



Path planning based on unmanned aerial vehicle performance with segmented cellular genetic algorithm

Ahmet Gezer^{1*}, Önder Turan^{2,3}, Tolga Baklacioğlu³

¹Eskisehir Technical University, Institute of Graduate Programs, 26555, Eskisehir, Türkiye

²Visiting Professor, Istanbul Commerce University, 34445, Istanbul, Türkiye

³Eskisehir Technical University, Faculty of Aeronautics and Astronautics, 26450, Eskisehir, Türkiye

Highlights:

- Aircraft-specific performance-based path planning
- Overprotective genetic crossover and mutation operators
- Predefined fixed initial population and segmented chromosome structure

Keywords:

- Path Planning
- Trajectory Planning
- Genetic Algorithm
- Evolutionary Algorithm
- UAV

Article Info:

Research Article

Received: 05.08.2022

Accepted: 03.02.2024

DOI:

10.17341/gazimmfd.1156817

Correspondence:

Author: Ahmet Gezer

e-mail:

ahmetgezer@gmail.com

phone: +90 505 391 9896

Graphical/Tabular Abstract

An important part of UAV technological development consists of improvements in the scope of path planning. In proposed algorithm, the initial population containing predefined fixed paths covering whole map (Figure A-(a)) are improved with the overprotective crossover and mutation strategy specific to the segmented chromosome structure (Figure A-(b)). So, the Segmented Cellular Genetic Algorithm (scGA) can preserve high population diversity compared to a conventional Genetic Algorithm (GA). After only 30 iterations, it can obtain a rich set of solutions containing many alternative paths (Figure A-(c)). It has had great success in improving the best individual and the entire population (Figure A-(d)).

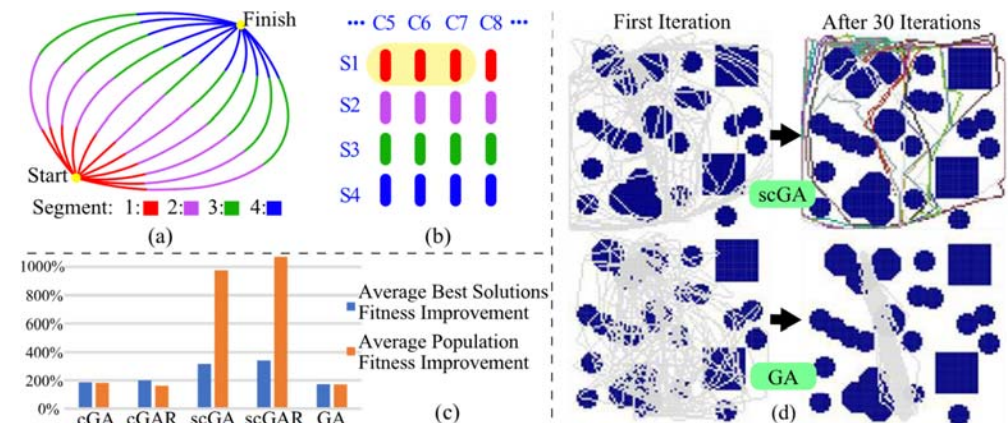


Figure A. (a) predefined fixed initial population (b) cellular crossover on segmented chromosome structure (c) comparison of fitness improvement (d) iteratively generated paths

Purpose: 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. This study aims to provide an option that prioritizes airtime with a fast and effective method.

Theory and Methods: The novel overprotective scGA which has a fixed initial population and segmented chromosome structure is proposed. 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. With these data, algorithm focuses on a realistic assessment of UAV movement in 3D environment. A cellular genetic crossover and mutation strategy customized to the segmented chromosome structure with an overprotective approach were used to improve the predefined fixed initial population.

Results: The novel algorithm 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 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. Despite the GA percentage of invalid results up to 50%, many variants of scGA performed far better with 0% invalid result success in all maps and scenarios.

Conclusion: It has been seen that scGA based on aircraft performance with predefined fixed initial population, segmented chromosome structure and overprotective approach is fast and effective in planning safe and fuel saving paths.



Parçalı hücresel genetik algoritma ile insansız hava aracı performansına dayalı yol planlama

Ahmet Gezer^{1*}, Önder Turan^{2,3}, Tolga Baklacioğlu³

¹Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, 26555, Eskişehir, Türkiye

²Ziyaretçi Profesör, İstanbul Ticaret Üniversitesi, 34445, İstanbul, Türkiye

³Eskişehir Teknik Üniversitesi, Havacılık ve Uzay Bilimleri Fakültesi, 26450, Eskişehir, Türkiye

Ö N E Ç İ K A N L A R

- Uçağa özel performansa dayalı yol planlaması
- Aşırı korumacı genetik çaprazlama ve mutasyon operatörleri
- Önceden tanımlanmış sabit başlangıç popülasyonu ve bölümlere ayrılmış kromozom yapısı

Makale Bilgileri

Araştırma Makalesi

Geliş: 05.08.2022

Kabul: 03.02.2024

DOI:

10.17341/gazimmfd.1156817

Anahtar Kelimeler:

Yol planlama,
yörünge planlama,
genetik algoritma,
evrimsel algoritma,
İHA

ÖZ

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 görülmüştür.

Path planning based on unmanned aerial vehicle performance with segmented cellular genetic algorithm

H I G H L I G H T S

- Aircraft-specific performance-based path planning
- Overprotective genetic crossover and mutation operators
- Predefined fixed initial population and segmented chromosome structure

Article Info

Research Article

Received: 05.08.2022

Accepted: 03.02.2024

DOI:

10.17341/gazimmfd.1156817

Keywords:

Path planning,
trajectory planning,
genetic algorithm,
evolutionary algorithm,
UAV

ABSTRACT

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. For every speed and altitude that the UAV can fly; fuel data of cruise, climb and descent phases are used in the path planning algorithm. 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 study, 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 seen 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.

*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : *ahmetgezer@gmail.com, onderturan@eskisehir.edu.tr, tbaklacioğlu@eskisehir.edu.tr / Tel: +90 505 391 9896

1. Giriş (Introduction)

İnsansız hava araçları (İHA), havacılık ve uzay teknolojilerinin yakın dönemdeki hızlı gelişimine paralel olarak çok geniş bir kullanım alanına ulaşmış ve artan ihtiyaçlar doğrultusunda bilimsel faaliyetler ivme kazanmıştır. Artan ilgiye bir örnek olarak Birleşik Krallık İş Enerji ve Endüstriyel Strateji Departmanı (BEIS) tarafından Düzenleyici Ufuklar Konseyi (RHC) aracılığıyla bir rapor (2021) yayınlanmıştır. Raporda, önümüzdeki on yıl içinde çoklu İHA, çarpışmadan kaçınma ve otomatikleştirilmiş "güvenli en düşük maliyetli yol planlaması"nın insansız hava araçlarının bir özelliği olmasının beklendiği belirtilerek İHA'lardaki yeni fırsatlara dikkat çekilmiştir [1].

Yol planlamada, yolculuk süresini en aza indirmek ve yakıt optimizasyonu arasındaki tercih, operasyon önceliklerine göre belirlenir. Bir kurtarma operasyonunda hız önemli olabilirken, keşif veya gözetleme operasyonlarında daha uzun süre havada kalma tercih edilebilir. Bu çalışmada operatöre, havada kalma süresini önceleyen bir alternatif sunulmaktadır.

İHA'ların enerji tüketimini en aza indirmek literatürde yer alan önemli konulardandır. İHA, sabit bir irtifada aynı hızla yol aldığı enerji sarfiyatının uçuş mesafesiyle orantılı olduğu gösterilebilir. Bu nedenle enerji sarfiyatını en aza indirmek için uçuş mesafesinin kısaltılması önerilir. Üç boyutlu (3D) yol planlamada ise yakıt tüketimi, aracın alçalma ve yükselme manevraları ile süreden bağımsız etmenlerden büyük oranda etkilenir.

