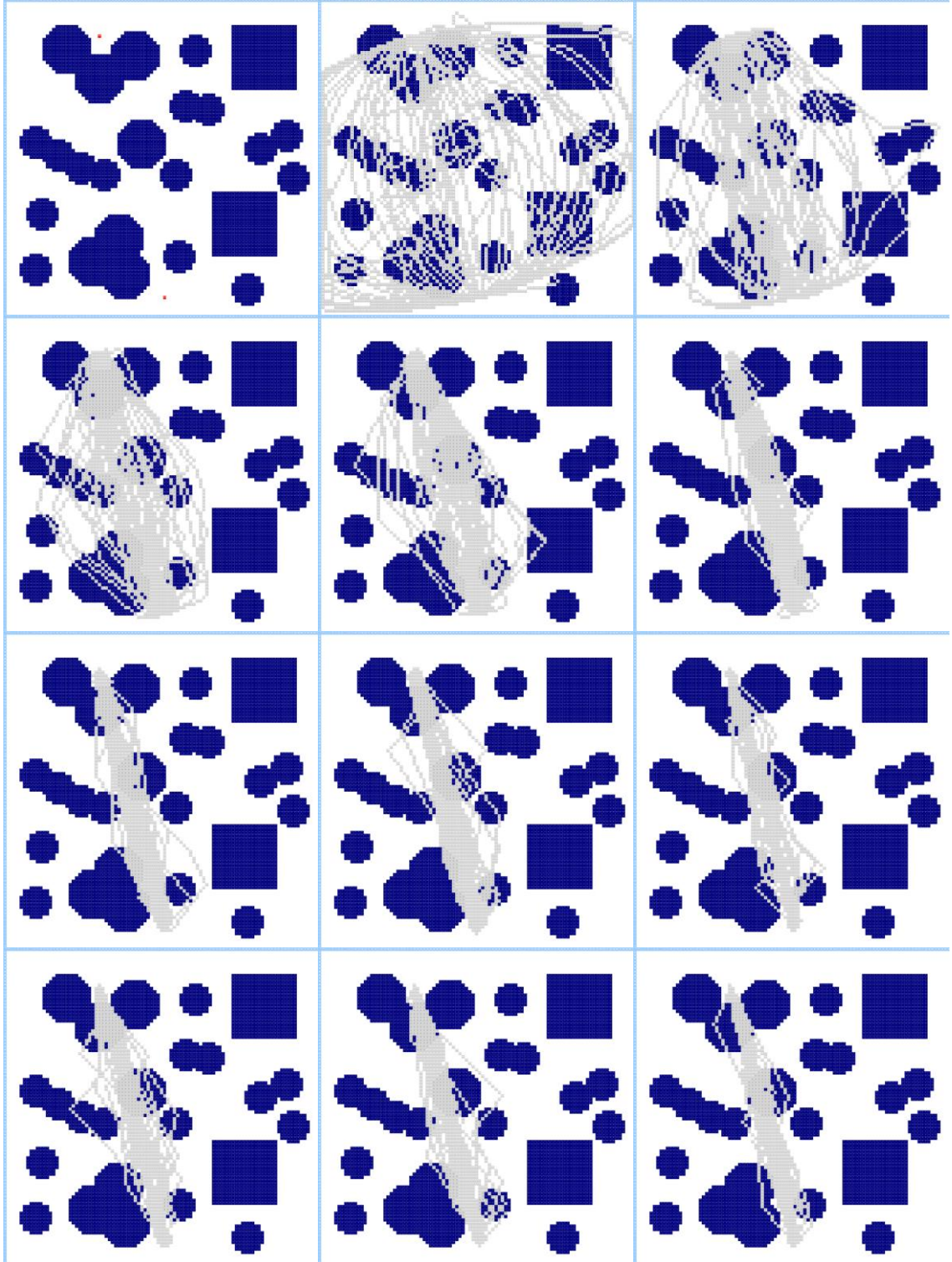
scGA, en iyi çözümü 3,2 kat, popülasyonu 9,8 kat iyileştirirken; scGAR, 3,4 ve 10,7; cGA, 1,9 ve 1,9; cGAR 2,0 ve 1,6; GA ise 1,7 ve 1,7 kat iyileştirme sağlayabilmiştir. scGA(R) algoritmaları açık ara önde bir başarı elde etmiştir. Sabit başlangıç popülasyonunun etkisiyle, scGA daha iyi bir uygunluk oranı ile yinelemelere başladığı için scGAR'nin iyileştirme payı daha yüksek olmuştur.



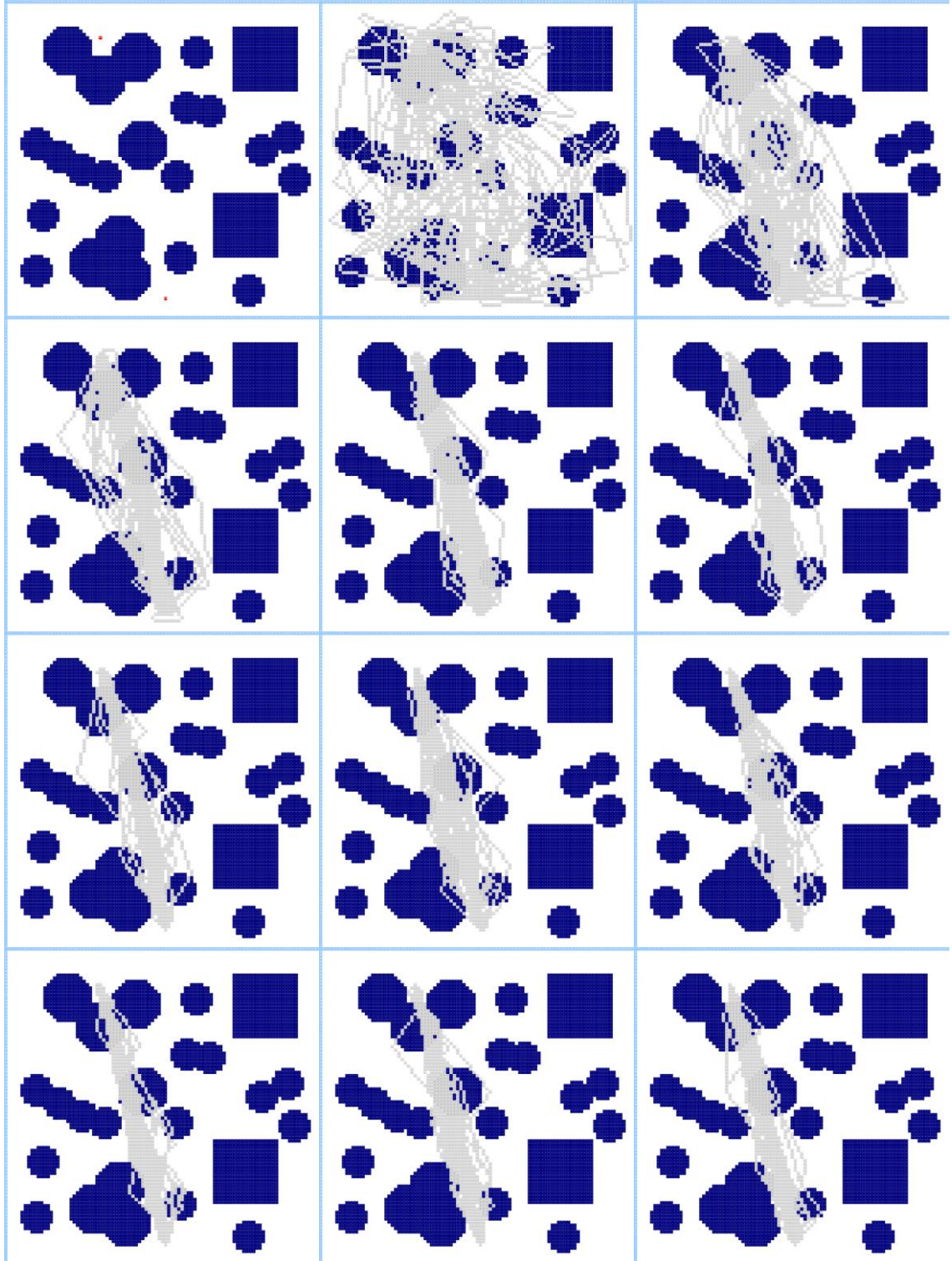
Şekil 4.15. Görece zor bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda scGA tarafından üretilen yolların gösterimi



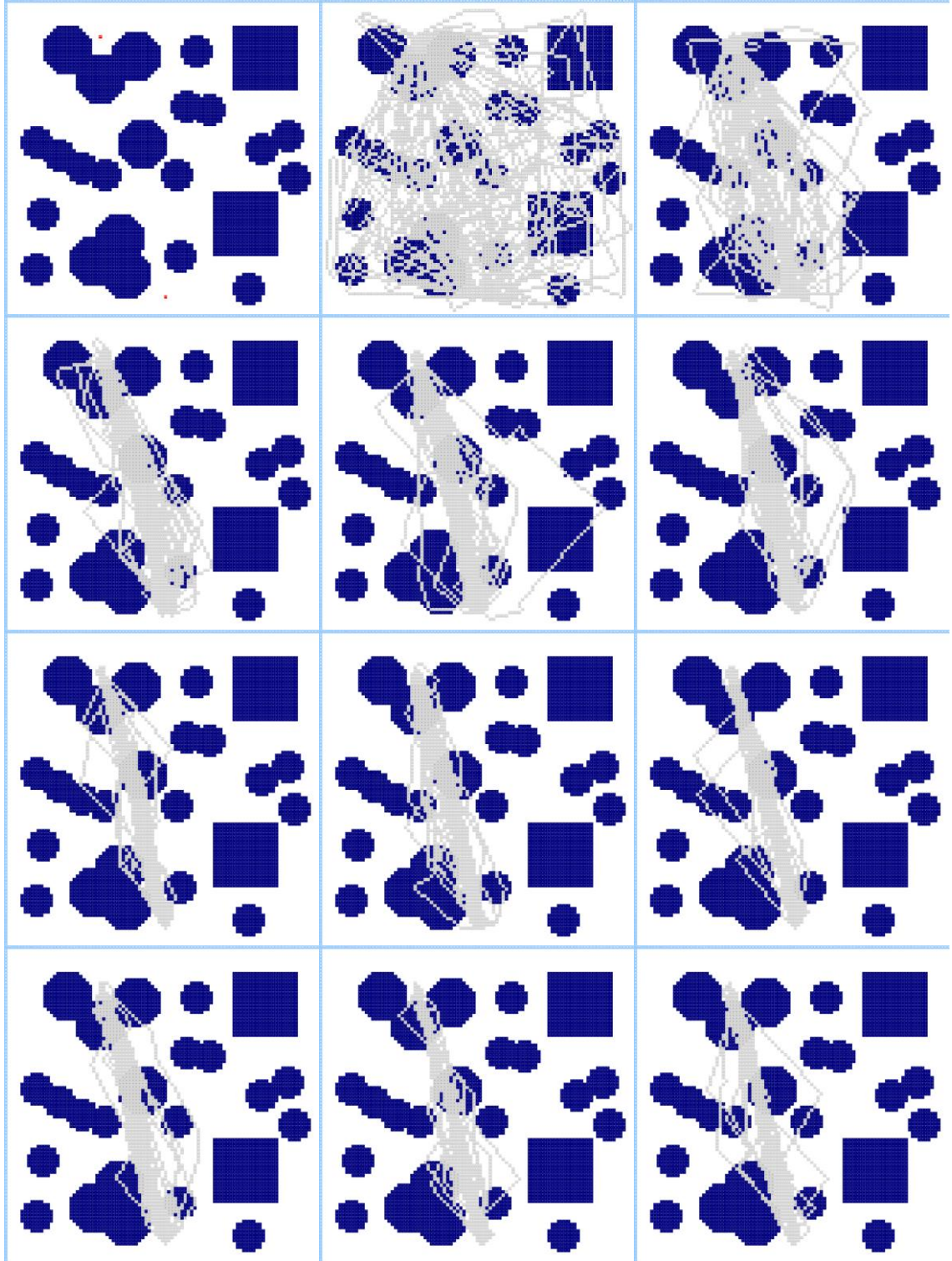
Şekil 4.16. Görece zor bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda scGAR tarafından üretilen yolların gösterimi



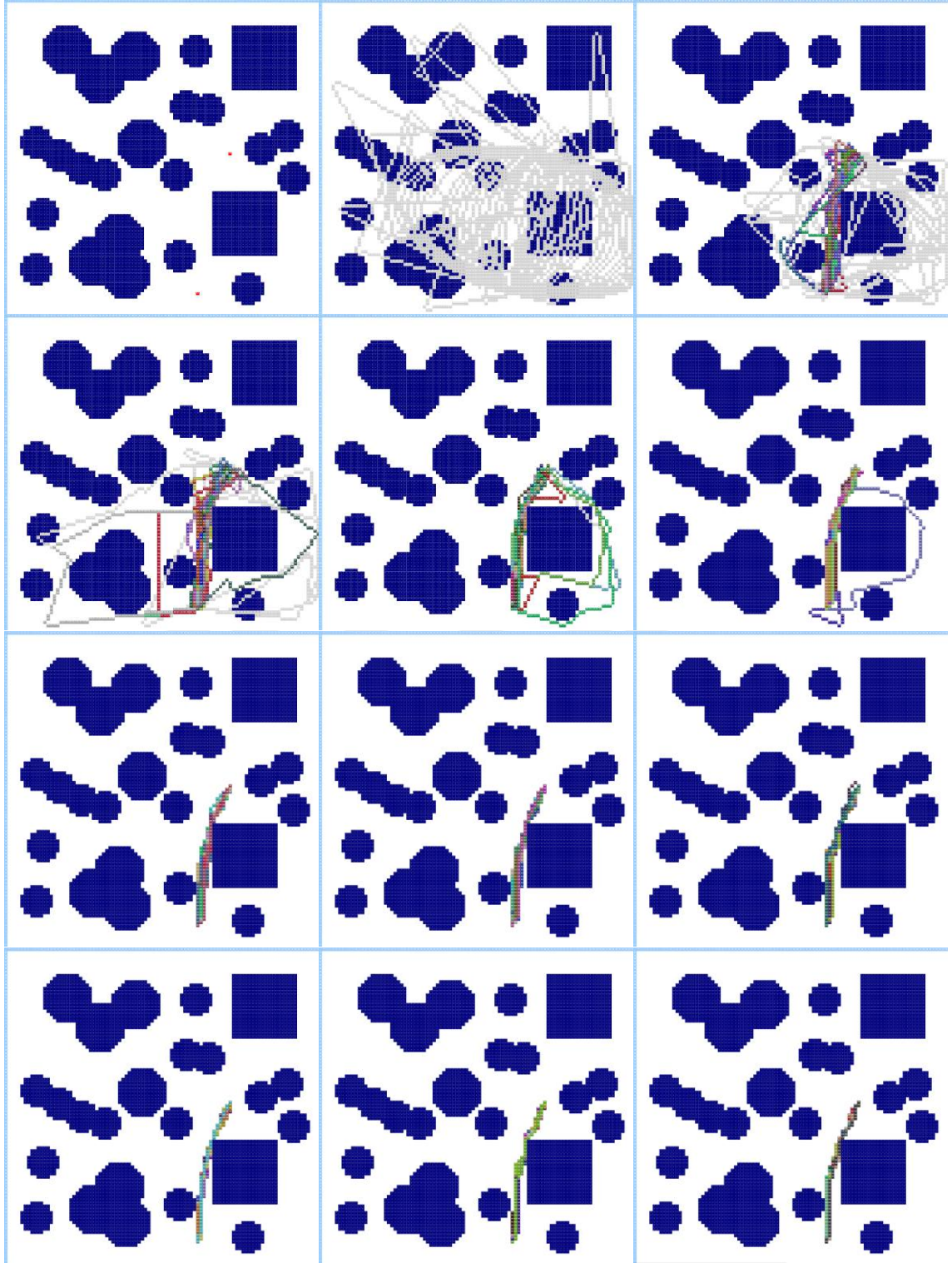
Şekil 4.17. Görece zor bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda cGA tarafından üretilen yolların gösterimi



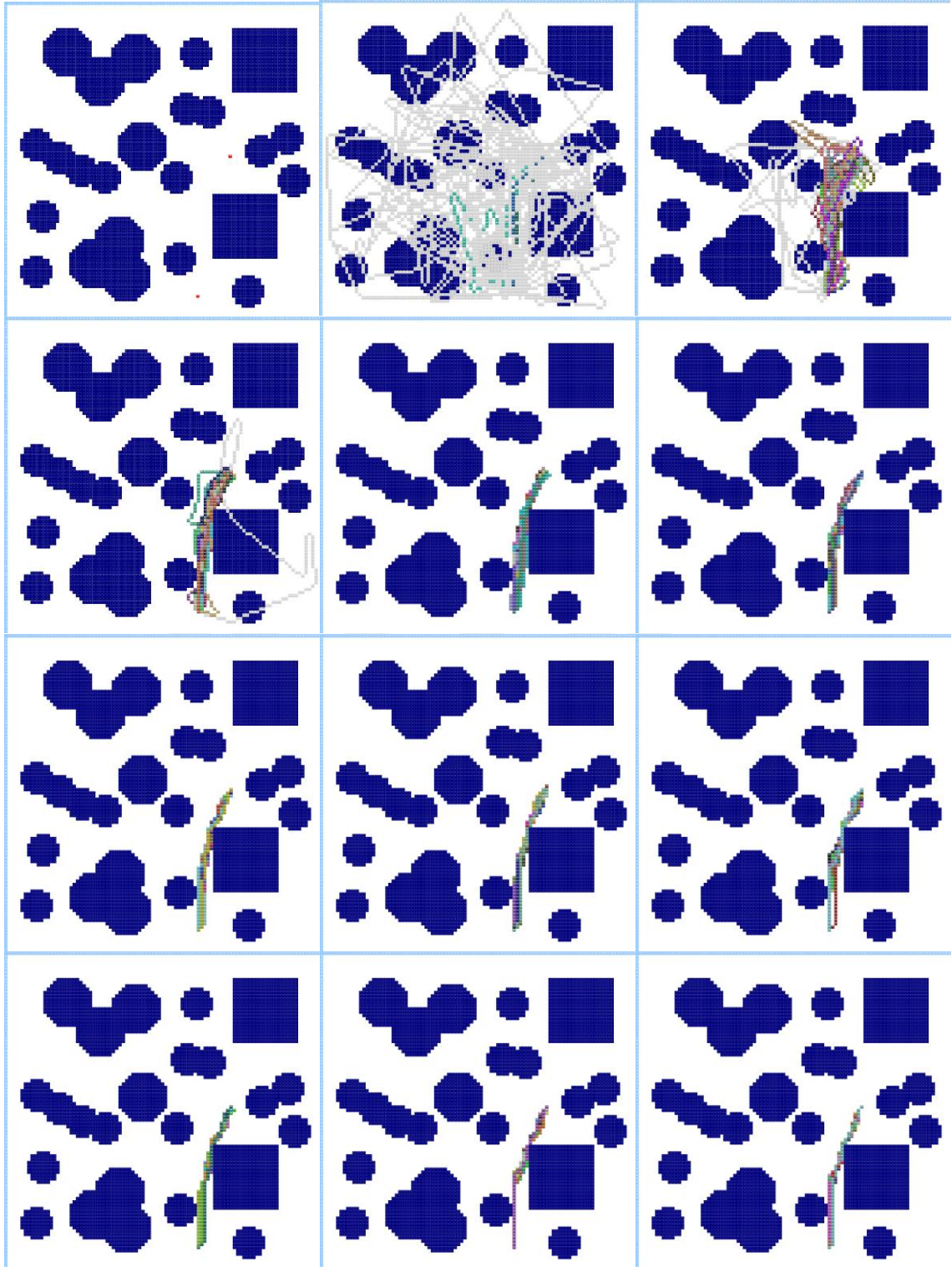
Şekil 4.18. Görece zor bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda cGAR tarafından üretilen yolların gösterimi



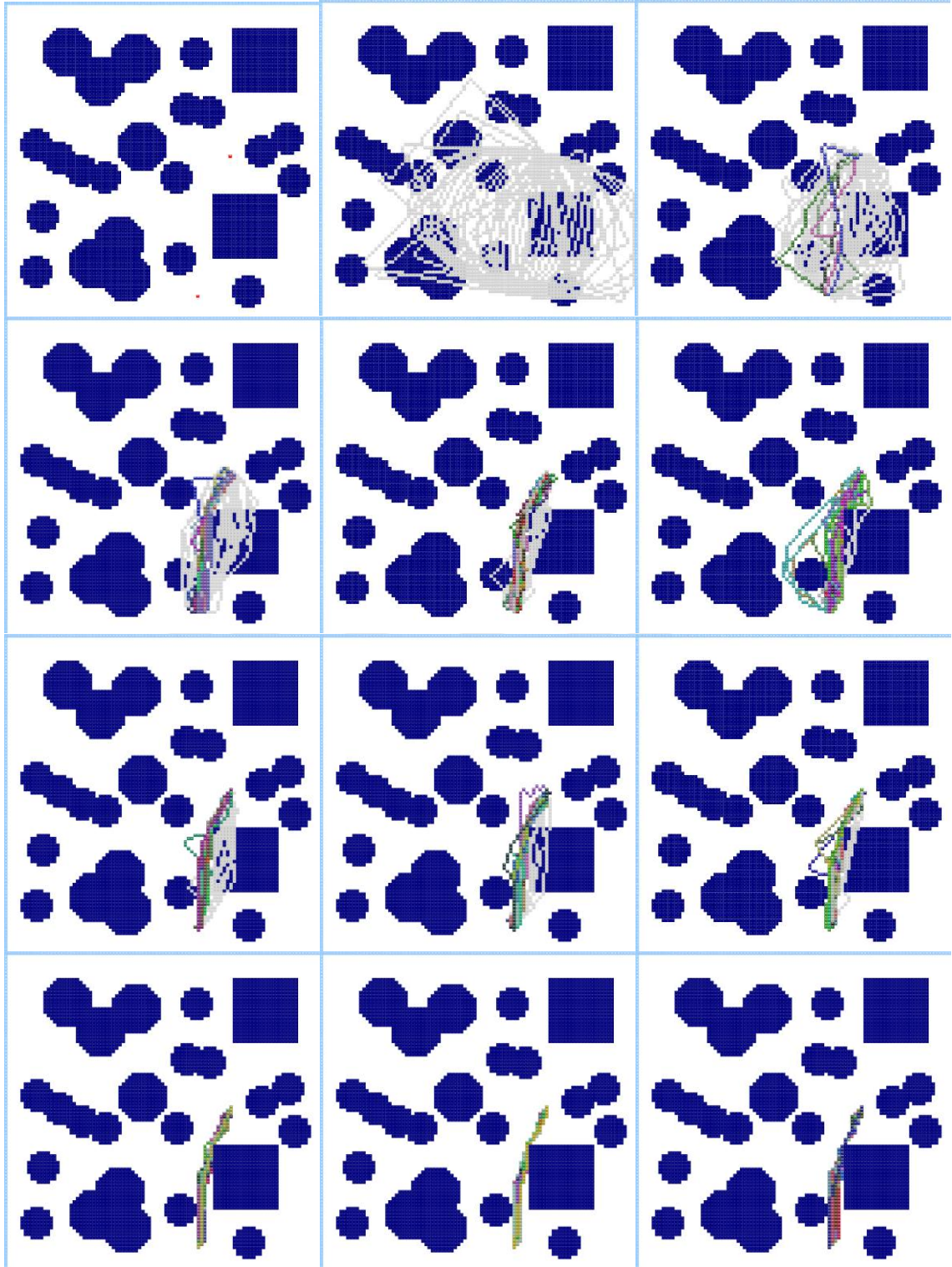
Şekil 4.19. Görece zor bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda GA tarafından üretilen yolların gösterimi



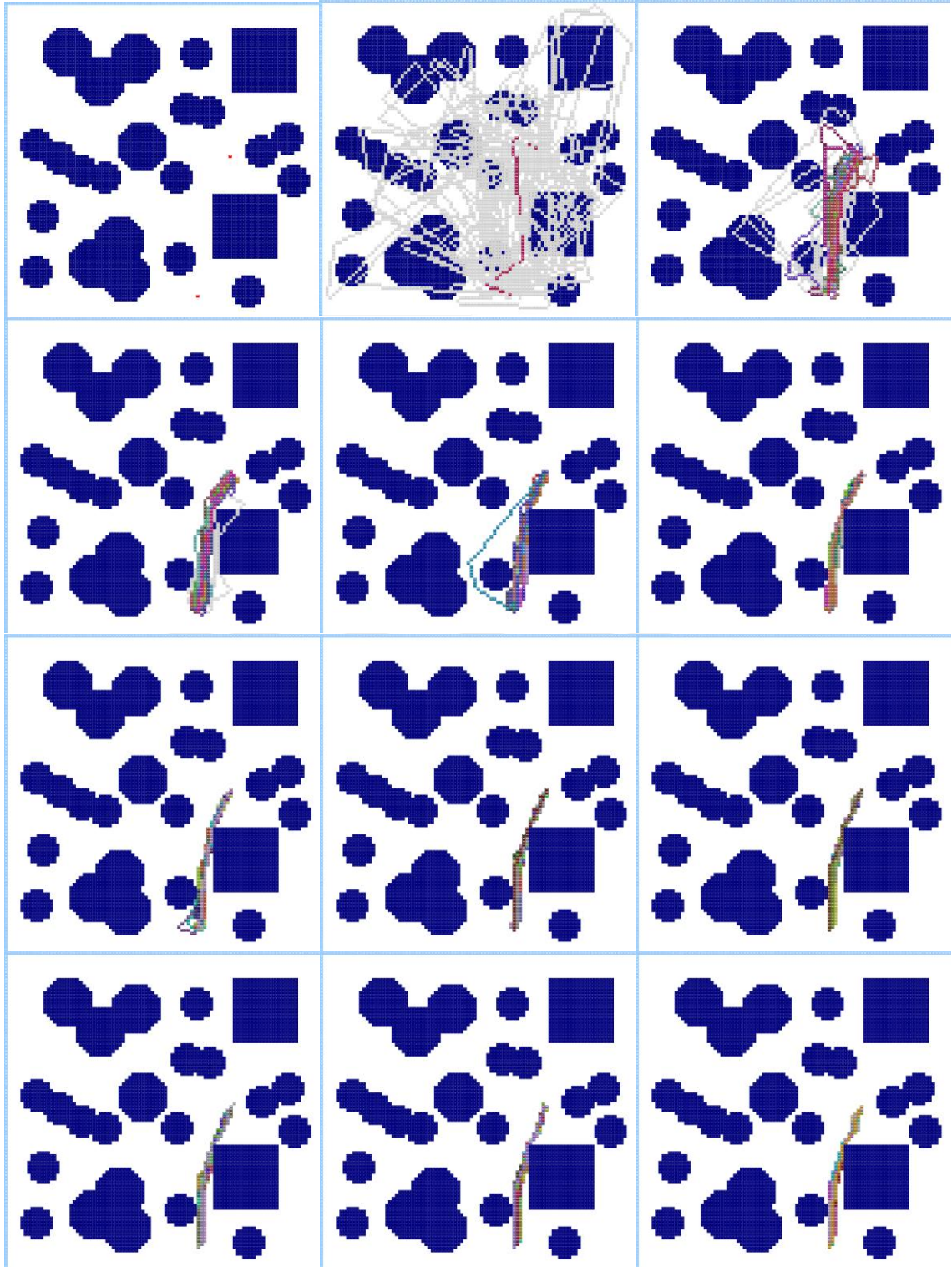
Şekil 4.20. Görece kolay bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda scGA tarafından üretilen yolların gösterimi



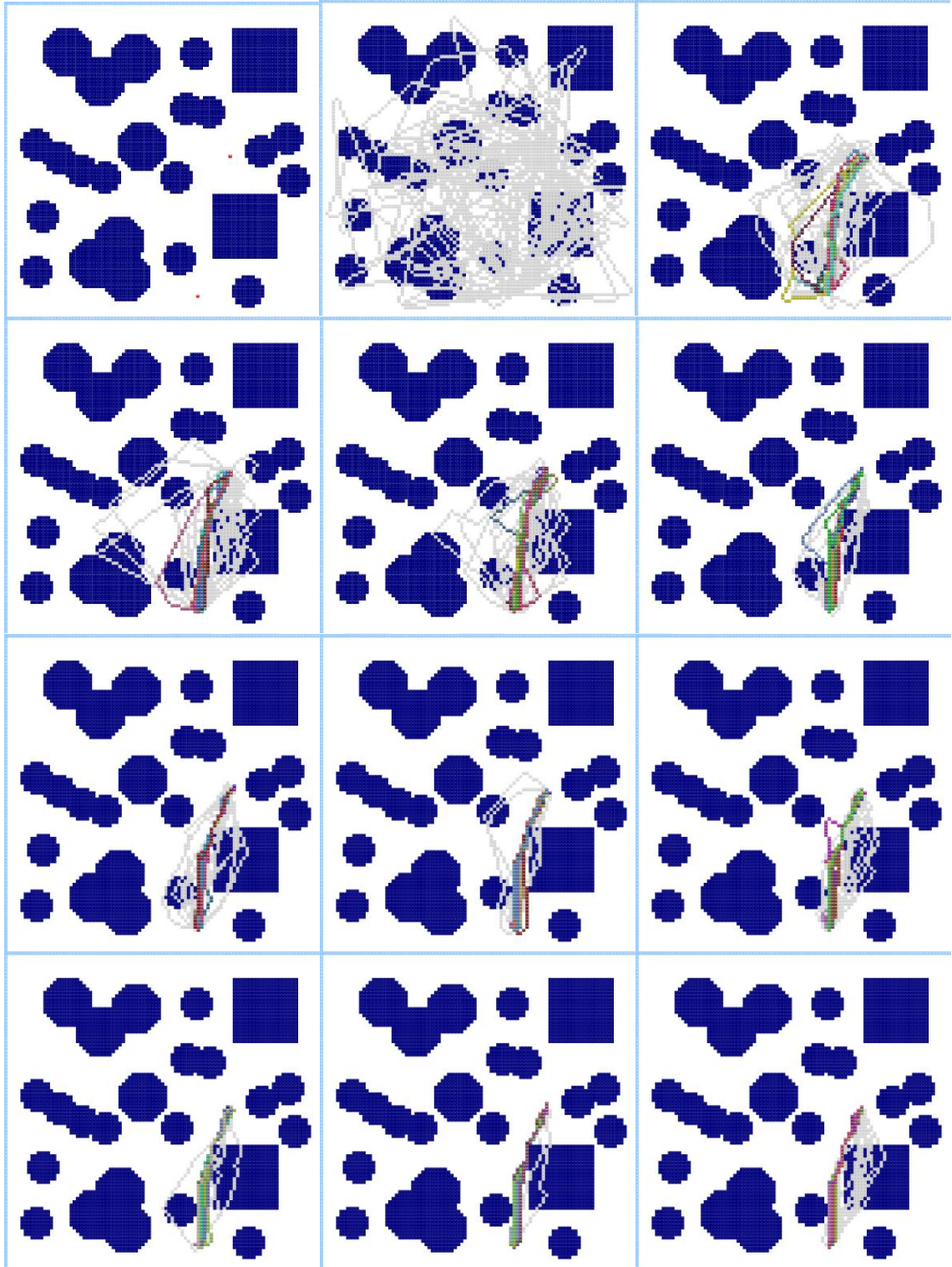
Şekil 4.21. Görece kolay bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda scGAR tarafından üretilen yolların gösterimi



Şekil 4.22. Görece kolay bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda cGA tarafından üretilen yolların gösterimi



Şekil 4.23. Görece kolay bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda cGAR tarafından üretilen yolların gösterimi



Şekil 4.24. Görece kolay bir senaryoda; (soldan sağa) 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yinelemeli çalıştırmalarda GA tarafından üretilen yolların gösterimi

Popülasyonun gelişim davranışı her algoritmada farklıdır. Bu gelişim, görece kolay ve zor iki farklı senaryoda karşılaştırılmıştır. Testler; 1, 10, 20, 30, 40, 50, 60, 70, 80, 90 ve 100 yinelemeli olarak ayrı ayrı çalıştırılmış ve her çalıştırma sonucunda üretilen yollar

(Bkz. Şekil 4.15-4.24) sunulmuştur. Gri yollar, geçersiz yolları; renkli yollar ise geçerli yolları temsil etmektedir. scGA(R)'nin, popülasyon çeşitliliğini (yollar geçersizken dahi) koruduğu, diğer algoritmaların ise popülasyon içindeki en iyi bireye hızla yakınsadığı görülmektedir.