İHA'lar için yol planlaması, genellikle en kısa mesafeyi seçmek amacını taşır ancak bu süreçte bazen ekonomik faktörlerin ihmal edildiği gözlemlenmiştir. Seçilen rotanın en kısa olması önemli olabilir, ancak hava aracının performansını değerlendirdiğimizde bazen daha uzun rotalar, ekonomik açıdan daha avantajlı olabilmektedir. ATR'nin yakıt ekonomisi raporu; daha yüksek seyir irtifası, düşük hızlı tırmanma ve düşük itki ile alçalma gibi yöntemlerin ATR 72-500'ün uçuş süresinden 8 dakika ödenerek 110 kg yakıt tasarrufu sağlayabileceğini göstermiştir [2]. Eurocontrol'un 71 temel uçak türüne dayanan çalışması da ekonomi öncelikli bir planlamaya örnek olarak gösterilebilir [3, 4]. Uçuş rotalarının belirlenmesinde navigasyon veritabanı, coğrafi bilgi sistemi ve çeşitli uçakların gerçek performans verilerine dayalı bilgisayarlı uçuş rotası parametreleri de kullanılmıştır [5]. Pervaneli bir uçağın sabit hızda ihtiyaç duyduğu yakıt miktarını ele alan bir diferansiyel denklem dördüncü dereceden Runge-Kutta yöntemiyle tahmin edilmiştir. Bu çalışma, görece yüksek hassasiyetle harcanan yakıt miktarını ortaya koymaktadır [6, 7]. Literatürdeki çoğu çalışma genellikle yakıt tüketiminin kabaca bir tahminine odaklanmıştır [7]. Genel olarak hava aracına özgü uçuş performans özellikleri yerine genel varsayımlar kullanılmaktadır. Bu durum, hesaplama hassasiyetinin düşük olmasına neden olmaktadır.

Rüzgârın avantajlı bir şekilde kullanımı için optimize edilen uçuş yolları; uçuş süresini ve menzili artırabilecek etkili bir yöntem olarak önerilmiştir [8, 9]. "Sonsuz uçuş" fenomeni için otomatik pil değişim yöntemleri ve rüzgârın yakıt ekonomisine etkileri üzerine yapılan çalışmalar dikkat çekicidir [11]. İHA'ların pil yönetimi ve değişimi için mobil hava gözetim platformları ile ilgili çalışmalar yapılmaktadır [10]. Rüzgârın yakıt ekonomisi üzerindeki etkileri [12] ve formasyon uçuşu [13] araştırmaları yakıt verimliliğini artırmaya yönelik çabaların bir yansımasıdır.

Yol planlama uygulamalarının ilk aşamasında, algoritmanın çalışma alanını tanıması ve yapılandırma alanını oluşturması önemlidir [14]. Bu kapsamda, görünürlük diyagramları, voronoi diyagramları ve

hücresel ayrıştırma gibi uzay ayrıştırma yöntemleri kullanılmaktadır. Yapılandırma alanının ızgara tabanlı bir şekilde gösterildiği çalışmalarda hücre boyutlarının aracın güvenli hareket edebileceği kadar geniş olması tercih edilir. Eşit kare hücre ızgara yönteminin kullanılması ile hesaplama kolaylığı hedeflenir [15]. Hücresel ayrıştırma, çeşitli verilerin depolanması bakımından oldukça kullanışlı bir özelliktir. Görsel algıya sahip bir mikro İHA uygulamasında, engelleri aşmak ve rüzgâr bilgilerini değerlendirerek etkili yollar oluşturmak için çoklu çözünürlüklü hücresel ayrıştırma yöntemi kullanılmıştır [16].

Yapılandırma alanındaki yapay zekâ teknikleriyle gerçekleştirilen arama, ikinci aşamayı oluşturur. Bu aşamada kullanılan arama algoritmaları, genellikle en kısa yolu değil, geometrik temsil ile belirlenen popülasyon içindeki en kısa yolu seçebilir [14]. Bu evrede çeşitli algoritmaların kullanımı mümkündür [17]. Genetik Algoritma (GA); çok boyutlu, geniş ve karmaşık bir arama alanına sahip, matematiksel bir modelle ifade edilemeyen, kesin bir çözüm yöntemi bulunmayan durumlarda etkili ve kullanışlı olan stokastik bir optimizasyon yöntemi olarak öne çıkmaktadır [18]. Rastgele sayılarla oluşturulan başlangıç popülasyonu [19], sıralı çaprazlama [20], iki noktalı çaprazlama [21], tek noktalı çaprazlama [19, 22], bitişik çaprazlama [23], sezgisel çaprazlama [24], yeni genetik operatörler [19], araya ekleme mutasyonu [20], tek tip mutasyon [22], tek tip olan ve olmayan mutasyon operatörlerinin bir arada kullanılması [24]; GA'ların genel çerçevede aynı olmalarına rağmen küçük değişikliklerle çok farklı sonuçlar verebileceğinin örnekleridir.

3D çözüm uzayında radar kısıtları, yakıt tüketimi ve ortalama irtifa faktörlerini değerlendiren Parçacık Sürü Optimizasyonu (PSO) ve GA'nın birlikte kullanıldığı çalışmalar da dikkat çekicidir [25]. GA'nın, ortalama irtifa ve tehlike bölgelerinden kaçınma perspektifinde PSO'ya göre daha etkileyici yönlere ürettiği ortaya konmuştur. Adaların potansiyel tehdit unsurları olarak ele alındığı bir uygulama senaryosunda, aday yolların matematiksel modelini oluşturmak için Diferansiyel Evrimsel Algoritma (DEA) ile yapılan çalışma [26], GA ile İHA'nın yer merkezli radar istasyonları tarafından tespit edilmesini engellemeyi hedefleyen çevrimiçi yol planlama çalışması [27], yerel optimuma sıkışma riskini azaltan ve yakınsama hızını artıran bir Parçacık GA önerisi [28], evrimsel algoritmaların İHA yol planlama alanında sunulan örnek çalışmalarıdır.

Hücresel Genetik Algoritma (cGA), GA'nın yapısal paralelleştirme yaklaşımı içinde ince taneli kategoride sınıflandırılacak özel bir türdür. Geleneksel GA uygulamalarında her birey, bütünsel bir uzayda konumlanan bir aday çözümü temsil eder [29]. Bu ayrılmış yapılar, hücre adı verilen birimler halinde tanımlanır ve komşu hücrelerle düzenli etkileşimlere izin verilir. "Hücresel Evrimsel Algoritma" terimi, Whitley tarafından 1993 yılında önerilmiştir [30] ancak bu anlayıştaki algoritmalar daha önce de çalışılmıştır [31, 32]. Dağıtılmış bir popülasyonun kullanılması, genetik farklılaşmanın popülasyon izolasyonu ile önemli ölçüde arttığı teorisine dayanmaktadır [33].

cGA daha önce çeşitli problemlere uygulanmıştır. Araç rotalama problemi [34], gerçekleştirilebilirlik (satisfiability) problemleri (SAT) [35, 36] ve çok amaçlı optimizasyon [37] bunlardan birkaçıdır. Bir görüntü sınıflandırma ve sıralama problemi içinde Gezin Satıcı Problemi'ne (TSP) benzeyen bir yaklaşım kullanılarak cGA ile çözüm getirilmiştir [38]. Ayrıca, azami süre ya da mesafe kısıtları içeren bir kapasite kısıtlı araç rotalama problemi (VRP), cGA kullanılarak başarıyla ele alınmıştır [34]. Bu çalışmalarda mahalle yapısı kromozomların özelliklerine göre değil, rastgele yerleştirilen kromozomlar arasında oluşturulmuştur. Bu çalışmada sabit başlangıç popülasyonu kullanan İHA performansına dayalı Parçalı Hücresel

Genetik Algoritma önerilmiştir. Klasik yol planlama probleminin; İHA kinematik özellikleri ve performans karakteristiği dikkate alınarak tanımlanan konfigürasyonda, gerçek ortamdan uyarlanmış engel haritaları üzerinde ve havada kalma süresi önceliği ile gerçek zamanlı çalışmaya entegre edilebilir bir hızda çözülmesi amaçlanmaktadır

Böylece uzun havada kalma süresi ile daha uzun operasyonel görevlerin gerçekleştirilmesine imkân sağlanmaktadır. 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.