5. SONUÇ VE ÖNERİLER

Kısaca yol planlama, bir başlangıç noktasından varış noktasına güvenli bir yol bulma probleminin çözümüdür. Yol planlaması için hesaplama verimliliği, yol optimalliliği ve yol bütünlüğü olarak üç kriter olduğu söylenebilir. Yol planlama algoritması tasarımı bu kriterler dikkate alınmaktadır.

Bu bölümde, yukarıda belirtilen kriterler göz önünde bulundurularak yol planlama yazılım paketi de dâhil olmak üzere önerilen yol planlama algoritmaları tartışılmakta ve sonuçlandırılmaktadır. Bu tez çalışmasının muhtemel gelişim potansiyeli değerlendirilmektedir.

Önerilen yol planlama algoritmalarının bir İHA için özelleştirilmiş, performans özellikleri temelinde, kinematik kısıtlara uyumlu, ekonomik ve havada kalma süresini uzatan yollar üretilmesinde gösterdikleri başarı ile orantılı olarak; yakıt sarfiyatının azaltılması, potansiyel riskleri azaltma, azami operasyon süresi limitinin uzaması ile operasyon başarısının yükseltilmesi ve aracın yaşam döngüsünü uzatma gibi katkılar sağlaması beklenmektedir. Bu doğrultuda daha az yakıt tüketimi ve karbon emisyonunun azaltılması ile çevresel kazanım sağlanmaktadır. Hesaplama maliyetinde verimlilik gerçek zamanlı çalışmaya imkân tanımaktadır.

Bu bölüm, spesifik olarak, meta-sezgisel arama stratejisini belirleyen ve genetik çaprazlama operatörünün kapsamını kısıtlayan algoritmanın, popülasyonun bir ızgara düzenlemesine bağlı bireylerinin komşuluk ilişkisi içerisindeki etkinliğini açıklamaktadır. Başlangıç popülasyonunda yapılan şekillendirmenin, parçalı kromozom yapısının, aşırı korumacı yaklaşımın ve farklı mahalle yapılarının verimliliğe etkisi gözlenmiştir. scGA ve cGA'nın umut verici test sonuçları göz önüne alındığında, scGA, evrimsel bir perspektiften ele alınabilecek yol planlama optimizasyon problemleri için uygun bir yöntem oluşturmaktadır.

Geleneksel GA ile önerilen algoritmalar arasında değerlendirmeye makul bir temel sağlamak üzere bir dizi test yapılmıştır. Testlerde hem GA hem de scGA ve cGA varyantları kullanılmıştır. GA'nın stokastik doğası, karşılaştırmalı çalışmayı zorlaştırır. Stokastik etki ile oluşan örneklem etkisini en aza indirmek için her senaryo, rastgeleliği yeniden başlatarak belirli sayıda tekrarlanmıştır. 10 rastgele rotada 50 testin sonuçlarının

ortalaması alınmıştır. Farklı yöntemlerden elde edilen sonuçların kendi aralarında uyumlu olduğu ve bir yargıya ulaşmak için yeterli olduğu değerlendirilmiştir.

Temelde üç performans ölçütü dikkate alınmıştır: yol ekonomikliği, hesaplama süresi ve başarı oranı. İHA'nın uzun mesafeler kat etmesinin beklendiği ve havada kalma süresinin daha öncelikli olduğu durumlarda, İHA bütünleşik enerji kaynağının önemi kritiktir. İHA'nın, algılama ve aktüatör sistemleri ile paylaşılan kısıtlı yerleşik enerji kaynaklarının, yol planlama için de kullanıldığı durumlarda bu performans parametresi, daha önemli bir hâl almaktadır. Test sonuçları, yol planlama algoritmalarının güvenilirliğini ve sağlamlığını ölçerken gerçek uygulamalara yaygınlaştırılma potansiyelini belirlemektedir.

Bir algoritmanın olabilecek bütün problemler üzerinde diğerlerinden daha başarılı bir performans sağlayacağını belirtmek gerçekçi değildir. Yol planlama problemleri açısından bir değerlendirmede bulunmak daha doğru olmaktadır.

Bu çalışmada, yol planlama problemlerini çözmek için yeni bir evrimsel genetik yaklaşım tanıtılmıştır. Önerilen algoritma, sabit bir başlangıç popülasyonunun parçalı kromozom temsili ile oluşturulan popülasyon çeşitliliğini destekleyen hava aracı performansını temel alarak yakıt ekonomisi sağlayan aşırı korumacı parçalı hücresel genetik bir algoritmadır. Algoritmanın İHA yol planlaması problemine uygulanmasının ana sonuçları aşağıda değerlendirilmiştir.

- 1) cGA, İHA yol planlama problemine uygulanmıştır.
- 2) cGA yöntemi kullanılarak, bir ızgara grafiğinde en ekonomik yolları bulmak için algoritmanın yeni bir varyantı olan sabit başlangıç popülasyonu ve parçalı kromozom yapısına sahip scGA geliştirilmiştir.
- 3) Geliştirilen yazılım ile, yol planlama parametreleri ve yöntem seçilerek testler yapılmıştır.
- 4) Üretilen yollar FlightGear üzerinde simülasyon ile gösterilmiştir.
- 5) Üretilen yol ve uçuş esnasında hava aracının izlediği yol, GoogleEarth üzerinde gösterilmiştir.

Sonuç olarak, önerilen yaklaşım ile algoritmanın seçim baskısında ve yakınsama hızında bir iyileşme sağlandığı gözlenmiştir. Test sonuçları, İHA toplam yakıt tüketiminin optimize edildiği bir performansı ortaya koymaktadır.

Önerilen Hücresel Genetik Algoritmalar (cGA ve scGA), mahalle seçiminin oluşturduğu etkiyi anlamak için farklı mahalle boyutlarında test edilmiştir. Örtüşen iki mahalle için aynı iki ebeveyni seçme ihtimalinin olmadığı doğrusal mahallelere odaklanılmıştır. Mahallelerin performansı karşılaştırıldığında bazı farklılıklar gözlenmiştir.

Tek süreçli testlerde scGA için 6 hücre genişliğindeki L6 mahallesinin test edilen diğer mahalle yapılarından daha iyi bir performans göstermiştir. Aslında 4, 5, 6, 7, 8 ve 9 genişliğinde mahalleler benzer sonuçlar vermektedir. Çok süreçli testlerde scGA için L8 mahalle yapısı en iyi sonucu elde etmiştir. Burada da benzer şekilde 5, 6, 7, 8 ve 9 genişliğindeki mahallelerin sonuçları birbirine yakındır. Genel olarak scGA'da geniş mahalle yapısı başarıyı artırmaktadır. Özellikle L6, L7 ve L8 tercih edilebilir. cGA için L2, cGA-Ç için L9 mahallesinin test edilen diğer mahalle yapılarından daha iyi bir performans göstermiştir. Genel olarak mahallelerin genişlemesiyle son nüfus ortalama uygunluk değerinin azaldığı görülmektedir.

Bölüm 3.3.1'de açıklandığı gibi, çoklu işlem algoritmalarında süreçler her yineleme tamamlandığında kendi popülasyonunu veritabanına yüklemekte ve komşu popülasyonları okumaktadır. Ancak yinelemeler aynı anda tamamlanmadığından güncellemeler farklı yineleme zamanlarında erişilebilir olurken bu durum, çok süreçli ve tek süreçli algoritmalarda test sonuçlarına da yansıyan bir farklılığa sebep olmaktadır.

Geçersiz sonuç bulma yüzdelere göre algoritmalar karşılaştırıldığında görece daha sade olan Harita-2 senaryosu hariç diğer senaryolarda cGA(R) ve GA algoritmalarının geçersizlik yüzdesi oldukça yüksek bir orandadır. scGA ise tüm harita senaryolarında olağanüstü bir başarı göstermiştir. Sadece tek süreçli testlerde L2, L3, L4 ve L5; çoklu işlem senaryolarında L4 ve L8 mahallelerinde %2 gibi düşük bir oran ile geçersiz yollar görülmüştür. Diğer tüm mahalle, harita ve senaryolarda %0 geçersiz sonuç başarısı ile açık ara daha iyi bir performans sergilemiştir. scGAR'da %0 ila %8; cGA'da %0 ila %46; cGAR'de %0 ila %58; GA'da %6 ila %50 başarısız yollar üretilmiştir. Haritalardaki zorluğa paralel olarak geçersiz sonuç üretme yüzdesi artmıştır.

Rastgele başlangıç popülasyonu kullanan cGAR algoritmasının başarısı cGA'ya ve diğer algoritmalara kıyasla oldukça düşüktür. Bu durum sabit başlangıç popülasyonun bir başarısı ve standart cGA'nın ancak önerilen sabit başlangıç popülasyonu ile istenen

başarıyı yakalayabileceğinin bir göstergesi olarak değerlendirilebilir. scGAR’de ise durum çok farklıdır. Parçalı kromozom yapısına özelleştirilmiş çaprazlama ve mutasyon stratejisi ve aşırı korumacı yaklaşım sayesinde scGAR üstün bir performans sergileyebilmiştir. scGA sabit başlangıç popülasyonunun da etkisi ile daha yüksek bir başarı yakalamıştır.

GA, cGA, scGA ve varyantlarının hesaplama verimliliği karşılaştırılmıştır. Farklı karmaşıklıkta haritalar için algoritmaları karşılaştırmak, deneysel değerlendirmemizin ana hedeflerinden biridir. Bölüm 4’te sunulan test sonuçlarında görüldüğü üzere scGA(R)’nin diğer algoritmalarından açıkça daha iyi bir performans sergilediği anlaşılmıştır. Yakınsama ile ilgili sonuçlar dikkat çekicidir. GA’da 30 nesilden sonra popülasyon büyük oranda devralınmış olsa da, cGA ve scGA(R)’nin 100 neslinde dahi popülasyon çeşitliliğini devam ettirdiği fark edilebilir. Görüldüğü gibi, cGA ve scGA(R), uzun nesiller büyük keşif özelliklerini korur ve algoritma çalışmasını tamamladığında GA’ya göre çok daha fazla Pareto cephe çözümü üretir ve bu açıdan daha verimlidir. scGA(R)’nin popülasyonda çeşitliliğin istenen şekilde korunmasına izin veren bir yakınsama elde edebilmekte olduğu ve hesaplama tamamlandığında aynı başlangıç ve varış noktaları arasında farklı güzergâhlarda geçerli alternatif yolların korunduğu görülebilmektedir. Böylece çoklu İHA yol planlamasında ihtiyaç duyulan alternatif yollar ek bir maliyet olmadan üretilebilmektedir. Sonuçlar, scGA(R)’nin keşif ve gelişim arasında uygun bir dengeyi nasıl başardığını açıkça göstermektedir.

Tablo 5.1. *Tüm harita ve mahalle yapılarının algoritmalara göre ortalamaları*

Algo.	Ort. performans dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç bulma yinleme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyum Değeri
scGA	38,56	2,92	0,49	2,02	0,13	1,46	0,19%	8,12	467,66
scGAR	39,13	3,27	0,87	3,10	0,25	2,36	1,41%	6,96	501,56
cGA	155,53	1,92	0,75	2,08	0,11	3,14	22,03%	6,09	647,77
cGAR	218,20	1,95	0,96	2,02	0,20	2,97	30,56%	5,78	669,36
GA	200,62	1,88	0,92	2,13	0,19	3,03	28,50%	5,60	794,75

cGA, yol planlama sorunları için çözüm çeşitliliğine ve yüksek hıza sahiptir. cGA, GA ile karşılaştırıldığında, ilk uygun çözüme daha hızlı ulaşmıştır. Bu açıdan scGA daha iyi sonuçlar vermiştir. En iyi çözümün geliştirilmesinde scGA(R) ve cGA'nın, GA'ya kıyasla oldukça başarılı olduğu görülmektedir.

Tek süreçli uygulamada hesaplama süresi gerçek zamanlı uygulamalar için nispeten uzundur. Çözümü hızlandırmak için geliştirilen çok süreçli algoritmalar ile yapılan testlerde hesaplama süresinin; GA'da %66, cGA(R)'de %48 scGA(R)'de %58 oranında azaldığı not edilmiştir.

Testlerin, sadece 4 gerçek ve 4 sanal süreç desteği olan mütevazı bir işlemcide yapılmış olması çoklu işlem çözüm süresinin sınırlayıcı faktörüdür. (Ryzen 2400G CPU, tez hazırlandığı dönemde piyasadaki çoklu işlemde en iyi işlemcilere kıyasla masaüstünde %425, iş istasyonunda %1525 daha düşük performansa sahiptir [359, 360].) PHP dilinin çoklu işlem desteğine sahip olmaması ve çoklu işlemin, veritabanı kullanımı ile sağlanmasının, çoklu işlem performans artışının beklenenden az olmasında en önemli etken olduğu değerlendirilmektedir. cGA(R) ve scGA(R) veritabanı üzerinde uygulanan çoklu işlem mimarisinde mahalleler arası veri alışverişi ile başarısını artırmış ancak bu durum bir hesaplama yükü oluşturmuştur buna karşın GA'da böyle bir işlemin olmaması test sonuçlarında gözlenmiştir. GA, çok süreçli testlerde geçerli yol üretmede yeterli başarıyı ortaya koyamamıştır. Ancak geçersiz çözümlerin artmasıyla hesaplamayı tamamlama hızında bir avantaj yakalamıştır.

Tablo 5.2. Algoritmaların geçerli ve geçersiz sonuç üretmedeki hesaplama süreleri

	Ortalama geçerli yol hesaplama süresi (s)	Ortalama geçersiz yol hesaplama süresi (s)	Süre değişim oranı
scGA	2,41	3,21	%133,5
cGA	3,23	1,93	%59,6
scGAR	3,84	3,53	%92,0
cGAR	2,93	1,90	%64,9
GA	3,01	1,99	%66,2
Ortalama	3,08	2,51	%83,2

scGA(R), neredeyse tüm senaryolar için hem uygunluk değerinde hem de hesaplama süresinde daha iyi performans göstermiştir (Bkz. Tablo 5.1). Hesaplama süresinde GA'nın önde olduğu görülmektedir. Ancak geçersiz sonuçların hesaplama hızı GA'da %66,2 daha kısadır (Bkz. Tablo 5.2). GA'nın ürettiği yolların da %28,50'si

geçersizdir. Geçerli ve geçersiz yolların üretilmesindeki hız farkı genellikle hatalı yakınsamadır. Genellikle, geçersiz en iyi yol, iki nokta arasındaki düz çizgidir (Bkz. Şekil 15-24). Bu nedenle geçersiz yoğunluklu popülasyon genelinde yollar kısalmakta ve hesaplama süresi büyük oranda düşmektedir. scGA’da durum tam tersidir. Popülasyon genelindeki çeşitlilik çalışma süresinin sonuna kadar korunur ve geçerli bir aday çözüme yakınsama gerçekleşmediği için harita genelindeki (genellikle uzun) yollar popülasyon içinde yer almaya devam etmektedir. Bu yollar elenmez çünkü geçerli bir çözüm olma potansiyelleri bulunmaktadır. Aslında geçersiz sonuç yüzdesine bakıldığında, scGA’nın neredeyse hiç geçersiz sonuç üretmemesi bu potansiyelin gerçekleştiğini gösterir. Ancak belirtmek gerekir ki hesaplanan %133,5 oranı gerçekçi değildir. Bu derecede yüksek bir değer bulunmasının nedeni, scGA’nın hesapladığı geçersiz yol sayısının çok az olması ve bu senaryoların genel olarak zor ve çözüm sürelerinin yüksek olmasıdır. Ancak scGAR’de değerlendirme yapmaya yeterli sayıda veri bulunduğundan scGA’nın %92 oranına yakın bir değerde olduğu değerlendirilebilir.

Simülasyon sonuçlarında gösterildiği gibi, önerilen scGA algoritması İHA yol planlamada dikkat çekici bir performans sağlamaktadır. scGA’da uygulanan parçalı kromozom yapısı ve aşırı korumacı yaklaşım bu başarıya katkı sağlamıştır.

Test sonuçlarında dikkat çeken bir diğer nokta da scGA(R) ve cGA(R)’nin ürettiği ortalama alternatif yol çeşitliliğinin geleneksel algoritmaya kıyasla yaklaşık 1,5 kat daha fazla olmasıdır. Son nesilde kaliteli yolların farklı güzergahlar içermesi bir avantajdır.

Farklı senaryoları değerlendirmek için çeşitli konfigürasyonlarda testler gerçekleştirilmiştir. İHA’ya özgü bir model kullanıldığından planlanan yol, aracın dinamiklerine uygun optimale yakın bir yoldur. Yol planlamasının uygulanabilirliği, FlightGear üzerinde simüle edilmiş; üretilen yolların geçerliliği test edilmiştir. Simülasyon ortamında, İHA herhangi bir çarpışma olmadan algoritmalar tarafından üretilen yollar üzerinde başarılı bir şekilde varış noktasına ulaşmaktadır. Sonuç olarak, İHA’lar için önerilen yol planlama yönteminin çarpışmayı önlediği, araç güvenliğini sağladığı, havada kalma süresini uzattığı, güvenli ve ekonomik yollar oluşturmada hızlı ve etkili olduğu gösterilmiştir.

Tez çalışmasının geneli değerlendirildiğinde sonraki çalışmalar için potansiyel araştırma önerileri şu şekilde açıklanabilir:

- Önerilen algoritmada çözüm süresini iyileştirmek adına bazı durumlar ihmal edilmiştir. Kullanılan verilerin kapsamını ve doğruluğunu en üst seviyeye çıkarmak ve böylece yol planlamayı kusursuz bir seviyeye taşımak için İHA'ya ait bazı modellerin oluşturulması tercih edilir:
 - Hız modeli,
 - Yakıt modeli,
 - Tırmanma oranı ve alçalma oranı modelleri,
 - Tırmanış programına göre model,
 - Eğrisel yörünge için yakıt modeli.
- Oluşturulacak modellerin amaç fonksiyonu içerisinde doğrudan kullanılmasının çözüm süresine ve çözüm kalitesine etkisi karşılaştırılabilir.
- Uygulanan algoritmalar, tüm engellerin statik olduğunu varsaymaktadır. Algoritmalar hareketli engelleri dikkate alacak şekilde düzenlenebilir.
- Uygulanan algoritmalarda yalnızca Aerosonde model bir İHA'nın verileri kullanılmıştır. Farklı tür İHA'larda algoritmaların başarısı sınanabilir.
- Uygulama, tekil İHA senaryoları ile sınanmıştır. Çoklu İHA tarafından yürütülen görevler açısından optimize edilmesi önemli bir geliştirme alanıdır.
- Kullanılan işlemci performansı, tez hazırlandığı dönemde çoklu işlemde CPU piyasasındaki en iyi işlemcilere kıyasla masaüstünde %425, iş istasyonunda %1525 daha düşük performansa sahip olduğundan testlerin daha hızlı işlemciler üzerinde gerçekleştirilmesi tercih edilebilir [359, 360].
- Test edilen tüm algoritmalar CPU üzerinde uygulanmıştır. GPU'lar üzerinde çalışabilecek şekilde düzenlenebilir ve daha yüksek bir performansa ulaşılabilir.
- Algoritmanın tez uygulamasında kullanılan bir betik dilden ziyade C/C++ gibi daha düşük seviyeli bir dil ile yazılması hesaplama süresini önemli ölçüde azaltabilir.
- GA için tasarlanmış farklı teknikler, cGA ve scGA'ya uygulanabilir.
- Önerilen algoritmalar yol planlamadan farklı çeşitli optimizasyon problemlerine uygulanabilir.

KAYNAKÇA

- [1] Kamal, W.A. (2005). *Safe trajectory planning techniques for autonomous air vehicles*. Leicester: University of Leicester.
- [2] Clarke, R. ve Moses, L.B. (2014). *The regulation of civilian drones' impacts on public safety*. Computer Law and Security Review (CLSR), 30(3), 263-285.
- [3] <https://www.marketsandmarkets.com/Market-Reports/unmanned-aerial-vehicles-uav-market-662.html> (Erişim tarihi: 24.04.2022)
- [4] Mohammed, F., Idries, A., Mohamed, N., Al-Jaroodi, J. ve Jawhar, I. (2014). UAVs for smart cities: Opportunities and challenges. *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 267-273.
- [5] <https://www.bloomberg.com/press-releases/2019-04-09/drone-services-market-worth-63-6-billion-by-2025-exclusive-report-by-marketsandmarkets> (Erişim tarihi: 06.09.2019)
- [6] Valavanis, K.P. (2008). *Advances in unmanned aerial vehicles: state of the art and the road to autonomy*. Heidelberg: Springer.
- [7] Yao, P., Wang, H. ve Su, Z. (2015). *Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment*. Aerospace Science and Technology, 47, 269-279.
- [8] Jamshidi, V., Nekoukar, V. ve Refan, M.H. (2020). *Analysis of parallel genetic algorithm and parallel particle swarm optimization algorithm UAV path planning on controller area network*. Journal of Control, Automation and Electrical Systems, 31(1), 129-140.
- [9] Roberge, V., Tarbouchi, M. ve Labonte, G. (2013). *Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning*. IEEE Transactions on Industrial Informatics, 9(1), 132-141.
- [10] <http://www.af.mil/factsheets/factsheet.asp?fsID=122> (Erişim tarihi: 12.11.2012)
- [11] <http://www.af.mil/factsheets/factsheet.asp?fsID=13225> (Erişim tarihi: 12.11.2012)
- [12] Kuwata, Y., Teo, J., Karaman, S., Fiore, G., Frazzoli, E. ve How, J. (2008). Motion planning in complex environments using closed-loop prediction. *AIAA Guidance, Navigation and Control Conference and Exhibit*. 1-22.
- [13] Huang, H.-M., Pavek, K., Albus, J. ve Messina, E. (2005). Autonomy levels for unmanned systems (alfus) framework: An update. *Unmanned Ground Vehicle Technology VII*. International Society for Optics and Photonics, 439-448.
- [14] Boskovic, J.D., Prasanth, R. ve Mehra, R.K. (2002). A multilayer control architecture for unmanned aerial vehicles. *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*. IEEE, 1825-1830.
- [15] Ceccarelli, N., Enright, J.J., Frazzoli, E., Rasmussen, S.J. ve Schumacher, C.J. (2007). Micro UAV path planning for reconnaissance in wind. *2007 American Control Conference*. IEEE, 5310-5315.
- [16] Mersheeva, V. ve Friedrich, G. (2015). Multi-UAV monitoring with priorities and limited energy resources. *Proceedings of the International Conference on Automated Planning and Scheduling*.
- [17] Saeed, A.S., Younes, A.B., Cai, C. ve Cai, G. (2018). *A survey of hybrid unmanned aerial vehicles*. Progress in Aerospace Sciences, 98, 91-105.
- [18] Ma'sum, M.A., Jati, G., Arrofi, M.K., Wibowo, A., Mursanto, P. ve Jatmiko, W. (2013). Autonomous quadcopter swarm robots for object localization and tracking. *MHS2013*. IEEE, 1-6.

- [19] Lee, K.U., Kim, H.S., Park, J.B. ve Choi, Y.H. (2012). Hovering control of a quadrotor. *2012 12th International Conference on Control, Automation and Systems*. IEEE, 162-167.
- [20] The Regulatory Horizons Council. (2021). *The Regulation of Drones: An exploratory study*. UK: Department for Business, Energy & Industrial Strategy (BEIS).
- [21] Lu, Y., Xue, Z., Xia, G.-S. ve Zhang, L. (2018). *A survey on vision-based UAV navigation*. *Geo-spatial information science*, 21(1), 21-32.
- [22] Pastor, E., Lopez, J. ve Royo, P. (2006). A hardware/software architecture for UAV payload and mission control. *2006 IEEE/AIAA 25TH Digital Avionics Systems Conference*. IEEE, 1-8.
- [23] Kendoul, F. (2012). *Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems*. *Journal of Field Robotics*, 29(2), 315-378.
- [24] Ronzhin, A., Ngo, T., Vu, Q. ve Nguyen, V. (2022). *Analysis of Approaches to the Control of Air Manipulation Systems*. *Ground and Air Robotic Manipulation Systems in Agriculture*. Heidelberg: Springer.
- [25] Hassan, M.M. (2019). Channel Modelling for Unmanned Aerial Vehicles. Yayınlanmamış Yüksek Lisans Tezi. Tampere: Tampere University.
- [26] Ramesh, P. ve Jeyan, J.M.L. (2020). Comparative analysis of the impact of operating parameters on military and civil applications of mini unmanned aerial vehicle (UAV). *AIP Conference Proceedings*. AIP Publishing LLC, 30-34.
- [27] Blyenburgh, P. (2010). Overview of the European UAS Community. *2010/2011 UAS Yearbook - UAS: The Global Perspective*. Paris: UVS International
- [28] Schöler, F., La Cour-Harbo, A. ve Bisgaard, M. (2012). *Generating Configuration Spaces and Visibility Graphs from a Geometric Workspace for UAV Path Planning*. *Autonomous Robots*.
- [29] Holub, J.S. (2010). Improving particle swarm optimization path planning through inclusion of flight mechanics. Yayınlanmamış Doktora Tezi. Ames: Iowa State University.
- [30] Russell, S. ve Norvig, P. (2010). *Intelligence artificielle: Avec plus de 500 exercices*. Paris: Pearson Education.
- [31] Raja, P. ve Pugazhenth, S. (2012). *Optimal path planning of mobile robots: A review*. *International journal of physical sciences*, 7(9), 1314-1320.
- [32] Mac, T.T., Copot, C., Tran, D.T. ve De Keyser, R. (2016). *Heuristic approaches in robot path planning: A survey*. *Robotics and Autonomous Systems*, 86, 13-28.
- [33] Wang, L.C., Yong, L.S. ve Ang, M.H. (2002). Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment. *Proceedings of the IEEE International Symposium on Intelligent Control*. IEEE, 821-826.
- [34] Gilmore, J.F. (1991). Autonomous vehicle planning analysis methodology. *AIAA guidance navigation control conference*. 1-15.
- [35] Papadimitriou, C.H., Raghavan, P., Sudan, M. ve Tamaki, H. (1994). Motion planning on a graph. *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE, 511-520.
- [36] Murphy, R.R. (2019). *Introduction to AI robotics*. Massachusetts: MIT Press.
- [37] Zhang, H.-Y., Lin, W.-M. ve Chen, A.-X. (2018). *Path planning for the mobile robot: A review*. *Symmetry*, 10(10), 450.

- [38] Techy, L. ve Woolsey, C.A. (2009). *Minimum-time path planning for unmanned aerial vehicles in steady uniform winds*. Journal of guidance, control, and dynamics, 32(6), 1736-1746.
- [39] Mcgee, T.G. ve Hedrick, J.K. (2006). Path planning and control for multiple point surveillance by an unmanned aircraft in wind. *2006 American Control Conference*. IEEE, 6-10.
- [40] Masehian, E. ve Sedighizadeh, D. (2007). *Classic and heuristic approaches in robot motion planning-a chronological review*. World Academy of Science, Engineering and Technology, 23(5), 101-106.
- [41] Szczerba, R.J., Galkowski, P., Glicktein, I.S. ve Ternullo, N. (2000). *Robust algorithm for real-time route planning*. Aerospace and Electronic Systems, IEEE Transactions on, 36(3), 869-878.
- [42] Wen, N., Zhao, L., Su, X. ve Ma, P. (2015). *UAV online path planning algorithm in a low altitude dangerous environment*. IEEE/CAA Journal of Automatica Sinica, 2(2), 173-185.
- [43] Krogh, B. ve Thorpe, C. (1986). Integrated path planning and dynamic steering control for autonomous vehicles. *Proceedings. 1986 IEEE International Conference on Robotics and Automation*. IEEE, 1664-1669.
- [44] Scheuer, A. ve Fraichard, T. (1997). Continuous-curvature path planning for car-like vehicles. *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS'97*. IEEE, 997-1003.
- [45] Wylie, M.P. ve Holtzman, J. (1996). The non-line of sight problem in mobile location estimation. *Proceedings of ICUPC-5th International Conference on Universal Personal Communications*. IEEE, 827-831.
- [46] Pearl, J. (1984). *Heuristics: intelligent search strategies for computer problem solving*. New Jersey: Addison-Wesley Longman Publishing Co., Inc.
- [47] Lavalle, S.M. (2006). *Planning algorithms*. Cambridge: Cambridge university press.
- [48] Lozano-Pérez, T. ve Wesley, M.A. (1979). *An algorithm for planning collision-free paths among polyhedral obstacles*. Communications of the ACM, 22(10), 560-570.
- [49] Lee, D.-T. (1978). *Proximity and reachability in the plane*. Champaign: University of Illinois at Urbana-Champaign.
- [50] Liu, Y.-H. ve Arimoto, S. (1992). *Path planning using a tangent graph for mobile robots among polygonal and curved obstacles: Communication*. The International Journal of Robotics Research, 11(4), 376-382.
- [51] Choset, H.M., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S. ve Arkin, R.C. (2005). *Principles of robot motion: theory, algorithms, and implementation*. Massachusetts: MIT press.
- [52] De Berg, M., Cheong, O. ve Van Kreveld, M. (2008). *Computational geometry: algorithms and applications*. Heidelberg: Springer.
- [53] Ferreira, T.I. (2009). Path planning for an unmanned terrestrial vehicle in an obstacle ridden environment. Yayınlanmamış Yüksek Lisans Tezi. Stellenbosch: University of Stellenbosch.
- [54] Medeiros, F. ve Da Silva, J. (2011). *A Dijkstra algorithm for fixed-wing UAV motion planning based on terrain elevation*. Advances in Artificial Intelligence-SBIA 2010, 213-222.