Parçalı kromozom yapısı ve sabit başlangıç popülasyonu ile oluşturulan hücresel evrimsel algoritmanın yer aldığı çalışmadan [39] sonra sadece hücresel genetik algoritmanın genetik algoritma ile karşılaştırmasının yapıldığı tek haritalı ceza puanına dayalı [40] ve hava aracı performansına dayalı [41] iki bildiri ile çalışma genişletilmiştir. Seçilmiş bir İHA'ya özel performans verileri esas alınarak yol planlama, aşırı korumacı değiştirme operatörü (Overprotective Replacement Operator-OPRO) ve çok süreçli algoritma yapısı ile Hücresel Genetik Algoritma (cGA) ve Parçalı Hücresel Genetik Algoritma (scGA) yöntemleri ilk defa bu çalışmada birlikte kullanılmıştır. Çalışmanın ilk aşamalarına [39] bulunmayan OPRO ve hava aracı performansına dayalı planlama çok büyük bir ilerleme sağlamıştır. Çoklu süreç desteği ve farklı senaryolar ile testler yapılması önceki çalışmalarda bulunmayan kazanımlardır.

Çalışma kapsamında geliştirilen web uygulaması elde edilen en uygun rotayı grafiksel bir formatta kullanıcıya sunmaktadır. Uygulama arayüzü, kullanıcılara çeşitli parametreleri özelleştirme, yeni haritalar oluşturma ve elde edilen sonuçları detaylı bir şekilde inceleme imkânı sunar. Bu interaktif platform, algoritmanın performansını değerlendirme ve uçuş planlamasını daha etkili bir şekilde tasarlama fırsatı sunar. Böylece, hava aracının en ekonomik rotasını belirleme süreci kullanıcı odaklı bir şekilde yönetilebilir ve optimize edilebilir. Ayrıca, İHA'nın simülasyonu için FlightGear kullanılarak eş zamanlı olarak planlanan ve izlenen yollar Google Earth üzerinde takip edilebilmektedir.

Çalışmanın ilerleyen kısmı şu şekilde düzenlenmiştir: İkinci bölümde, problemle ilgili deneysel yöntem, algoritmanın özellikleri ve çalışmanın literatürle olan ilişkisi ayrıntılı bir şekilde sunulmaktadır. Üçüncü bölümde, test konfigürasyonları detaylı bir şekilde açıklanmakta olup, elde edilen test sonuçları ve bu sonuçlara yönelik yapılan tartışmalar yer almaktadır. Elde edilen sonuçların kapsamlı bir analizi ise dördüncü bölümde sunulmaktadır.

2. Deneysel Yöntem (Experimental Method)

Bu bölümde, İHA yol planlama çözümü detaylı bir şekilde incelenmektedir. Çalışmada kullanılan çevresel modelleme, genetik algoritmalar ve popülasyona dayalı teknikler ile engellerin temsili gibi konular ele alınmıştır. Çalışmanın literatürdeki yeni netleştirilmiştir. Algoritmanın ayrıntıları sahte kodlar ve denklemlerle gösterilmiştir. MATLAB Simulink modelinden elde edilen uçuş verilerini kullanan İHA'ya özgü planlama vurgulanmıştır. Başlangıç popülasyonuna ilişkin farklılıklar, algoritmaya ait operatör, araç ve parametreler açıklanmıştır. Simülasyon ve uygulama adımlarıyla birlikte çalışmanın detayları anlatılmıştır. Geliştirilen web tabanlı yazılım paketinin ayrıntıları ve sağladığı faydalar sunulmuştur.

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

2.1. Çevresel Modelleme (Environment Modeling)

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. Bu çalışmada kullanılan ızgara tabanlı yaklaşımda ortam hücrelere ayrılmıştır. Hücre boyutları, İHA'nın güvenli bir şekilde hareket edeceği büyüklükte tanımlanmaktadır [15].

Önerilen yaklaşımda hesaplama kolaylığı açısından ortam ayrıştırması aşamasında ızgara hücrelerinin genişliğinin asgari dönüş yarıçapından büyük olması esastır. Böylece, hava aracının ek bir hesaplama gerektirilmeyen harita üzerindeki tüm yolları güvenli bir şekilde dönebilmesi mümkün olmuştur. Test senaryolarında bu genişlik 500 m boyutunda tanımlanmıştır.

Zemini temsil etmek için koordinata göre yüksekliği gösteren 2D bir matris kullanılmıştır [25]. Web üzerinden erişime sunulmuş farklı harita uygulamaları, koordinatlardaki yükselti bilgilerine erişmek için farklı tekniklere imkân vermektedir. Tekil koordinat veya bir koordinatlar dizisinin yükselti değerlerine ulaşılabilir [42]. 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 [43]. Böylece, her birim alan içerisinde 64 noktanın azami değeri 0 birim alanın yükseltisi kabul edilmiştir. Sistem, istenilen koordinat aralığı için yüksek çözünürlükte bu işlemleri otomatik yapabilecek şekilde tasarlanmıştır.

Harita üzerinde tüm engel çeşitleri çokgenler şeklinde tanımlanabilmektedir. Bunlara bir örnek olarak radar tehditleri gösterilebilir. Her radarın parametreleri kendine özgüdür. Her radar, 3D silindirik bir tehdit alanına sahip olduğu varsayılarak basitleştirilmiştir [44]. Mobil tehditlerin tehlike oluşturduğu alanlar, diğer kısıtlarda olduğu gibi operatör tarafından belirlenebilir.

2.2. Hücresel Genetik Algoritma (Cellular Genetic Algorithm)

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 çalışmada cGA'nın başarılı yeni bir varyantı olan scGA önerilmiştir. Algoritmalar ve alt prosedürleri kategoriler halinde sahte kodlar verilerek detaylandırılmıştır.

cGA yapılandırılmış popülasyon yapısının başarılı bir örneğidir. Bir cGA, mekânsal olarak belirlenmiş sabit bir yapıya sahiptir. Her hücre, popülasyondaki bireylerden birini içermektedir. Her birey, yalnızca bir şablon ile belirlenmiş bir dizi komşu ile iletişime sahiptir. Uygulanan cGA algoritması Tablo 1'de, scGA algoritması Tablo 2'te ve geleneksel GA Tablo 3'te gösterilmiştir.

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 [45] her yinelemede kullanılan seçim ve varyasyon yöntemlerine göre farklılık göstermektedir [46].

Mahalle içinde uygulanan genetik operasyonlar sırayla (eşzamanlı) ya da paralel (eşzamanlı) gerçekleştirilebilir [47]. Tablo 1'de gösterilen cGA ve Tablo 2'de gösterilen scGA'da popülasyon, yeni popülasyona ait bireyler oluşturulduktan hemen sonra güncellendiği için eşzamanlı olarak sınıflandırılır. Böylece, yeni bireyler, ebeveynlerinin nesline ait olanlarla etkileşime girebilmektedir.

Tablo 1. cGA(R)

S	İşlev	S	İşlev
1	mahalleYapısınıOluştur(p)	10	$uygunluk \leftarrow$ uygunlukHesap($yavru$)
2	$p \leftarrow$ başlangıçPopülasyonuOluştur()	11	$p, i \leftarrow$ dahaİyiyeDeğiş($yavru, uygunluk$)
3	$uygunluk \leftarrow$ uygunlukHesap(p)	12	$yavru \leftarrow$ mutasyon ($yavru$)
4	<i>Döngü başlat</i> , durma koşulu sağlanana kadar:	13	yoluDüzeltil ($yavru$)
5	<i>Döngü başlat</i> , popülasyonun her bireyi (i) için:	14	$uygunluk \leftarrow$ uygunlukHesap($yavru, uygnlk$)
6	$mahalle \leftarrow$ komşularıHesapla(p, i)	15	$p, i \leftarrow$ dahaİyiyeDeğiş($yavru, uygunluk$)
7	$ebeveynler \leftarrow$ seçim($p, mahalle, uygunluk$)	16	<i>Döngü sonu</i>
8	$yavru \leftarrow$ çaprazlama ($ebeveynler, mahalle$)	17	<i>Döngü sonu</i>
9	yoluDüzeltil ($yavru$)	18	<i>bitir</i> \leftarrow enİyiBirey(p)

Tablo 2. scGA(R)

S	İşlev	S	İşlev
1	ızgaraYapılandırmasınıTanımla()	11	$u^2, p^3 \leftarrow$ uygunlukHesap(y^l)
2	mahalleYapısınıOluştur(p)	12	$p, i \leftarrow$ dahaİyiyeDeğiş(y^l, u^2, p^3)
3	$p \leftarrow$ başlangıçPopülasyonuOluştur()	13	$y^l \leftarrow$ mutasyon (y^l, p^3)
4	$u^2, p^3 \leftarrow$ uygunlukHesap(p)	14	yoluDüzeltil (y^l)
5	<i>Döngü başlat</i> , durma koşulu sağlanana kadar:	15	$u^2, p^3 \leftarrow$ uygunlukHesap(y^l)
6	<i>Döngü başlat</i> , popülasyonun her bireyi (i) için:	16	$p, i \leftarrow$ dahaİyiyeDeğiş(y^l, u^2, p^3)
7	$mahalle \leftarrow$ komşularıHesapla(p, i)	17	<i>Döngü sonu</i>
8	$ebeveynler \leftarrow$ seçim($p, mahalle, u^2$)	18	<i>Döngü sonu</i>
9	$y^l \leftarrow$ çaprazlama ($ebeveynler, mahalle, u^2, p^3$)	19	<i>bitir</i> \leftarrow enİyiBirey(p)
10	yoluDüzeltil (y^l)		Değişkenler: ¹ $yavru$, ² $uygunluk$, ³ $parcaUygunluk$

Tablo 3. Geleneksel GA (*conventional GA*)

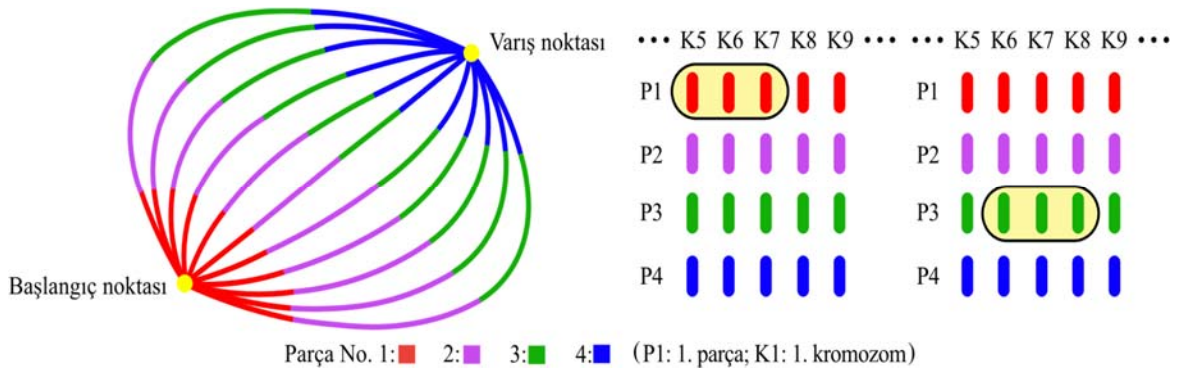
S	İşlev	S	İşlev
1	$p \leftarrow$ başlangıçPopülasyonuOluştur()	9	$p, i \leftarrow$ dahaİyiyeDeğiş($yavru, uygunluk$)
2	$uygunluk \leftarrow$ uygunlukHesap(p)	10	$yavru \leftarrow$ mutasyon ($yavru$)
3	<i>Döngü başlat</i> , durma koşulu sağlanana kadar:	11	yoluDüzeltil ($yavru$)
4	<i>Döngü başlat</i> , popülasyonun her bireyi (i) için:	12	$uygunluk \leftarrow$ uygunlukHesap($yavru, uygunluk$)
5	$ebeveynler \leftarrow$ seçim($p, uygunluk$)	13	$p, i \leftarrow$ dahaİyiyeDeğiş($yavru, uygunluk$)
6	$yavru \leftarrow$ çaprazlama ($ebeveynler$)	14	<i>Döngü sonu</i>
7	yoluDüzeltil ($yavru$)	15	<i>Döngü sonu</i>
8	$uygunluk \leftarrow$ uygunlukHesap($yavru$)	16	<i>bitir</i> \leftarrow enİyiBirey(p)

Tablo 4. Çok süreçli scGA(R) (*Multi-process scGA(R)*)

S	İşlev	S	İşlev
1	$veritabanı \leftarrow$ yakıt, parametre	6	$uygunluk, parcaUygunluk \leftarrow$ uygunlukHesap(p)
2	süreçleriÇalıştır(<i>süreçSayısı</i>)	7	$veritabanı \leftarrow$ görevleriBaslat($p, süreçSayısı$);
3	ızgaraYapılandırmasınıTanımla()	8	<i>Bekle: görevTamam</i> \leftarrow görevDurumu() \leftarrow $veritabanı$
4	mahalleYapısınıOluştur(p)	9	<i>Eğer görevTamam=1</i> 'se $p \leftarrow$ $veritabanı$ değil: <i>Bekle</i>
5	$p \leftarrow$ başlangıçPopülasyonuOluştur()	10	<i>Bitir</i> \leftarrow enİyiBirey(p)
<i>Süreç Alt Uygulaması</i>			
1	$parametre \leftarrow$ $veritabanı$	12	$p, i \leftarrow$ dahaİyiyeDeğiş($yavru, uygunluk, parcaUygunluk$)
2	<i>Bekle: görevVar</i> \leftarrow görevDurumu() \leftarrow $veritabanı$	13	$yavru \leftarrow$ mutasyon ($yavru$)
3	<i>Eğer görevVar=0</i> ise <i>Bekle değilse</i> :	14	yoluDüzeltil ($yavru$)
4	$p \leftarrow$ $veritabanı$	15	$uygunluk, parcaUygnlk \leftarrow$ uygunlukHsp($yavru$)
5	<i>Döngü başlat</i> , durma koşulu sağlanana kadar:	16	
6	<i>Döngü başlat</i> , popülasyonun her bireyi (i) için:	17	$p, i \leftarrow$ dahaİyiyeDeğiş($yavru, uygunluk, parcaUygnlk$)
7	$mahalle \leftarrow$ komşularıHesapla(p, i)	18	<i>Döngü sonu</i>
8	$ebeveynler \leftarrow$ seçim($p, mahalle, uygunluk$)	19	$veritabanı \leftarrow$ dahaİyiYiVeritabanınaYükle(p)
9	$yavru \leftarrow$ çaprazlama ($ebeveynler$)	20	$p \leftarrow$ dahaİyiYiVeritabanındanAl() \leftarrow $veritabanı$
10	yoluDüzeltil ($yavru$)	21	<i>Döngü sonu</i>
11	$uygunluk, parcaUygnlk \leftarrow$ uygunlukHsp($yavru$)		$veritabanı \leftarrow$ göreviBitir(p)

Bu çalışma kapsamında karşılaştırılan cGA(R) (Tablo 1), scGA(R) (Tablo 2) ve GA (Tablo 3) algoritmalarının çalışma şekli aşağıda özetlenmiştir. Öncelikle cGA(R) algoritmasında seçilen hava aracına ait parametreler ve yakıt bilgileri okunur. Seçilen başlangıç ve bitiş noktaları alınır. Çok süreçli algoritmalarda hava aracı parametre, yakıt bilgileri ve harita bilgileri veritabanına yazılır ve süreç uygulamaları çalıştırılır. Hesaplama süresi başlatılır. Başlangıç popülasyonu oluşturulur. cGA ve scGA'da, sabit yollar veritabanından okunur ve başlangıç/varış noktalarına göre ayarlanır. Tüm popülasyonun uygunluk değeri hesaplanır. cGA(R) ve scGA(R)'de mahalle yapısı oluşturulur. 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 istatistik bilgilerin hesaplanmasına geçilir. İstatistikler hesaplanır, kaydedilir ve görüntülenir.

scGA, GA ve cGA algoritmalarında tekil süreçlerdeki farklılıklar korunmakla birlikte çok süreçli mimari, benzerlik gösterir: Örnek bir algoritma Tablo 4'te gösterilmiştir. Öncelikle, süreç görevlerinin başlama bilgisi veritabanına yazılır. Başlangıç popülasyonu, süreç sayısı kadar parçaları 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 çalıştırdıktan 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 popülasyonda güncellenir. 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 yansayan bir farklılığa sebep olmaktadır.