- [55] Jiang, K., Seneviratne, L. ve Earles, S. (1993). Finding the 3D shortest path with visibility graph and minimum potential energy. *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*. IEEE, 679-684.
- [56] Omar, R.B. (2012). Path planning for unmanned aerial vehicles using visibility line-based methods. Yayınlanmamış Doktora Tezi. Leicester: University of Leicester.
- [57] Bygi, M.N. ve Ghodsi, M. (2007). *3D visibility graph*. Computational Science and its Applications, Kuala Lumpur.
- [58] Medeiros, F.L.L. ve Silva, J.D.S.D. (2010). *Computational modeling for automatic path planning based on evaluations of the effects of impacts of UAVs on the ground*. Journal of Intelligent & Robotic Systems, 61(1-4), 181-202.
- [59] Omar, R. ve Gu, D. (2010). 3D Path Planning for Unmanned Aerial Vehicles using Visibility Line based Method. *ICINCO (1)*. 80-85.
- [60] Li, S.-M., Boskovic, J.D., Seereeram, S., Prasanth, R., Amin, J., Mehra, R.K., Beard, R.W. ve McLain, T.W. (2002). Autonomous hierarchical control of multiple unmanned combat air vehicles (UCAVs). *American Control Conference, 2002. Proceedings of the 2002*. IEEE, 274-279.
- [61] Xiao-Wei, F., Zhong, L. ve Xiao-Guang, G. (2010). Path planning for UAV in radar network area. *Intelligent Systems (GCIS), 2010 Second WRI Global Congress on*. IEEE, 260-263.
- [62] Shanmugavel, M., Tsourdos, A. ve White, B.A. (2010). Collision avoidance and path planning of multiple UAVs using flyable paths in 3D. *Methods and Models in Automation and Robotics (MMAR), 2010 15th International Conference on*. IEEE, 218-222.
- [63] Qu, Y.H., Pan, Q. ve Yan, H.G. (2005). *Flight path planning of UAV based on heuristically search and genetic algorithms*. Iecon 2005: Thirty-First Annual Conference of the Ieee Industrial Electronics Society, 1-3, 45-49.
- [64] Pehlivanoglu, Y.V. (2012). *A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV*. Aerospace Science and Technology, 16(1), 47-55.
- [65] Men, J.Z., E, Q. ve Yao, K.M. (2012). *Study on the Route Planning for Anti-Submarine Ship-Based UAV Based on Genetic Algorithm*. Advanced Materials Research, 433-440, 4823-4826.
- [66] Hu, Y.-F., Song, B.-F. ve Wang, X. (2009). *High Survivability Path Planning for Unmanned Air Vehicles*. Aeronautical Computing Technique, 5, 1-9.
- [67] Jian-You, Z. (2007). *Research on Unmanned Aerial Vehicle Mission Planning*. Radio Engineering of China, 12, 1-18.
- [68] Eun, Y. ve Bang, H. (2009). *Cooperative task assignment/path planning of multiple unmanned aerial vehicles using genetic algorithm*. Journal of aircraft, 46(1), 338-343.
- [69] Sud, A., Andersen, E., Curtis, S., Lin, M.C. ve Manocha, D. (2008). *Real-time path planning in dynamic virtual environments using multiagent navigation graphs*. IEEE transactions on visualization and computer graphics, 14(3), 526-538.
- [70] Krozel, J. ve Andrisani, D. (1990). *Navigation path planning for autonomous aircraft: Voronoi diagram approach*. Journal of Guidance, Control, and Dynamics, 13(6), 1152-1154.

- [71] Spritzer, Z.W. (2004). *Comparison of path-planning and search methods for cooperating unmanned aerial vehicles*. West Virginia: West Virginia University.
- [72] Brooks, R.A. ve Lozano-Perez, T. (1985). *A subdivision algorithm in configuration space for findpath with rotation*. IEEE Transactions on Systems, Man, and Cybernetics(2), 224-233.
- [73] Maron, O. ve Lozano-Perez, T. (1998). *Visible decomposition: Real-time path planning in large planar environments*. MIT, 1-17.
- [74] Zhu, D. ve Latombe, J.-C. (1989). *New heuristic algorithms for efficient hierarchical path planning*. California: Stanford Univ CA Dept. of Computer Science.
- [75] Björnsson, Y., Enzenberger, M., Holte, R., Schaeffer, J. ve Yap, P. (2003). Comparison of different grid abstractions for pathfinding on maps. *IJCAI*. 1511-1512.
- [76] Tozour, P. ve Austin, I. (2002). *Building a near-optimal navigation mesh*. AI game programming wisdom, 1, 298-304.
- [77] <http://www-cs-students.stanford.edu/~amitp/gameprog.html> (Erişim tarihi: 04.11.2021)
- [78] García, H.J.Q. ve Garrido, L. (2007). Towards exploration of unknown dynamic worlds using multiple robots. *2007 Sixth Mexican International Conference on Artificial Intelligence, Special Session (MICAI)*. IEEE Computer Society, 407-417.
- [79] Birch, C.P., Oom, S.P. ve Beecham, J.A. (2007). *Rectangular and hexagonal grids used for observation, experiment and simulation in ecology*. Ecological modelling, 206(3-4), 347-359.
- [80] Li, S., Sun, X. ve Xu, Y. (2006). Particle swarm optimization for route planning of unmanned aerial vehicles. *Information Acquisition, 2006 IEEE International Conference on*. IEEE, 1213-1218.
- [81] Gao, D., Gong, G., Wang, J. ve Han, L. (2012). Multi-resolution path planning for Miniature Air Vehicles with wind effect. *Industrial Informatics (INDIN), 2012 10th IEEE International Conference on*. IEEE, 167-171.
- [82] Bandi, S. ve Thalmann, D. (2000). *Path finding for human motion in virtual environments*. Computational Geometry, 15(1), 103-127.
- [83] Horng, J.-H. ve Tienyi Li, J. (2002). *Vehicle path planning by using adaptive constrained distance transformation*. Pattern Recognition, 35(6), 1327-1337.
- [84] Brener, N.E., Iyengar, S.S., Looney, H.C., Vakamudi, N., Yu, D., Huang, Q. ve Barhen, J. (2004). *Three-dimensional route planning for large grids*. Journal of the Indian Institute of Science, 84(3 & 4), 67.
- [85] Krawiec, B.M. (2012). *A*-Based Path Planning for an Unmanned Aerial and Ground Vehicle Team in a Radio Repeating Operation*. Yayınlanmamış Yüksek Lisans Tezi. Blacksburg: Virginia Tech.
- [86] Mcmanus, I.A., Clothier, R.A. ve Walker, R.A. (2005). Highly autonomous UAV mission planning and piloting for civilian airspace operations. *Eleventh Australian International Aerospace Congress, AIAC-11, First Australasian Unmanned Air Vehicles Conference*. 2-38.
- [87] Ghandi, S. ve Masehian, E. (2015). *Review and taxonomies of assembly and disassembly path planning problems and approaches*. Computer-Aided Design, 67, 58-86.
- [88] Lavalle, S.M. ve Kuffner, J.J. (2001). *Rapidly-Exploring Random Trees: Progress and Prospects: Steven M. LaValle, Iowa State University, A James J. Kuffner, Jr.,*

- University of Tokyo, Tokyo, Japan. *Algorithmic and Computational Robotics*, 303-307.
- [89] Kavraki, L.E., Svestka, P., Latombe, J.-C. ve Overmars, M.H. (1996). *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. *IEEE transactions on Robotics and Automation*, 12(4), 566-580.
 - [90] Lavalley, S.M. (1998). *Rapidly-exploring random trees: A new tool for path planning*. Iowa: Iowa State University.
 - [91] Vonásek, V., Faigl, J., Krajník, T. ve Přeučil, L. (2009). *RRT-path—a guided rapidly exploring random tree*. *Robot Motion and Control 2009*. Heidelberg: Springer.
 - [92] Bruce, J. ve Veloso, M. (2002). Real-time randomized path planning for robot navigation. *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. IEEE, 2383-2388.
 - [93] Yang, K., Keat Gan, S. ve Sukkarieh, S. (2013). *A Gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV*. *Advanced Robotics*(ahead-of-print), 1-13.
 - [94] Karaman, S. ve Frazzoli, E. (2012). Sampling-based algorithms for optimal motion planning with deterministic μ -calculus specifications. *2012 American Control Conference (ACC)*. IEEE, 735-742.
 - [95] Dias, A., Fernandes, T., Almeida, J., Martins, A. ve Silva, E. (2018). 3d Path Planning Methods For Unmanned Aerial Vehicles In Search And Rescue Scenarios. *Human-Centric Robotics: Proceedings of CLAWAR 2017: 20th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. World Scientific, 213-220.
 - [96] Janson, L., Schmerling, E., Clark, A. ve Pavone, M. (2015). *Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions*. *The International journal of robotics research*, 34(7), 883-921.
 - [97] Morin, M. (2010). Multi-criteria path planning with terrain visibility constraints. *Yayımlanmamış Yüksek Lisans Tezi*. Québec: Laval University.
 - [98] Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. IEEE, 500-505.
 - [99] Overmars, M.H. ve Svestka, P. (1994). *A probabilistic learning approach to motion planning*. Utrecht: Utrecht University.
 - [100] <http://www-cs-faculty.stanford.edu/~eroberts/courses/soco/projects/1998-99/robotics/basicmotion.html> (Erişim tarihi: 04.05.2013)
 - [101] Luukkonen, T. (2011). *Modelling and control of quadcopter 22*. Espoo: Aalto University.
 - [102] Gancet, J., Hattenberger, G., Alami, R. ve Lacroix, S. (2005). Task planning and control for a multi-UAV system: architecture and algorithms. *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 1017-1022.
 - [103] Miralles, Á.S. ve Bobi, M.Á.S. (2004). *Global path planning in gaussian probabilistic maps*. *Journal of Intelligent and Robotic Systems*, 40(1), 89-102.
 - [104] Magid, E., Keren, D., Rivlin, E. ve Yavneh, I. (2006). Spline-based robot navigation. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2296-2301.
 - [105] Tang, L., Dian, S., Gu, G., Zhou, K., Wang, S. ve Feng, X. (2010). A novel potential field method for obstacle avoidance and path planning of mobile robot.

- 2010 3rd international conference on computer science and information technology. IEEE, 633-637.
- [106] Mora, M.C. ve Tornero, J. (2008). Path planning and trajectory generation using multi-rate predictive artificial potential fields. *2008 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2990-2995.
 - [107] Brož, P. (2006). Path planning in combined 3D grid and graph environment. *Proceedings of the 10th Central European Seminar on Computer Graphics*. Citeseer, 1-7.
 - [108] Yin, L., Yin, Y. ve Lin, C.J. (2009). *A new potential field method for mobile robot path planning in the dynamic environments*. Asian Journal of Control, 11(2), 214-225.
 - [109] Kitamura, Y., Tanaka, T., Kishino, F. ve Yachida, M. (1995). 3-D path planning in a dynamic environment using an octree and an artificial potential field. *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*. IEEE, 474-481.
 - [110] Bate, S. ve Stanley, K. (1988). Heuristic route planning: an application to fighter aircraft. *Aerospace and Electronics Conference, 1988. NAECON 1988., Proceedings of the IEEE 1988 National*. IEEE, 1114-1120.
 - [111] Meng, B.-B. (2010). UAV path planning based on bidirectional sparse A* search algorithm. *2010 International Conference on Intelligent Computation Technology and Automation*. IEEE, 1106-1109.
 - [112] Schouwenaars, T., How, J. ve Feron, E. (2004). Receding horizon path planning with implicit safety guarantees. *Proceedings of the 2004 American control conference*. IEEE, 5576-5581.
 - [113] Chen, Y., Han, J. ve Zhao, X. (2012). *Three-dimensional path planning for unmanned aerial vehicle based on linear programming*. Robotica, 30(5), 773-781.
 - [114] Wang, G. ve Ming, L. (2007). *UCAV path planning based on modified PSO algorithm*. Aeronautical Computing Technique, 37(4), 10-13.
 - [115] Zhou, S., Zhu, G., Li, H., Wang, Y. ve Liu, X. (2011). Real-time route planning for uav based on weather threat. *2011 International Conference on Remote Sensing, Environment and Transportation Engineering*. IEEE, 2342-2345.
 - [116] Lei, L. ve Shiru, Q. (2012). Path planning for unmanned air vehicles using an improved artificial bee colony algorithm. *Proceedings of the 31st Chinese control conference*. IEEE, 2486-2491.
 - [117] Nikolos, I.K., Valavanis, K. P., Tsourveloudis, N. C., Kostaras, A. N. (2003). *Evolutionary algorithm based offline/online path planner for UAV navigation*. IEEE Trans Syst Man Cybern B Cybern, 33(6), 898-912.
 - [118] Kok, J., Gonzalez, L.F. ve Kelson, N. (2012). *FPGA implementation of an evolutionary algorithm for autonomous unmanned aerial vehicle on-board path planning*. IEEE transactions on evolutionary computation, 17(2), 272-281.
 - [119] Brintaki, A.N. ve Nikolos, I.K. (2005). *Coordinated UAV path planning using differential evolution*. Operational Research, 5(3), 487-502.
 - [120] Challita, U., Saad, W. ve Bettstetter, C. (2018). Deep reinforcement learning for interference-aware path planning of cellular-connected UAVs. *2018 IEEE International Conference on Communications (ICC)*. IEEE, 1-7.
 - [121] Hussein, A., Gaber, M.M., Elyan, E. ve Jayne, C. (2017). *Imitation learning: A survey of learning methods*. ACM Computing Surveys (CSUR), 50(2), 1-35.

- [122] Roberge, V., Tarbouchi, M. ve Labonté, G. (2011). Parallel Implementation and Comparison of Two UAV Path Planning Algorithms. *International Conference on Evolutionary Computation Theory and Applications*. SCITEPRESS, 162-167.
- [123] Schouwenaars, T., De Moor, B., Feron, E. ve How, J. (2001). Mixed integer programming for multi-vehicle path planning. *2001 European control conference (ECC)*. IEEE, 2603-2608.
- [124] King, E., Kuwata, Y. ve How, J.P. (2006). *Experimental demonstration of coordinated control for multi-vehicle teams*. International journal of systems science, 37(6), 385-398.
- [125] El-Sherbeny, N.A. (2010). *Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods*. Journal of King Saud University-Science, 22(3), 123-131.
- [126] Dijkstra, E.W. (1959). *A note on two problems in connexion with graphs*. Numerische mathematik, 1(1), 269-271.
- [127] Bozyiğit, A., Alankuş, G. ve Nasiboğlu, E. (2017). Public transport route planning: Modified dijkstra's algorithm. *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 502-505.
- [128] Fusic, S.J., Ramkumar, P. ve Hariharan, K. (2018). Path planning of robot using modified dijkstra Algorithm. *2018 National Power Engineering Conference (NPEC)*. IEEE, 1-5.
- [129] Wang, S.-X. ve Zhao, X.-Q. (2011). The improved Dijkstra's shortest path algorithm. *2011 Seventh International Conference on Natural Computation*. IEEE, 2313-2316.
- [130] Guo, S.-L., Duan, J., Zhu, Y., Li, X.-C. ve Chen, T.-W. (2018). Improved Dijkstra Algorithm Based on Fibonacci Heap for Solving the Shortest Path Problem with Specified Nodes. *Computer Science and Artificial Intelligence: Proceedings of the International Conference on Computer Science and Artificial Intelligence (CSAI2016)*. World Scientific, 52-61.
- [131] Erwig, M. ve Hagen, F. (2000). *The graph Voronoi diagram with applications*. Networks, 36(3), 156-163.
- [132] Labat, J.-M. ve Pomerol, J.-C. (2003). *Are Branch and Bound and A* algorithms identical?* Journal of Heuristics, 9(2), 131-143.
- [133] Hart, P.E., Nilsson, N.J. ve Raphael, B. (1968). *A formal basis for the heuristic determination of minimum cost paths*. IEEE transactions on Systems Science and Cybernetics, 4(2), 100-107.
- [134] Stentz, A. (1997). *Optimal and efficient path planning for partially known environments*. Intelligent unmanned ground vehicles. Heidelberg: Springer.
- [135] Daniel, K., Nash, A., Koenig, S. ve Felner, A. (2010). *Theta*: Any-angle path planning on grids*. Journal of Artificial Intelligence Research, 39, 533-579.
- [136] Newaz, A.a.R., Pratama, F.A. ve Chong, N.Y. (2013). Exploration priority based heuristic approach to uav path planning. *2013 IEEE RO-MAN*. IEEE, 521-526.
- [137] Nash, A., Koenig, S. ve Tovey, C. (2010). Lazy Theta*: Any-angle path planning and path length analysis in 3D. *Proceedings of the AAAI Conference on Artificial Intelligence*. 147-154.
- [138] Faria, M., Marín, R., Popović, M., Maza, I. ve Viguria, A. (2019). *Efficient lazy theta* path planning over a sparse grid to explore large 3d volumes with a multirotor UAV*. Sensors, 19(1), 165-174.
- [139] Wang, C., Wang, L., Qin, J., Wu, Z., Duan, L., Li, Z., Cao, M., Ou, X., Su, X. ve Li, W. (2015). Path planning of automated guided vehicles based on improved A-