Şekil 1. cGA ve scGA'nın başlangıç popülasyonu ile scGA'daki kromozom (yol) parçaları ve L3 mahalle yapısının parçalı kromozomlarda kullanımı
(Initial population of cGA and scGA and segments of chromosomes (paths) and use of L3 neighborhood structure in segmented chromosomes)

2.2.1. Mahalle yapısı (Neighborhood structure)

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 [48]. Mahallelerin tanımlanmasının, C_n kompakt formu ve L_n lineer formu olarak tanımlanan iki klasik yolu bulunmaktadır [29]. Burada n her komşuluğun boyutunu gösterir. Literatürde birçok ızgara şekli ve komşuluğu önerilmiş ve incelenmiştir. Bir cGA'nın davranışı, mahalle boyutu ve geometrisine bağlıdır.

Önerilen algoritmalarda komşu kromozomlar koordinatlarına göre belirlenmektedir yani kromozom komşuluğunun başlangıçtaki ortak özelliklere göre belirlendiği bir yaklaşım ile başlangıç popülasyonuna göre birbirine yakın yollar mahallelerde de aynı yakınlıkta yer alır.

2.2.2. Kromozom şeması (Chromosome scheme)

Genel olarak kromozomlar bir bütün olarak kabul edilir. Önerilen yöntemde ise parçalı bir kromozom yapısı tercih edilmiştir. Farklı GA türlerinde parçalı kromozom yapısının örnekleri bulunmaktadır [49]. Kromozom parçalarına uygunluk değeri verilebilir [50]. 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. Parça uygunluk değeri kullanımı, yolun sadece bir kısmında engel bulunduğu durumlarda 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.

scGA ve cGA arasındaki en önemli fark, mahalle yapısının bireylerde kullanıldığı gibi kromozom parçalarında da kullanılmasıdır. Böylece, aynı mahalledeki kromozom parçaları referans alınarak çaprazlama yapılabilir. Şekil 1'de L3 mahalle yapısının parçalı kromozomlarda kullanımına bir örnek gösterilmiştir [39]. Her kromozomun, sağında ve solunda yer alan kromozomlar ile üçlü bir mahalle oluşturduğu daire içerisinde gösterilerek sembolize edilmiştir.

cGA'da yapılandırma ızgarası (popülasyon büyüklüğü ile ilişkili olarak) 55×1 birey boyutundadır. scGA'daki yerleşim kromozomlara

göre olduğundan 55x3, 55x4 gibi ızgara yapılandırılmaları kromozom uzunluğuna göre otomatik belirlenmektedir. ızgara yapılandırmasındaki satır sayısı, ortadaki (sabit başlangıç popülasyonunda en düz yola sahip) kromozomun düğüm sayısı ile orantılı olarak belirlenir. Test konfigürasyonunda satır sayısı: düğüm sayısı 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.

2.2.3. Genetik operatörler (Genetic operators)

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.

Yavruların ebeveynleri ile karşılaştırılarak sonraki nesle hangi bireylerin taşınmasına karar veren değiştirme operatörü yaygın olarak, daha yüksek uygunluk değerine sahip bireylerin bir sonraki nesle aktarılmasını sağlar. cGA ve GA'da yaygın yaklaşıma uygun biçimde yavrular daha iyi oldukları takdirde yeni nesle aktarılır. 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 bakıldığında yüksek kromozom parçası sayısı ebeveyn kromozoma kıyasla azalmışsa yavru da bir iyileşme olduğu değerlendirilerek ceza puanı 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 [51] veya popülasyonun belirli sayıda iyi bireyin bir sonraki nesli doğrudan etkilemesi [52] işlemleridir. Önerilen algoritmalarda uygulanan operatör farklı bir konsepttedir.

Bu çalışma kapsamında geliştirilen yazılımda; cGA(R) ve GA ile scGA(R) algoritmalarındaki 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. Böylece çaprazlamaya katılacak kromozom parçasının daha iyi olması amaçlanır.

Seçilmiş olan kromozomlarda çaprazlama noktaları aşağıdaki şekilde 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 iki yöntem izlenir.

1. Yöntem: 2. kromozomun başlangıç ve bitiş genlerinin her birisi 1. kromozomdaki eşlerinin belirli bir oranda öncesinde veya sonrasında olacak şekilde rastgele seçilir. Bu oran test konfigürasyonunda %70 olarak belirlenmiştir. Böylece çaprazlanacak iki kromozomda seçilen noktaların (2. yönteme) nispeten yakın bölgede olması sağlanır.

2. Yöntem: Diğer algoritmalarda olduğu gibi kromozomun ilk geni ve son geni arasından rastgele seçilir.

Test konfigürasyonunda bu yöntemler yarı yarıya bir olasılıkla gerçekleşir. Yöntemlerin birbirine üstünlükleri bu çalışmada incelenmemiş, performansa etkisi öngörülerek sonraki çalışmalara kolaylık sağlanması amacıyla algoritma içinde tanımlanmıştır.

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 sonrasında kesişim olursa yol düzeltilir. Böylece, hava aracı kinematik kısıtlarına uyumsuz bir yola izin verilmez.

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. Bununla birlikte kromozom uygunluk değeri, ceza puanından küçükse veya ceza puanı olsa dahi parça uygunluk değerlerinde cezadan yüksek kromozom parçası sayısı ebeveyn kromozoma kıyasla azalmışsa yavru tercih edilir ve değişim yapılır.

Mutasyon operatörleri, her yinelemede bireylere 0.1 ihtimal ile uygulanır. Öncelikle kromozom üzerinde rastgele bir gen seçilir. scGA(R)'de mutasyona uğrıtılacak nokta ters rulet yöntemiyle seçilen kromozom parçası üzerinde rastgele belirlenir. Bu genin pozisyonundan, 1 ile "mutasyon aralığı" arasında rastgele üretilecek bir sayı kadar uzaklığı kromozomdan silinir. Silinen genin dikey ve yatay alellere (koordinatlarına) ayrı ayrı, 1 ile "mutasyon aralığı" arasında rastgele uzaklıkta oluşturulan yeni gen, yol sürekliliği sağlanarak silinen bölgede kromozoma bağlanır. Yavru kromozom uygunluk değeri daha iyi ise ebeveyn ile değiştirilir.

Yukarıda çaprazlama operatöründe de açıklandığı gibi mutasyonda da scGA(R)'de aşırı korumacı bir yaklaşım ile OPRO, yavru kromozom uygunluk değeri ancak ceza puanından küçükse veya parça uygunluk değerlerinde ceza puanından yüksek kromozom parçası sayısı ebeveyn kromozoma kıyasla azalmışsa değişimi gerçekleştirir.

2.2.4. Amaç fonksiyonu (Fitness function)

Amaç fonksiyonu şu şekilde tasarlanmıştır:

Denkleminde (Eş. 1) gösterilen d , başlangıç noktasından (x_s, y_s) varış noktasına (x_f, y_f) öklid mesafesidir.

$$d = \sqrt{(x_s - x_f)^2 + (y_s - y_f)^2} \quad (1)$$

$$P_d = \begin{cases} 0 & \text{yol engelsizse} \\ c_1 & \text{yol üzerinde engel veya erişimi imkânsız yükselti varsa} \end{cases} \quad (2)$$

Yolun tehditlerle kesişip kesişmediğinin belirlenmesi önemlidir. Amaç fonksiyonuna göre kullanılması mümkün olmayan yolların maliyeti, mümkün yolların maliyetinden daha yüksek olmalıdır. Bu amaçla kullanılacak P_d ceza puanının hesaplanmasında (Eş. 2); yol tehditlerle kesişiyorsa kullanılacak c_1 ceza sabiti, geçerli yolların uygunluk değerlerinin ulaşamayacağı bir değer olarak uygulanan testlerde 1000 değeri kabul edilmiştir.

Amaç fonksiyonları, farklı veri kaynaklarını kullanabilir. Yaygın kısıt işleme tekniği ceza puanı yöntemidir [53]. Literatürde çoğu çalışmada hava aracının yükselme durumunda bir ceza puanı eklenmektedir [54, 55]. Önerilen algoritalarda tercih edilen yöntem, hava aracı performans modeli temelinde bir yaklaşım ile hava aracı hareketinin gerçeğe yakın bir değerlendirilmesine odaklanır.

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

Bu çalışma kapsamında önerilen yöntemler için uygunluk değeri (amaç) fonksiyonu f_c denklemi (Eş. 3) 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 tüketimi hesaplanan 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 amaçlanmıştır. Uygunluk değerinin daha doğru bir değer üretebilmesi için V_i 'nin, yer hızı olarak 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.

Çalışmamızda, Bölüm 2.4'te açıklanacağı üzere İHA uçuş verileri, hava aracının MATLAB Simulink 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.

Önerilen yaklaşım ile 3D yollar üretilmesi hedeflenmektedir, sabit irtifa (2D) üretilen yolda yeryüzü yükselteleri veya engeller dikkate alınarak irtifa düzenlenmesi yapılmaktadır [56]. İHA kinematik kısıtları dikkate alınarak yükselme ve alçalma gereksinimlerine göre 3D yollar oluşturulmaktadır. İrtifa belirlenirken hava aracının en ekonomik uçaacağı irtifa ve hız tercih edilmektedir. Tırmanma ve alçalma oranları da aynı şekilde seçilmektedir. Algoritmada, çaprazlama ve mutasyon işlemlerinden sonra, irtifa ayarlama prosedürü çalışmakta sonrasında uygunluk değeri hesaplanmaktadır.

Tablo 5'te sahte kodları verilen amaç fonksiyonunda 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 yüksekte 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. Seçilen hava aracı karakteristiğine göre irtifa azaldıkça daha ekonomik bir uçuş gerçekleştiği için ek bir kontrol yapılmaz.

Yolun her noktasında imkânsızlığa düşüp düşmediği kontrol edilir. Eğer nokta, başlangıç veya varış noktasına göre yükselinemez veya alçalmaz 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 değışkene eklenir. Böylece, her nokta kontrol edilmiş olur. Zemin yüksekliğine, istenen asgari yerden yükseklik (AGL) değeri eklenmiştir.

Tepe noktaları dizisi büyükten küçüğe 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. Bu

Tablo 5. Hava aracı performans modeline dayalı amaç fonksiyonu (Fitness function based on aircraft performance model)

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

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.

2.2.5. Başlangıç popülasyonu (Initial population)

Başlangıç popülasyonun 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. GA'da başlangıç popülasyonu birçok çalışmada genellikle rastgele seçilir [57]. 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. Bu kapsamda bir çalışmada başlangıç popülasyonu için çeşitli algoritmalar kullanılarak oluşturulmuş 100 yol ve 95 rastgele yol kullanılmıştır [58]. Bir diğer çalışmada, başlangıç popülasyonunun kalitesini artırmak amacıyla, tüm topografyanın taranmasını sağlayan bir ön-işlem yöntemi olarak iskeletizasyon algoritmasının kullanıldığı belirtilmiştir. [59].

Önerilen yöntemde, başlangıçtan varış konumuna düz çizgi olan en kısa yol ile sağında ve solunda 20'er adet olmak üzere yarım yay şeklinde yollar sabit başlangıç popülasyonu olarak kabul edilmektedir (Şekil 1) [40].

Başlangıç popülasyonunun boyutu, değiştirilebilir test konfigürasyonunun bir parçasıdır. cGA ve scGA'da başlangıç popülasyonunun çözüm uzayını kapsayacak bir şekilde önceden belirlenmiş olması yöntemin olmazsa olmaz bir özelliği iken popülasyon boyutu ve rastgele eklenen popülasyon boyutu ayarlanabilir. Yapılan testlerde; GA, scGAR ve cGAR algoritmalarında 55 rastgele, cGA ve scGA için ise 41 sabit 14 rastgele başlangıç popülasyonu oluşturulmaktadır. cGA ve scGA'da rastgele popülasyon eklenerek mahalle içi çaprazlamalardaki stokastik etki artırılmak istenmiştir. Eklenen rastgele oluşturulmuş

bireylerin aralarında eşit mesafe olması sağlanarak popülasyona homojen dağılmasına özen gösterilir. Testlerde kabul edilen 41 sabit popülasyona karşın 14 rastgele popülasyon konfigürasyonunda, 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. Sıralamanın amacı kromozomların benzer çözümler ile mümkün olduğunca aynı mahallede olmalarını sağlamaktır.

2.2.6. Parametreler (Parameters)

Önerilen algoritmada parametreler Tablo 6'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 iyiyse değiştirilir buna ek olarak scGA'da OPRO kullanılmaktadır.

2.3. Hava Aracı Performansı (Aircraft Performance)

Yol planlayıcının seyir, yükselme ve alçalma fazları için hava aracı performansını doğru değerlendirmesi başarıyı artırmaktadır. Bu amaçla uçuş verilerine erişilebilen Aerosonde İHA seçilmiştir.

Aerosonde, uzun menzilli hava durumu verilerinin alınması için tasarlanmış küçük otonom bir İHA'dır. 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 McGeer'in (1992) araştırma makalesinde önerilmiştir [60]. Gelişmiş bir komuta ve kontrol sistemi ile bütünlük Aerosonde'nin en büyük esnekliği, hemen hemen her yerden konuşlandırılabilmesi ve kontrol edilebilmesidir. Literatürde çok sayıda çalışmada incelenen bir hava aracı olarak öne çıkmaktadır [61-64]. Aerosonde İHA'nın algoritmalarda kullanılacak kısıtları Tablo 7'de listelenmiştir.

Tablo 7. İHA fiziksel kısıtları (UAV physical constraints)

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
Ayırma mesafesi	200 m

MATLAB Simulink, İHA'ları gerçeğe yakın modellemeye imkân sağlayan araçlar içerir [65]. Aerosonde İHA sistemi, MATLAB'da

Tablo 6. GA parametreleri (GA parameters)

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

Aerosim Havacılık Simülasyon Blok Seti kullanılarak bir dizi test uçuşunun simüle edilmesiyle modellenmiştir [66].

Aerosonde İHA'ya ait MATLAB Simulink modeli kullanılarak hava aracının havada kalabildiği her hız ve her irtifada; tırmanma, alçalma (Şekil 2) ve seyir (Şekil 3) fazları için yakıt verileri elde edilerek yol planlama algoritmasında kullanılmıştır [41]. Hava aracının sağa veya sola dönüşünde oluşan yol uzunluğuna dayalı yakıt farklılığı dikkate alınmıştır ancak manevra kaynaklı yakıt tüketimi farklılığı ihmal edilmiştir.