- Star algorithm. *2015 IEEE International Conference on Information and Automation*. IEEE, 2071-2076.
- [140] Liu, G. ve Ramakrishnan, K. (2001). A* Prune: an algorithm for finding K shortest paths subject to multiple constraints. *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*. IEEE, 743-749.
 - [141] Garip, Z.B., Atali, G., Karayel, D. ve Ozkan, S.S. (2018). Path planning for multiple mobile robots in static environment using hybrid algorithm. *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE, 1-4.
 - [142] Aljazzar, H. ve Leue, S. (2011). *K**: A heuristic search algorithm for finding the k shortest paths. *Artificial Intelligence*, 175(18), 2129-2154.
 - [143] Likhachev, M., Ferguson, D.I., Gordon, G.J., Stentz, A. ve Thrun, S. (2005). Anytime Dynamic A*: An Anytime, Replanning Algorithm. *ICAPS*. 262-271.
 - [144] Sohrabi, S. (2019). AI Planning for Enterprise: Putting Theory Into Practice. *IJCAI*. 6408-6410.
 - [145] Aine, S. ve Likhachev, M. (2016). *Truncated incremental search*. *Artificial Intelligence*, 234, 49-77.
 - [146] Korf, R.E. (1990). *Real-time heuristic search*. *Artificial intelligence*, 42(2), 189-211.
 - [147] Undeger, C. ve Polat, F. (2007). *RTTES: Real-time search in dynamic environments*. *Applied Intelligence*, 27(2), 113-129.
 - [148] Likhachev, M. ve Stentz, A. (2006). PPCP: Efficient probabilistic planning with clear preferences in partially-known environments. *Proceedings of the National Conference on Artificial Intelligence*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 860-869.
 - [149] Dorigo, M. (1992). Optimization, learning and natural algorithms. Yayınlanmamış Doktora Tezi. Milano: Politecnico di Milano.
 - [150] Lučić, P. ve Teodorović, D. (2003). *Computing with bees: attacking complex transportation engineering problems*. *International Journal on Artificial Intelligence Tools*, 12(03), 375-394.
 - [151] Xu, C., Duan, H. ve Liu, F. (2010). *Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle (UCAV) path planning*. *Aerospace Science and Technology*, 14(8), 535-541.
 - [152] Zhang, Y.D., Wu, L.N. ve Wang, S.H. (2011). *UCAV path planning based on FSCABC*. *Information-an International Interdisciplinary Journal*, 14(3), 687-692.
 - [153] Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*: Luniver press.
 - [154] Hidalgo-Paniagua, A., Vega-Rodríguez, M.A., Ferruz, J. ve Pavón, N. (2017). *Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach*. *Soft Computing*, 21(4), 949-964.
 - [155] Alweshah, M. ve Abdullah, S. (2015). *Hybridizing firefly algorithms with a probabilistic neural network for solving classification problems*. *Applied Soft Computing*, 35, 513-524.
 - [156] Sutantyo, D., Levi, P., Möslinger, C. ve Read, M. (2013). Collective-adaptive lévy flight for underwater multi-robot exploration. *2013 IEEE International Conference on Mechatronics and Automation*. IEEE, 456-462.

- [157] Wang, G., Guo, L., Duan, H., Liu, L. ve Wang, H. (2012). *A modified firefly algorithm for UCAV path planning*. International Journal of Hybrid Information Technology, 5(3), 123-144.
- [158] Saremi, S., Mirjalili, S. ve Lewis, A. (2017). *Grasshopper optimisation algorithm: theory and application*. Advances in Engineering Software, 105, 30-47.
- [159] Elmi, Z. ve Efe, M.Ö. (2021). *Online path planning of mobile robot using grasshopper algorithm in a dynamic and unknown environment*. Journal of Experimental & Theoretical Artificial Intelligence, 33(3), 467-485.
- [160] Eberhart, R. ve Kennedy, J. (1995). Particle swarm optimization. *Proceedings of the IEEE international conference on neural networks*. Citeseer, 1942-1948.
- [161] Di Mario, E., Navarro, I. ve Martinoli, A. (2014). The role of environmental and controller complexity in the distributed optimization of multi-robot obstacle avoidance. *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 571-577.
- [162] Saska, M., Chudoba, J., Přeučil, L., Thomas, J., Loianno, G., Třešňák, A., Vonásek, V. ve Kumar, V. (2014). Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance. *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 584-595.
- [163] Foo, J., Knutzon, J., Oliver, J. ve Winer, E. (2006). Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization. *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, Virginia*. 1-11.
- [164] Fu, Y., Ding, M. ve Zhou, C. (2012). *Phase Angle-Encoded and Quantum-Behaved Particle Swarm Optimization Applied to Three-Dimensional Route Planning for UAV*. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 42(2), 511-526.
- [165] Huang, H.-C. ve Tsai, C.-C. (2011). Global path planning for autonomous robot navigation using hybrid metaheuristic GA-PSO algorithm. *SICE Annual Conference 2011*. IEEE, 1338-1343.
- [166] Yue, X. ve Zhang, W. (2018). UAV path planning based on k-means algorithm and simulated annealing algorithm. *2018 37th Chinese Control Conference (CCC)*. IEEE, 2290-2295.
- [167] Huo, L., Zhu, J., Wu, G. ve Li, Z. (2020). *A novel simulated annealing based strategy for balanced UAV task assignment and path planning*. Sensors, 20(17), 47-69.
- [168] Ryan, J.L., Bailey, T.G., Moore, J.T. ve Carlton, W.B. (1999). *Unmanned aerial vehicle (UAV) route selection using reactive tabu search*. Military Operations Research, 4(3), 5-24.
- [169] Rebollo, J.J., Ollero, A. ve Maza, I. (2007). Collision avoidance among multiple aerial robots and other non-cooperative aircraft based on velocity planning. *In Proceedings of ROBOTICA 2007 Conference, Paderne, Portugal*. 1-6.
- [170] Zadeh, L.A. (1965). *Information and control*. Fuzzy sets, 8(3), 338-353.
- [171] Zavlangas, P.G. ve Tzafestas, S.G. (2003). *Motion control for mobile robot obstacle avoidance and navigation: a fuzzy logic-based approach*. Systems Analysis Modelling Simulation, 43(12), 1625-1637.
- [172] Castellano, G., Attolico, G. ve Distanto, A. (1997). *Automatic generation of fuzzy rules for reactive robot controllers*. Robotics and Autonomous Systems, 22(2), 133-149.

- [173] Sabo, C. ve Cohen, K. (2012). *Fuzzy logic unmanned air vehicle motion planning*. Advances in Fuzzy Systems, 2012.
- [174] Zhuoning, D., Rulin, Z., Zongji, C. ve Rui, Z. (2010). *Study on UAV path planning approach based on fuzzy virtual force*. Chinese Journal of Aeronautics, 23(3), 341-350.
- [175] Hong, Z., Chun-Long, S., Zi-Jun, Z., Wei, A., De-Qiang, Z. ve Jing-Jing, W. (2015). A modified dynamic window approach to obstacle avoidance combined with fuzzy logic. *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*. IEEE, 523-526.
- [176] Motlagh, O.R.E., Hong, T.S. ve Ismail, N. (2009). *Development of a new minimum avoidance system for a behavior-based mobile robot*. Fuzzy sets and systems, 160(13), 1929-1946.
- [177] Jain, A.K., Mao, J. ve Mohiuddin, K.M. (1996). *Artificial neural networks: A tutorial*. Computer, 29(3), 31-44.
- [178] Mcculloch, W.S. ve Pitts, W. (1943). *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics, 5(4), 115-133.
- [179] Janglová, D. (2004). *Neural networks in mobile robot motion*. International Journal of Advanced Robotic Systems, 1(1), 15-22.
- [180] Wang, X., Hou, Z.-G., Zou, A., Tan, M. ve Cheng, L. (2008). *A behavior controller based on spiking neural networks for mobile robots*. Neurocomputing, 71(4-6), 655-666.
- [181] Li, Q.-L., Song, Y. ve Hou, Z.-G. (2015). *Neural network based FastSLAM for autonomous robots in unknown environments*. Neurocomputing, 165, 99-110.
- [182] Capi, G., Kaneko, S. ve Hua, B. (2015). *Neural network based guide robot navigation: an evolutionary approach*. Procedia Computer Science, 76, 74-79.
- [183] Medina-Santiago, A., Camas-Anzueto, J., Vazquez-Feijoo, J.A., Hernández-De León, H. ve Mota-Grajales, R. (2014). *Neural control system in obstacle avoidance in mobile robots using ultrasonic sensors*. Journal of applied research and technology, 12(1), 104-110.
- [184] Abubaker, A. (2012). *A novel mobile robot navigation system using neuro-fuzzy rule-based optimization technique*. Research Journal of Applied Sciences, Engineering and Technology, 4(15), 2577-2583.
- [185] Zhu, W. ve Duan, H. (2014). *Chaotic predator-prey biogeography-based optimization approach for UCAV path planning*. Aerospace science and technology, 32(1), 153-161.
- [186] Zhang, S., Zhou, Y., Li, Z. ve Pan, W. (2016). *Grey wolf optimizer for unmanned combat aerial vehicle path planning*. Advances in Engineering Software, 99, 121-136.
- [187] Duan, H. ve Huang, L. (2014). *Imperialist competitive algorithm optimized artificial neural networks for UCAV global path planning*. Neurocomputing, 125, 166-171.
- [188] Coello, C.a.C., Lamont, G.B. ve Van Veldhuizen, D.A. (2007). *Evolutionary algorithms for solving multi-objective problems*. 5. Heidelberg: Springer.
- [189] Peng, X. ve Xu, D. (2012). *Intelligent Online Path Planning for UAVs in Adversarial Environments*. International Journal of Advanced Robotic Systems, 9(1), 1-12.

- [190] Holland, J.H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Massachusetts: MIT press.
- [191] Goldberg David, E. ve Henry, H. (1988). *Genetic algorithms and machine learning*. Mach Learn, 3(2), 95-99.
- [192] Urso, A., Fiannaca, A., La Rosa, M., Ravi, V. ve Rizzo, R. (2018). *Data mining: Prediction methods*. Encycl. Bioinforma. Comput. Biol. ABC Bioinforma., 1, 413-430.
- [193] Jonasson, J. ve Norgren, E. (2016). *Investigating a genetic algorithm simulated annealing hybrid applied to university course time tabling problem*. Stockholm: KTH Royal Institute of Technology School of Computer Science and Communication.
- [194] Mitchell, M. (1998). *An introduction to genetic algorithms*. Massachusetts: MIT press.
- [195] Siriwardene, N.R. ve Perera, B. (2006). *Selection of genetic algorithm operators for urban drainage model parameter optimisation*. Mathematical and Computer Modelling, 44(5-6), 415-429.
- [196] Xu, H. (2015). Solver tuning with genetic algorithms. Yayınlanmamış Doktora Tezi. Dundee: University of Dundee.
- [197] Picek, S., Jakobovic, D. ve Golub, M. (2013). On the recombination operator in the real-coded genetic algorithms. *2013 IEEE Congress on Evolutionary Computation*. IEEE, 3103-3110.
- [198] Eiben, A.E. ve Smith, J.E. (2003). *Introduction to evolutionary computing*. 53. Heidelberg: Springer.
- [199] Goldberg, D.E. ve Deb, K. (1991). *A comparative analysis of selection schemes used in genetic algorithms*. Foundations of genetic algorithms. 1. Massachusetts: Elsevier.
- [200] Cobb, H.G. ve Grefenstette, J.J. (1993). *Genetic algorithms for tracking changing environments*. Washington DC: Naval Research Lab.
- [201] Grefenstette, J.J. (1992). Genetic algorithms for changing environments. *Ppsn*. Citeseer, 137-144.
- [202] Cedeno, W. ve Vemuri, V.R. (1997). On the use of niching for dynamic landscapes. *Evolutionary Computation, 1997., IEEE International Conference on*. IEEE, 361-366.
- [203] Mori, N., Kita, H. ve Nishikawa, Y. (2001). *Adaptation to changing environments by means of the memory based thermodynamical genetic algorithm*. Transactions of the Institute of Systems, Control and Information Engineers, 14(1), 33-41.
- [204] Peng, X., Gao, X. ve Yang, S. (2011). *Environment identification-based memory scheme for estimation of distribution algorithms in dynamic environments*. Soft Computing, 15(2), 311-326.
- [205] Yang, S. ve Yao, X. (2008). *Population-based incremental learning with associative memory for dynamic environments*. Evolutionary Computation, IEEE Transactions on, 12(5), 542-561.
- [206] Branke, J. (1999). Memory enhanced evolutionary algorithms for changing optimization problems. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*. IEEE, 1875-1882.
- [207] Ursem, R.K. (2000). Multinational GAs: Multimodal optimization techniques in dynamic environments. *Proceedings of the Genetic and Evolutionary Computation Conference*. 19-26.

- [208] Branke, J., Kaußler, T., Smidt, C. ve Schmeck, H. (2000). *A multi-population approach to dynamic optimization problems*. Evolutionary Design and Manufacture. Heidelberg: Springer.
- [209] Wineberg, M. ve Oppacher, F. (2000). Enhancing the GA's ability to cope with dynamic environments. *Proceedings of Genetic and Evolutionary Computation Conference*. 3-10.
- [210] Li, C. ve Yang, S. (2012). *A general framework of multipopulation methods with clustering in undetectable dynamic environments*. Evolutionary Computation, IEEE Transactions on, 16(4), 556-577.
- [211] Cakir, M. (2015). 2D path planning of UAVs with genetic algorithm in a constrained environment. *2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*. IEEE, 1-5.
- [212] Özalp, N. ve Sahingoz, O.K. (2013). Optimal UAV path planning in a 3D threat environment by using parallel evolutionary algorithms. *2013 International conference on unmanned aircraft systems (ICUAS)*. IEEE, 308-317.
- [213] Fu, X. ve Gao, X. (2010). Genetic algorithm with adaptive immigrants for dynamic flight path planning. *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems*. IEEE, 630-634.
- [214] Zhao, L. ve Murthy, V. (2007). Optimal flight path planner for an unmanned helicopter by evolutionary algorithms. *AIAA guidance, navigation and control conference and exhibit*. 6741.
- [215] Jia, D. ve Vagners, J. (2004). Parallel evolutionary algorithms for UAV path planning. *AIAA 1st intelligent systems technical conference*. 6230.
- [216] Xiao, J., Michalewicz, Z., Zhang, L. ve Trojanowski, K. (1997). *Adaptive evolutionary planner/navigator for mobile robots*. IEEE transactions on evolutionary computation, 1(1), 18-28.
- [217] Cekmez, U., Ozsiginan, M. ve Sahingoz, O.K. (2013). Adapting the GA approach to solve Traveling Salesman Problems on CUDA architecture. *2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE, 423-428.
- [218] Nikolos, I.K. ve Brintaki, A.N. (2005). *Coordinated UAV path planning using differential evolution*. 2005 IEEE International Symposium on Intelligent Control & 13th Mediterranean Conference on Control and Automation, Vols 1 and 2, 549-556.
- [219] Nikolos, I., Zografos, E. ve Brintaki, A. (2007). *UAV path planning using evolutionary algorithms*. Innovations in Intelligent Machines-1, 77-111.
- [220] Sahingoz, O.K. (2013). Flyable path planning for a multi-UAV system with Genetic Algorithms and Bezier curves. *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 41-48.
- [221] Liu, X.-F., Peng, Z.-R., Chang, Y.-T. ve Zhang, L.-Y. (2012). *Multi-objective evolutionary approach for UAV cruise route planning to collect traffic information*. Journal of Central South University, 19(12), 3614-3621.
- [222] Liu, X., Peng, Z., Zhang, L. ve Li, L. (2012). *Unmanned Aerial Vehicle Route Planning for Traffic Information Collection*. Journal of Transportation Systems Engineering and Information Technology, 12(1), 91-97.
- [223] Fu, Y.-G., Zhou, C.-P. ve Hu, H.-P. (2012). *Research on Differential Evolution Algorithm for Path Planning for Unmanned Aerial Vehicle in Ocean Environment*. Acta Armamentarii, 3, 1-9.