2.3.1. Kinematik kısıtlar (Kinematic constraints)

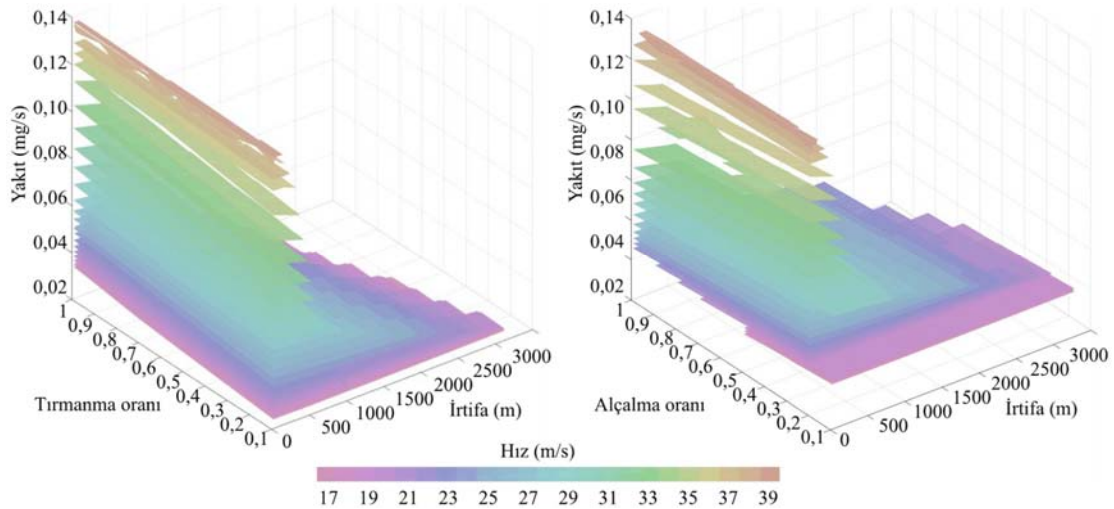
Literatürde yol planlama algoritmalarında uçuş mekaniği açısından çok sayıda kısıt dikkate alınmıştır [67-70]. Önerilen algoritmalarda uçuş süresi, azami irtifa, azami tırmanma hızı, azami yatış açısı gibi geometrik kısıtlar olarak ifade edilebilecek kısıtlar dikkate alınmaktadır.

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 [70]. Algoritma, yalnızca izin verilen dönüş hızları aralığında yörüngeler oluşturarak aracın manevra kabiliyetini aşmamasını sağlamaktadır. Bu parametreler için denklemler aşağıda verilmiştir.

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

$$R_{min} = \frac{v^2}{19 \tan \theta} \quad (5)$$

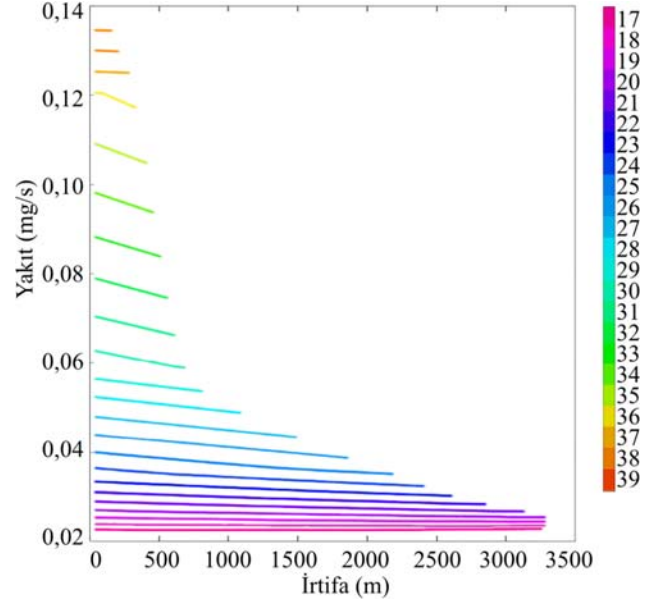
Denklem (Eş. 4) ile asgari dönüş yarıçapı (R_{min}) için yüksek doğrulukta bir hesaplama yapılabilmektedir. Denklemde, W , ağırlık; $\frac{W}{S}$ kanat yüklemesi; $\frac{T}{W}$ itki yüklemesi; C_{D0} Sıfır kaldırma sürüklenim katsayısı; ρ yoğunluktur. Ancak Denklem (Eş. 5) [71] kullanılarak elde edilecek 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 2. Aerosonde İHA'ya ait alçalma ve tırmanma fazı yakıt karakteristiği (Descent and climb phase fuel characteristics of Aerosonde UAV)

2.4. Geliştirilen Yol Planlama Yazılım Paketi (Developed Path Planning Software Package)

Geliştirilen yazılım paketi, yol planlama süreçlerini sistematik ve kullanıcı dostu bir biçimde yönetebilmek amacıyla dizayn edilmiştir. Yazılım arayüzü, tüm grafik çıktıları tek bir pencerede sunar. En uygun bulunan rota, uçuş hızı ve irtifa bilgileri ile birlikte bir noktalar dizisi olarak görselleştirilir ve hesaplama süreci zamanı ekran üzerinde gösterilir.



Şekil 3. Aerosonde İHA'ya ait seyir fazı yakıt karakteristiği (Cruise phase fuel characteristics of Aerosonde UAV)

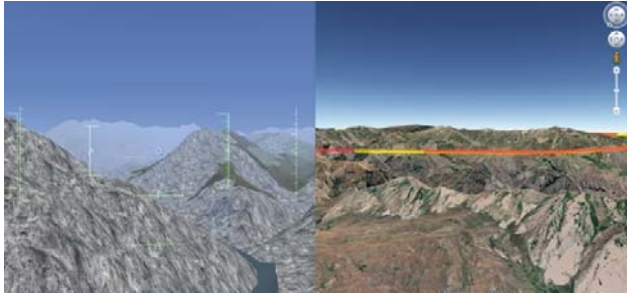
Web tabanlı olarak geliştirilen uygulama ile kullanıcıların çeşitli algoritmik parametreleri değiştirerek yeni haritalar oluşturması, rotaları ve yöntemleri seçmesi ve sonuçları grafikler aracılığıyla incelemesi mümkündür. Her bir planlama işlemi sonrasında, grafiksel analizler yapılabilmekte ve oluşturulan rotalar simülasyon ortamında test edilebilmektedir.

Algoritmanın işleç sayısı, yinleme sayısı, kromozom parçalarının sayısı, mahalledeki komşu sayısı, mutasyon oranı, bitirme kriterleri

(örneğin, belirli sayıda yineleme yapılması ya da uygun bir sonuç elde edilene kadar devam etmesi) ve başlangıç popülasyon büyüklüğü gibi parametreler kullanıcı tarafından tanımlanabilir. Ayrıca, hesaplama tamamlandığında alternatif tüm yolları gösterilmesini sağlayan bir seçenek işaretlenebilir. Yazılım, harita üzerinde 2D uçuş yollarını gösterme işlevine sahiptir ve görev parametreleri olarak ara noktaları ve engelleri belirleme yeteneği sunar. Algoritmalar, çeşitli haritalar ve senaryolar üzerinde otomatik olarak test edilebilir ve hesaplama tamamlandığında asgari ve ortalama uygunluk değerleri oluşturulan çizelgelerde sunulur.

2.5. Simülasyon Ortamı (Simulation Environment)

İHA simülasyon araçları, gerçeğe yakın durumları modellemek için İHA uçuşlarını sanal ortamda taklit etmektedirler. İHA simülasyonu için kullanılan çok sayıda araç bulunmaktadır. Microsoft AirSim [72], FlightGear [73], JMavSim [74], Gazebo [75], X-Plane [76] ve UE4Sim [77] dâhil olmak üzere çok sayıda simülasyon literatürde yer almıştır. 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 simülatördür [73, 78]. Birçok akademik çalışmada kullanılmıştır [79-82].



Şekil 4. Planlanan yolun (kırmızı) ve İHA hareket izinin (sarı) görüntülediği GoogleEarth ve FlightGear eşzamanlı ekran görüntüsü
(Simultaneous screenshot of FlightGear and GoogleEarth showing planned path (red) and UAV motion track (yellow))

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 [83]. Bu çalışmada da hazırlanan yol ve manevralar koordinat tabanlı yörüngeler olarak iki uçuş izi şeklinde eşzamanlı olarak GoogleEarth üzerinde görüntülenmektedir. Şekil 4'te FlightGear, planlanan yol ve İHA hareket izinin görüntülediği GoogleEarth ekran görüntüsüne örnek verilmiştir.