- [224] Fu, S.-Y., Han, L.-W., Tian, Y. ve Yang, G.-S. (2012). Path planning for unmanned aerial vehicle based on genetic algorithm. *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2012 IEEE 11th International Conference on.* IEEE, 140-144.
- [225] Zheng, R., Feng, Z.-M. ve Lu, M.-Q. (2011). *Application of Particle Genetic Algorithm to Path Planning of Unmanned Aerial Vehicle.* Computer Simulation, 6, 88-92.
- [226] Pakkan, B. ve Ermiş, M. (2010). *İnsansız hava araçlarının genetik algoritma yöntemiyle çoklu hedeflere planlanması.* Havacılık ve Uzay Teknolojileri Dergisi, 4(3), 77-84.
- [227] Nikolos I. K., T.N.C., Valavanis K. P. (2001). Evolutionary algorithm based 3-D path planner for UAV navigation. *CD-ROM Proceedings, 9 th Mediterranean Conference on Control and Automation, Dubrovnik, Croatia.* 1-12.
- [228] Mittal, S. ve Deb, K. (2007). Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms. *2007 IEEE Congress on Evolutionary Computation.* IEEE, 3195-3202.
- [229] Oz, I., Topcuoglu, H.R. ve Ermis, M. (2013). *A meta-heuristic based three-dimensional path planning environment for unmanned aerial vehicles.* Simulation, 89(8), 903-920.
- [230] Cheng, Z., Sun, Y. ve Liu, Y. (2011). Path planning based on immune genetic algorithm for UAV. *2011 International Conference on Electric Information and Control Engineering.* IEEE, 590-593.
- [231] Shima, T. ve Schumacher, C. (2008). *Assigning cooperating UAVs to simultaneous tasks on consecutive targets using genetic algorithms.* Journal of the Operational Research Society, 60(7), 973-982.
- [232] Ergezer, H. ve Leblebicioglu, K. (2013). *Path Planning for UAVs for Maximum Information Collection.* Ieee Transactions on Aerospace and Electronic Systems, 49(1), 502-520.
- [233] Sun, Y., Wu, C., Zhang, Y. ve Hu, N. (2012). *Holes Detection in Wireless Sensor Network and Redeploy Route Planning Based on Unmanned Aerial Vehicle.* Journal of Information and Computational Science, 9, 1-13.
- [234] Qingtian, H., Wenjing, C. ve Jia, C. (2012). Research of Route Planning Based on Genetic Algorithm. *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on.* IEEE, 199-202.
- [235] Pierre, D.M., Zakaria, N. ve Pal, A.J. (2012). *Quantitative and Qualitative Analysis of Unmanned Aerial Vehicle's Path Planning Using Master-Slave Parallel Vector-Evaluated Genetic Algorithm.* Proceedings of the International Conference on Soft Computing for Problem Solving (Socpros 2011), Vol 1, 130, 567-577.
- [236] Alba, E. ve Dorronsoro, B. (2009). *Cellular genetic algorithms.* 42. Heidelberg: Springer Science & Business Media.
- [237] De Jong, K.A. (1975). *An analysis of the behavior of a class of genetic adaptive systems.* Michigan: University of Michigan.
- [238] Goldberg, D.E. ve Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms.* Hillsdale, NJ: Lawrence Erlbaum, 41-49.

- [239] Gorges-Schleuter, M. (1990). Explicit parallelism of genetic algorithms through population structures. *International Conference on Parallel Problem Solving from Nature*. Springer, 150-159.
- [240] Cantú-Paz, E. (1998). *A survey of parallel genetic algorithms*. *Calculateurs paralleles, reseaux et systems repartis*, 10(2), 141-171.
- [241] Whitley, L.D. (1993). Cellular genetic algorithms. *Proceedings of the 5th International Conference on Genetic Algorithms*. 1-12.
- [242] Manderick, B. ve Spiessens, P. (1989). Fine-grained parallel genetic algorithms. *Proceedings of the third international conference on Genetic algorithms*. Morgan Kaufmann Publishers Inc., 428-433.
- [243] Robertson, G.G. (1987). Parallel implementation of genetic algorithms in a classifier system. *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms*. Massachusetts Institute of Technology, Cambridge: Hillsdale, NJ: L. Erlbaum Associates, 140-147.
- [244] Wright, S. (1943). *Isolation by distance*. *Genetics*, 28(2), 114-138.
- [245] Alba, E. ve Tomassini, M. (2002). *Parallelism and evolutionary algorithms*. *IEEE transactions on evolutionary computation*, 6(5), 443-462.
- [246] Spiessens, P. ve Manderick, B. (1991). A massively parallel genetic algorithm: Implementation and first analysis. *Proceedings of the Fourth International Conference on Genetic Algorithms*. 279-286.
- [247] Collins, R.J. ve Jefferson, D.R. (1991). *Selection in massively parallel genetic algorithms*. California:: University of California.
- [248] Alba, E. ve Troya, J.M. (2002). *Improving flexibility and efficiency by adding parallelism to genetic algorithms*. *Statistics and Computing*, 12(2), 91-114.
- [249] Dittrich, T. ve Elmenreich, W. (2015). Comparison of a spatially-structured cellular evolutionary algorithm to an evolutionary algorithm with panmictic population. *2015 12th International Workshop on Intelligent Solutions in Embedded Systems (WISES)*. IEEE, 145-149.
- [250] Tomassini, M. (2010). *Cellular evolutionary algorithms*. *Simulating Complex Systems by Cellular Automata*. Heidelberg: Springer.
- [251] Dorronsoro, B. ve Bouvry, P. (2010). Differential evolution algorithms with cellular populations. *International Conference on Parallel Problem Solving from Nature*. Springer, 320-330.
- [252] Neumann, J.V. (1966). *Theory of self-reproducing automata*. Edited by Arthur W. Burks. Urbana and London: University of Illinois Press.
- [253] Toffoli, T. ve Margolus, N. (1987). *Cellular automata machines: a new environment for modeling*. Massachusetts: MIT press.
- [254] Sipper, M. (1997). *Evolution of parallel cellular machines*. 4. Heidelberg: Springer.
- [255] Alba, E. ve Dorronsoro, B. (2008). *The state of the art in cellular evolutionary algorithms*. *Cellular Genetic Algorithms*. Heidelberg: Springer.
- [256] Mantere, T. (2009). Image ordering by cellular genetic algorithms with TSP and ICA. *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 822-829.
- [257] Alba, E. ve Dorronsoro, B. (2004). *Solving the vehicle routing problem by using cellular genetic algorithms*. *Evolutionary Computation in Combinatorial Optimization*. Heidelberg: Springer.

- [258] Alba, E., Alfonso, H. ve Dorronsoro, B. (2005). Advanced models of cellular genetic algorithms evaluated on SAT. *Proceedings of the 2005 conference on Genetic and evolutionary computation*. ACM, 1123-1130.
- [259] Folino, G., Pizzuti, C. ve Spezzano, G. (1998). Combining cellular genetic algorithms and local search for solving satisfiability problems. *Tools with Artificial Intelligence, 1998. Proceedings. Tenth IEEE International Conference on*. IEEE, 192-198.
- [260] Murata, T. ve Gen, M. (2002). Cellular genetic algorithm for multi-objective optimization. *Proc. of the 4th Asian Fuzzy System Symposium*. 538-542.
- [261] Huy, N.Q., Soon, O.Y., Hiot, L.M. ve Krasnogor, N. (2009). *Adaptive cellular memetic algorithms*. Evolutionary Computation, 17(2), 231-256.
- [262] Janson, S., Alba, E., Dorronsoro, B. ve Middendorf, M. (2006). Hierarchical cellular genetic algorithm. *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer, 111-122.
- [263] Durrett, R. (1995). *Ten lectures on particle systems*. Lectures on Probability Theory. Heidelberg: Springer.
- [264] Schönfisch, B. ve De Roos, A. (1999). *Synchronous and asynchronous updating in cellular automata*. BioSystems, 51(3), 123-143.
- [265] Dubins, L.E. (1957). *On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents*. American Journal of mathematics, 79(3), 497-516.
- [266] Mann, S. ve Davidchuk, M. (1997). A parametric hybrid triangular Bézier patch. *Proceedings of the international conference on Mathematical methods for curves and surfaces II Lillehammer*. 335-342.
- [267] Farin, G. (1988). *Curves and surfaces for computer aided geometric design: A practical guide*. San Diego, CA, Academic Press, Inc., 1988, 348.
- [268] De Boor, C. ve De Boor, C. (1978). *A practical guide to splines*. 27. New York: springer-verlag.
- [269] Shu-Bo, T., Yan, Z., Lei, L. ve Wei, W. (2004). Discrete trajectory tracking control of wheeled mobile robots. *2004 IEEE International Conference on Robotics and Biomimetics*. IEEE, 344-349.
- [270] Huang, Y., Chen, J., Wang, H. ve Su, G. (2019). *A method of 3D path planning for solar-powered UAV with fixed target and solar tracking*. Aerospace Science and Technology, 92, 831-838.
- [271] Uragun, B. (2011). Energy efficiency for unmanned aerial vehicles. *2011 10th International Conference on Machine Learning and Applications and Workshops*. IEEE, 316-320.
- [272] Peng, H., Razi, A., Afghah, F. ve Ashdown, J. (2018). *A unified framework for joint mobility prediction and object profiling of drones in UAV networks*. Journal of Communications and Networks, 20(5), 434-442.
- [273] Chen, J., Mitra, U. ve Gesbert, D. (2019). Optimal UAV relay placement for single user capacity maximization over terrain with obstacles. *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 1-5.
- [274] Tran, T.X., Hajisami, A. ve Pompili, D. (2017). *Cooperative hierarchical caching in 5G cloud radio access networks*. IEEE Network, 31(4), 35-41.
- [275] Langelan, J.W. (2009). *Gust energy extraction for mini and micro uninhabited aerial vehicles*. Journal of guidance, control, and dynamics, 32(2), 464-473.

- [276] Langelaan, J. (2007). Long distance/duration trajectory optimization for small UAVs. *AIAA Guidance, Navigation and Control Conference and Exhibit*. 1-14
- [277] Zeng, Y. ve Zhang, R. (2017). *Energy-efficient UAV communication with trajectory optimization*. IEEE Transactions on Wireless Communications, 16(6), 3747-3760.
- [278] Floreano, D. ve Mondada, F. (1996). *Evolution of homing navigation in a real mobile robot*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(3), 396-407.
- [279] Williams, A. ve Yakimenko, O. (2018). Persistent mobile aerial surveillance platform using intelligent battery health management and drone swapping. *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 237-246.
- [280] Fujii, K., Higuchi, K. ve Rekimoto, J. (2013). Endless flyer: a continuous flying drone with automatic battery replacement. *2013 IEEE 10th international conference on ubiquitous intelligence and computing and 2013 IEEE 10th international conference on autonomic and trusted computing*. IEEE, 216-223.
- [281] Chen, Y., Baek, D., Bocca, A., Macii, A., Macii, E. ve Poncino, M. (2018). A case for a battery-aware model of drone energy consumption. *2018 IEEE International Telecommunications Energy Conference (INTELEC)*. IEEE, 1-8.
- [282] Lyu, J., Zeng, Y. ve Zhang, R. (2016). *Cyclical multiple access in UAV-aided communications: A throughput-delay tradeoff*. IEEE Wireless Communications Letters, 5(6), 600-603.
- [283] Qi, Y.C. ve Zhao, Y.J. (2005). *Energy-efficient trajectories of unmanned aerial vehicles flying through thermals*. Journal of Aerospace Engineering, 18(2), 84-92.
- [284] Kniffin, C.A., Dogan, A. ve Blake, W.B. (2016). Formation flight for fuel saving in coronet mission-part a: Sweet spot determination. *AIAA Atmospheric Flight Mechanics Conference*. 1-14.
- [285] https://www.atraircraft.com/userfiles/files/Fuel_Saving_2011.pdf (Erişim tarihi: 09.08.2016)
- [286] Mayer, R. (2002). A flight trajectory model for a pc-based airspace analysis tool. *AIAA Modeling and Simulation Technologies Conference and Exhibit*. 1-10.
- [287] <https://www.eurocontrol.int/model/bada> (Erişim tarihi: 20.03.2022)
- [288] Li, D.C., Cheng, N., Cheng, P. ve Song, J.Y. (2012). Design and Development of an Integrated Flight Planning and Rehearsal System Based on GIS and Navigation Database. *Applied Mechanics and Materials*. Trans Tech Publ, 169-177.
- [289] Labonte, G. (2015). *Simple formulas for the fuel of climbing propeller driven airplanes*. Advances in aircraft and spacecraft science, 2(4), 367-389.
- [290] Roberge, V., Tarbouchi, M. ve Labonté, G. (2018). *Fast genetic algorithm path planner for fixed-wing military UAV using GPU*. IEEE Transactions on Aerospace and Electronic Systems, 54(5), 2105-2117.
- [291] Holland, G.J., Mcgeer, T. ve Youngren, H. (1992). *Autonomous aerosondes for economical atmospheric soundings anywhere on the globe*. Bulletin of the American Meteorological Society, 73(12), 1987-1998.
- [292] Niculescu, M. (2001). Lateral track control law for Aerosonde UAV. *39th Aerospace Sciences Meeting and Exhibit*. 1-11.
- [293] Sliwinski, J., Gardi, A., Marino, M. ve Sabatini, R. (2017). *Hybrid-electric propulsion integration in unmanned aircraft*. Energy, 140, 1407-1416.

- [294] Hendrickson, S.P. ve Mcgeer, T. (1999). *A Miniature Powerplant for Very Small Very Long Range Autonomous Aircraft*. Bingen, WA (US): The Insitu Group Inc.
- [295] Al-Hababbeh, O.M. (2019). *Configuration of hybrid fuel-electric airplane model based on full flight path performance*. Jurnal Teknologi, 84(3), 59-67.
- [296] Lobit Cerrato, A. (2016). *Simulación de vuelo de múltiples UAVs de ala fija*. Sevilla: Universidad de Sevilla.
- [297] Bayraktar, O., Özdemir, F., Çetin, Ö. ve Yılmaz, G. (2012). *İnsansız Hava Araçları İçin Otonom İniş Sistemi Simülâtörü Tasarımı*. Bilişim Teknolojileri Dergisi, 5(2), 1-8.
- [298] Petty, C. (1997). *Diffusion (cellular) models*. Handbook of evolutionary computation. Oxford: Oxford Univ. Press.
- [299] Garcia-Najera, A. ve Bullinaria, J.A. (2008). A multi-objective density restricted genetic algorithm for the vehicle routing problem with time windows. *2008 UK Workshop on Computational Intelligence*. 47-52.
- [300] Sarma, J. ve De Jong, K. (1996). An analysis of the effects of neighborhood size and shape on local selection algorithms. *International Conference on Parallel Problem Solving From Nature*. Springer, 236-244.
- [301] Rudolph, G. (2001). *On takeover times in spatially structured populations: Array and ring*. Dortmund: Universitat Dortmund.
- [302] Li, D., Gu, H. ve Zhang, L. (2013). *A hybrid genetic algorithm-fuzzy c-means approach for incomplete data clustering based on nearest-neighbor intervals*. Soft Computing, 17(10), 1787-1796.
- [303] Su, J.-P., Lee, T.-E. ve Yu, K.-W. (2009). *A combined hard and soft variable-structure control scheme for a class of nonlinear systems*. IEEE Transactions on industrial electronics, 56(9), 3305-3313.
- [304] Lee, K.-Y., Han, S.-N. ve Roh, M.-I. (2003). *An improved genetic algorithm for facility layout problems having inner structure walls and passages*. Computers & Operations Research, 30(1), 117-138.
- [305] Nogueras, R. ve Cotta, C. Using distribution-based operators in genetic algorithms. *Proceedings of the 6th Metaheuristic International Conference*, págs. 707-712.
- [306] Gezer, A., Turan, Ö. ve Baklacioğlu, T. (2016). *UAV path planning using segmented cellular evolutionary algorithm*. International Journal of Sustainable Aviation, 2(3), 222-234.
- [307] Srinivas, N. ve Deb, K. (1994). *Muiltiobjective optimization using nondominated sorting in genetic algorithms*. Evolutionary computation, 2(3), 221-248.
- [308] Deb, K., Pratap, A., Agarwal, S. ve Meyarivan, T. (2002). *A fast and elitist multiobjective genetic algorithm: NSGA-II*. IEEE transactions on evolutionary computation, 6(2), 182-197.
- [309] Chiarella, C. ve Tô, T.-D. (2006). *The multifactor nature of the volatility of futures markets*. Computational Economics, 27(2), 163-183.
- [310] Shim, Y. ve Husbands, P. (2015). *Incremental embodied chaotic exploration of self-organized motor behaviors with proprioceptor adaptation*. Frontiers in Robotics and AI, 2, 1-7.
- [311] Kanury, S. ve Song, Y. (2006). Flight management of multiple aerial vehicles using genetic algorithms. *2006 Proceeding of the Thirty-Eighth Southeastern Symposium on System Theory*. IEEE, 33-37.

- [312] Mcmanus, I.A. (2005). A multidisciplinary approach to highly autonomous UAV mission planning and piloting for civilian airspace. Yayınlanmamış Doktora Tezi. Brisbane: Queensland University of Technology.
- [313] Homaifar, A., Qi, C.X. ve Lai, S.H. (1994). *Constrained optimization via genetic algorithms*. Simulation, 62(4), 242-253.
- [314] Xia, S. ve Zhang, X. (2021). Constrained path planning for unmanned aerial vehicle in 3D terrain using modified multi-objective particle swarm optimization. *Actuators*. Multidisciplinary Digital Publishing Institute, 255-265.
- [315] Mcmanus, I.A. ve Walker, R.A. (2006). *Multidisciplinary approach to intelligent unmanned-airborne-vehicles mission planning*. Journal of Aircraft, 43(2), 318-335.
- [316] Li, S.D., Liu, S., Ai, Q. ve Li, J. (2013). *Fast 3D Path Pre-Planning Method for UAVs*. Applied Mechanics and Materials, 278, 1676-1681.
- [317] Zheng, C., Ding, M. ve Zhou, C. (2011). *Real-Time Route Planning for Unmanned Air Vehicle with an Evolutionary Algorithm*. International Journal of Pattern Recognition and Artificial Intelligence, 17(01), 63-81.
- [318] Mao, S.S.L. (2000). *Path Planning Of Multi Mobile Robots Using Genetic Algorithms [J]*. ACTA Automatica Sinica, 26(5), 672-676.
- [319] Lin, L. ve Goodrich, M.A. (2009). *UAV intelligent path planning for wilderness search and rescue*. Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, 709-714.
- [320] Sun, T.Y., Huo, C.L., Tsai, S.J. ve Liu, C.C. (2008). *Optimal UAV flight path planning using skeletonization and particle swarm optimizer*. 2008 Ieee Congress on Evolutionary Computation, Vols 1-8, 1183-1188.
- [321] Wilburn, B.K., Perhinschi, M.G. ve Wilburn, J.N. (2014). *A modified genetic algorithm for UAV trajectory tracking control laws optimization*. International Journal of Intelligent Unmanned Systems, 2(2), 58-90.
- [322] Jie, L., Liu, W., Sun, Z. ve Teng, S. (2017). *Hybrid fuzzy clustering methods based on improved self-adaptive cellular genetic algorithm and optimal-selection-based fuzzy c-means*. Neurocomputing, 249, 140-156.
- [323] Alba, E. ve Dorronsoro, B. (2005). *The exploration/exploitation tradeoff in dynamic cellular genetic algorithms*. IEEE transactions on evolutionary computation, 9(2), 126-142.
- [324] Al-Naqi, A., Erdogan, A.T. ve Arslan, T. (2013). *Adaptive three-dimensional cellular genetic algorithm for balancing exploration and exploitation processes*. Soft Computing, 17(7), 1145-1157.
- [325] Baldonado, M., Chang, C.-C.K., Gravano, L. ve Paepcke, A. (1997). *The Stanford digital library metadata architecture*. International journal on digital libraries, 1(2), 108-121.
- [326] <http://www.u-dynamics.com/aerosim/> (Erişim tarihi: 01.02.2014)
- [327] Yingxun, W. ve Zongji, C. (1999). *Genetic algorithms (GA) based flight path planning with constraints*. J. Beijing Univ. Aeronaut. Astronaut.(China), 25(3), 355-358.
- [328] Schopferer, S. ve Pfeifer, T. (2015). Performance-aware flight path planning for unmanned aircraft in uniform wind fields. *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. IEEE, 1138-1147.
- [329] Yakovlev, K., Makarov, D. ve Baskin, E. (2015). *Automatic path planning for an unmanned drone with constrained flight dynamics*. Scientific and Technical Information Processing, 42(5), 347-358.

- [330] Myers, D., Batta, R. ve Karwan, M. (2016). *A real-time network approach for including obstacles and flight dynamics in UAV route planning*. The Journal of Defense Modeling and Simulation, 13(3), 291-306.
- [331] Faa. (2009). *Pilot's handbook of aeronautical knowledge*. New York: Skyhorse Publishing Inc.
- [332] Scherer, S. (2011). *Low-altitude operation of unmanned rotorcraft*. Pensilvanya: Carnegie Mellon University.
- [333] Singh, Y., Sharma, S., Hatton, D. ve Sutton, R. (2018). *Optimal path planning of unmanned surface vehicles*. Indian Journal of Geo Marine Sciences, 47(7), 1325-1334.
- [334] Shorakaei, H., Vahdani, M., Imani, B. ve Gholami, A. (2016). *Optimal cooperative path planning of unmanned aerial vehicles by a parallel genetic algorithm*. Robotica, 34(4), 823-836.
- [335] Luque, G., Alba, E. ve Dorronsoro, B. (2009). *Analyzing parallel cellular genetic algorithms*. Optimization Techniques for Solving Complex Problems, 49-62.
- [336] Folino, G., Pizzuti, C. ve Spezzano, G. (2003). *A scalable cellular implementation of parallel genetic programming*. IEEE Transactions on Evolutionary Computation, 7(1), 37-53.
- [337] Alba, E. (2005). *Parallel metaheuristics: a new class of algorithms*. 47. New Jersey: John Wiley & Sons.
- [338] Hossain, R., Magierowski, S. ve Messier, G.G. (2014). GPU enhanced path finding for an unmanned aerial vehicle. *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*. IEEE, 1285-1293.
- [339] Cekmez, U., Ozsiginan, M. ve Sahingoz, O.K. (2014). A UAV path planning with parallel ACO algorithm on CUDA platform. *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 347-354.
- [340] Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C. ve Burgard, W. (2013). *OctoMap: An efficient probabilistic 3D mapping framework based on octrees*. Autonomous robots, 34(3), 189-206.
- [341] Boulos, M.N.K. (2005). *Web GIS in practice III: creating a simple interactive map of England's strategic Health Authorities using Google Maps API, Google Earth KML, and MSN Virtual Earth Map Control*. 4(22), 1-8.
- [342] Schmidt, M. ve Weiser, P. (2012). *Web mapping services: development and trends*. Online maps with APIs and WebServices: Springer.
- [343] Hentati, A.I., Krichen, L., Fourati, M. ve Fourati, L.C. (2018). Simulation tools, environments and frameworks for UAV systems performance analysis. *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 1495-1500.
- [344] Madaan, R., Gyde, N., Vemprala, S., Brown, M., Nagami, K., Taubner, T., Cristofalo, E., Scaramuzza, D., Schwager, M. ve Kapoor, A. (2020). Airsim drone racing lab. *NeurIPS 2019 Competition and Demonstration Track*. PMLR, 177-191.
- [345] Perry, A.R. (2004). The flightgear flight simulator. *Proceedings of the USENIX Annual Technical Conference*. 1-53.
- [346] Spencer, J., Lee, J., Paredes, J.A., Goel, A. ve Bernstein, D. (2021). *An Adaptive PID Autotuner for Multicopters with Experimental Results*. ArXiv, 2109(12797), 1-7.

- [347] Wright, T., West, A., Licata, M., Hawes, N. ve Lennox, B. (2021). *Simulating Ionising Radiation in Gazebo for Robotic Nuclear Inspection Challenges*. Robotics, 10(3), 86-102.
- [348] Henriques, L.C.F. ve De Souza, C.P. (2021). *Development of an Embedded Longitudinal Flight Control Based on X-Plane Flight Simulator*. IEEE Latin America Transactions, 19(10), 1684-1691.
- [349] Mueller, M., Casser, V., Lahoud, J., Smith, N. ve Ghanem, B. (2017). *Ue4sim: A photo-realistic simulator for computer vision applications*. arXiv, 1708(05869), 1-18.
- [350] Shah, S., Dey, D., Lovett, C. ve Kapoor, A. (2018). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. *Field and service robotics*. Springer, 621-635.
- [351] Wang, S., Chen, J., Zhang, Z., Wang, G., Tan, Y. ve Zheng, Y. (2017). Construction of a virtual reality platform for UAV deep learning. *2017 Chinese Automation Congress (CAC)*. IEEE, 3912-3916.
- [352] Elsadig, M.A. ve Elbakri, M.A. (2017). Design of autopilot platform using HILS approach. *Communication, Control, Computing and Electronics Engineering (ICCCCEE), 2017 International Conference on*. IEEE, 1-6.
- [353] Huang, H., Xu, Y.-P. ve Deng, Z.-W. (2007). *Real-time visual flight simulation system based on flightgear simulator*. Journal of System simulation, 19(19), 780-785.
- [354] Sorton, E. ve Hammaker, S. (2005). *Simulated flight testing of an autonomous unmanned aerial vehicle using flightgear*. Infotech@ Aerospace, 1-11.
- [355] Joscelyne, R. ve Young, T. (2011). An approach to formulate and model aerodynamic drag on aircraft landing gear. *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, including the AIAA Balloon Systems Conference and 19th AIAA Lighter-Than*. 1-21.
- [356] Romaniuk, S., Gosiewski, Z. ve Ambroziak, L. (2016). *A Ground control station for the UAV flight simulator*. acta mechanica et automatica, 10(1), 28-32.
- [357] Lum, C. (2009). *Coordinated searching and target identification using teams of autonomous agents*. Washington: University of Washington.
- [358] Zhao, X., Wang, Y., Zhang, W. ve Zhang, X. (2017). Personal Desktop-Level Jet Fighter Simulator for Training or Entertainment. *International Conference on Intelligent Robotics and Applications*. Springer, 417-427.
- [359] <https://cpu.userbenchmark.com/Compare/Intel-Core-i9-12900KS-vs-AMD-Ryzen-5-2400G/m1821519vsm433194> (Erişim tarihi: 04.05.2022)
- [360] https://www.cpu-monkey.com/en/compare_cpu-amd_ryzen_threadripper_pro_3995wx-1899-vs-amd_ryzen_5_2400g-858 (Erişim tarihi: 04.05.2022)

ÖZGEÇMİŞ

Genel Bilgiler

Adı Soyadı : Ahmet GEZER

Yabancı Dili : İngilizce

Öğrenim Durumu

Doktora : Eskişehir Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü
Sivil Havacılık

Yüksek Lisans : Gazi Üniversitesi Bilişim Enstitüsü
Elektronik Bilgisayar Eğitimi

Lisans : Gazi Üniversitesi Teknik Eğitim Fakültesi
Bilgisayar Sistemleri Öğretmenliği

Lise : Fatih Sultan Mehmet Yabancı Dil Ağırlıklı Lisesi, Ankara

Mesleki Geçmişi

Ankara Nallıhan Fettah Güngör Teknik (Öğretmen, Bilişim (2005-
ve Endüstri Meslek Lisesi ve MEM Teknolojileri Alan Şefi) 2010)

Sakarya Geyve Mesleki ve Teknik (Öğretmen, Bilişim (2010-
Eğitim Merkezi Teknolojileri Alan Şefi) 2014)

Sakarya Karapürçek Atatürk Ortaokulu (Okul Müdürü) (2014-
2018)

Sakarya Akyazı Atatürk Ortaokulu (Öğretmen) (2018)

Ölçme, Değerlendirme ve Sınav (Öğretmen) (2018-
Hizmetleri Genel Müdürlüğü - Veri Devam
Analizi, İzleme ve Değerlendirme DB. Ediyor)

Ödülleri

01/06/2007	Teşekkür Belgesi	Ankara İl Milli Eğitim Müdürlüğü
18/06/2007	Takdir Belgesi	Kaymakamlık
22/02/2008	Takdir Belgesi	Kaymakamlık
18/02/2010	Teşekkür Belgesi	Ankara İl Milli Eğitim Müdürlüğü

04/03/2010	Takdir Belgesi	Kaymakamlık
02/03/2020	Başarı Belgesi	Bakan
21/01/2022	Başarı Belgesi	Bakan
01/06/2022	Başarı Belgesi	Bakan

Uluslararası Dergilerde Yayımlanan Makaleler

Gezer, A., Turan, Ö. ve Baklacioğlu, T. (2016). UAV path planning using segmented cellular evolutionary algorithm. *International Journal of Sustainable Aviation*, 2(3), 222-234.

Bildiriler

Gezer, A., Turan, Ö. ve Baklacioğlu, T. (2015). UAV path planning using segmented cellular evolutionary algorithm. *International Symposium on Sustainable Aviation (ISSA-2015)*'de sunulan poster. Türkiye: Anadolu Üniversitesi

Gezer, A., Turan, Ö. ve Baklacioğlu, T. (2022). Hücresel Genetik Algoritma ile İnsansız Hava Araçlarında Yol Planlama. *Proceeding 3rd International Conference on Applied Engineering and Natural Sciences*. Konya, Türkiye.

Gezer, A., Turan, Ö. ve Baklacioğlu, T. (2022). İnsansız Hava Aracı Performansına Dayalı Yol Planlama. *Proceeding 3rd International Conference on Applied Engineering and Natural Sciences*. Konya, Türkiye.