Yollara ait noktalar "waypoint" olarak "routemanager"e telnet üzerinden yüklenir ve otomatik pilot bu "waypoint"ler üzerinden çalışmaktadır. Otomatik pilotun kullandığı hız bilgileri telnet üzerinden kontrol edilmiştir.

3. Deneysel Sonuçlar ve Tartışmalar (Experimental Results and Discussions)

Bu bölümde, önerilen scGA, cGA ve varyantlarının etkinliği standart bir GA ile birlikte karşılaştırılmıştır. Bu karşılaştırma, algoritmaların İHA rotalama probleminde uygulanabilirliğini ve etkinliğini

anlamamıza önemli bir katkı sunmaktadır. Bu bağlamda, GA varyantlarının İHA rotalama problemindeki performanslarına dair detaylı bir anlayış elde edilerek bu alanda potansiyel avantaj ve dezavantajlar ele alınmaktadır. Farklı başlangıç popülasyonları ve mahalle yapıları kullanılarak algoritmaların etkinliği kapsamlı bir şekilde incelenmiştir. Parametre çeşitliliği, algoritmaların performansını çeşitli senaryolarda değerlendirmemize olanak tanımıştır. İteratif süreç boyunca performansın nasıl değiştiği, popülasyon çeşitliliği ve uygunluk değerlerindeki değişimlerle detaylı bir şekilde analiz edilmiştir.

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. Bu çalışmayı netleştirebilmek için mümkün olan en iyi komşuluğun uzunluğu belirlenmeye çalışılmıştır. 2, 3, 5, 6, 7, 8 ve 9 bireylik mahalle uzunluklarında testler gerçekleştirilmiştir.

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. 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 13200 deneye tekabül eden 66 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 de hesaplama süresi değeri kullanılmaktadır. Bu nedenle, geçerli sonuç üretemeyen bir algoritmanın hız açısından daha başarılı görünme algısına karşın analizler, geçerli sonuç üretilme perspektifinden değerlendirilmelidir. Test sonuçları değerlendirilirken, PHP'nin betik bir dil olduğu dikkate alınmalıdır. Bununla birlikte testlerde aynı sistem ve yazılım dili kullanılmış olduğundan algoritmalar arasında adil bir karşılaştırma yapılabilmektedir.

Algoritmaların parametre farklılıklarının performans üzerindeki etkileri; yol ekonomisi, planlama süresi ve algoritma performansı açısından değerlendirilmektedir. Bu nedenle, bu çalışma 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" eki ise çoklu işlem kullanıldığını göstermektedir.

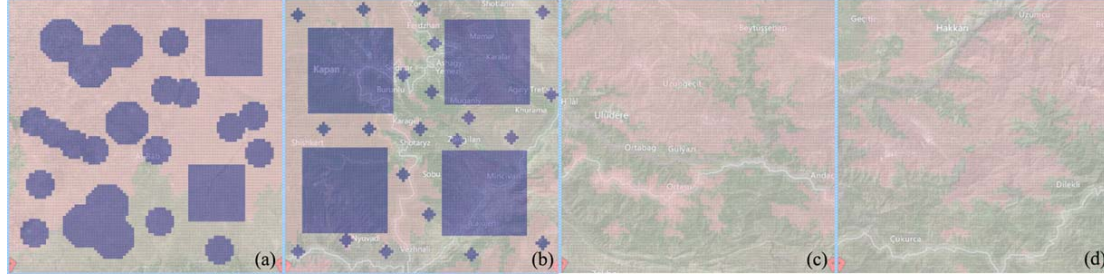
Testler tek süreçli ve çok süreçli olarak iki şekilde uygulanmıştır. Testlerde dört farklı harita kullanılmıştır. Haritalar, 47,5 km² genişliğindedir. Şekil 5'te gösterilen haritalardan; birinci harita, daha yumuşak yeryüzü şekillerine sahiptir ancak çeşitli engeller tanımlanmıştır. İkinci harita, engeller içeren engebeli ve yükseltili bir arazi yapısındadır. Üçüncü ve dördüncü haritalar, engel içermeyen farklı yeryüzü şekillerinde yükseltili arazi yapısına sahiptir (Tablo 8).

Literatürde çözüm çeşitliliğinin karşılaştırılması için genel bir kabul yoktur [84]. Önerilen yaklaşımda çeşitlilik oranı (C_p) için özgün bir hesaplama geliştirilmiştir. Tüm popülasyon için C_p aşağıdaki gibi (Eş. 6) hesaplanmaktadır.

$$C_p = \begin{cases} f(c) & \text{geçerli yol varsa} \\ 0 & \text{geçerli yok yoksa} \end{cases} \quad (6)$$

Tablo 8. Haritalar (Maps)

No	Koordinat aralığı	Asgari yükseklik	Azami yükseklik	Engel durum
1	36,18511383-37,26516926; 36,61304176-37,79441901	356	629	Var
2	36,18511383-37,26516926; 36,61304176-37,79441901	272	3221	Var
3	37,19608466-42,75250172; 37,62401259-43,28875772	555	3286	Yok
4	37,19324349-43,51862462; 37,62117142-44,05485687	805	3962	Yok

**Şekil 5.** (a) Harita-1 (b) Harita-2 (c) Harita-3 (d) Harita-4 ((a) Map-1 (b) Map-2 (c) Map-3 (d) Map-4)

t , yol sayısı; n , toplam nokta sayısı; k , benzersiz nokta sayısı; m , toplam uyum değeri olmak üzere $f_{(c)}$, denklem (Eş. 7) ile hesaplanır:

$$f_{(c)} = \frac{t.k.10^9}{n.m^2} \quad (7)$$

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

Hava aracı performansına dayalı yaklaşımda algoritmalar karşılaştırıldığında; Harita-1'de, 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. Harita-2'de, scGAR-L8 algoritması en iyi, cGAR-L5-Ç algoritması ise en kötü sonuçları elde etmiştir. Harita-3'te, scGA-L5-Ç algoritması en iyi, cGAR-L4-Ç algoritması ise en kötü sonuçları elde etmiştir. Harita-4'te, scGA-L5-Ç algoritması en iyi, cGAR-L4-Ç algoritması ise en kötü sonuçları elde etmiştir (Tablo 9).

Tek süreçli testlerde scGA için 6 hücre genişliğindeki L6 mahallesi test edilen diğer mahalle yapılarından daha iyi bir performans göstermiştir. 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. scGA'da genel olarak mahallelerin genişlemesiyle son nüfus ortalama uygunluk değerinin iyileştiği görülmektedir. cGA için L2, cGA-Ç için L9 mahallesi test edilen diğer mahalle yapılarından daha iyi bir performans göstermiştir (Tablo 9).

Geçersiz sonuç bulma yüzdelerine göre algoritmalar karşılaştırıldığında görece daha sade olan Harita-2 senaryosu hariç diğer senaryolarda cGA(R) ve GA algoritmalarının geçersizlik yüzdesi oldukça yüksektir. 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 oranda geçersiz yollar görülmüştür. Diğer tüm mahalle,

harita ve senaryolarda %0 geçersiz sonuç başarıları 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 (Tablo 10).

Son popülasyona göre test sonuçları yukarıda sunulmuştur. Algoritma başarısının bir diğer ölçütü popülasyonun iyileşme hızıdır. Yapılan 13200 testin ortalamaları dikkate alınarak hazırlanan tablolarda, yinelemelere göre ortalama en iyi çözüm uygunluk değeri değişimi ve yinelemelere göre ortalama popülasyon uygunluk değeri değişimi verilmiştir (Şekil 7). 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 bu bakış açısıyla da 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.

Popülasyonun gelişim davranışı her algoritmada farklıdır. Bu gelişim, görece kolay ve zor iki farklı senaryoda karşılaştırılmıştır. Testler; 1, 10, 20, 30 ve 40 yinelemeli olarak ayrı ayrı çalıştırılmış ve her çalıştırma sonucunda üretilen yollar Şekil 6'da 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. Geçerli yolların farklı renklerde, geçersiz yolların gri renkli olarak gösterildiği Şekil 6'da görüleceği üzere scGA'da harita çok daha erken renklenmekte ve renklilik haritanın geneline yayılmaktadır.

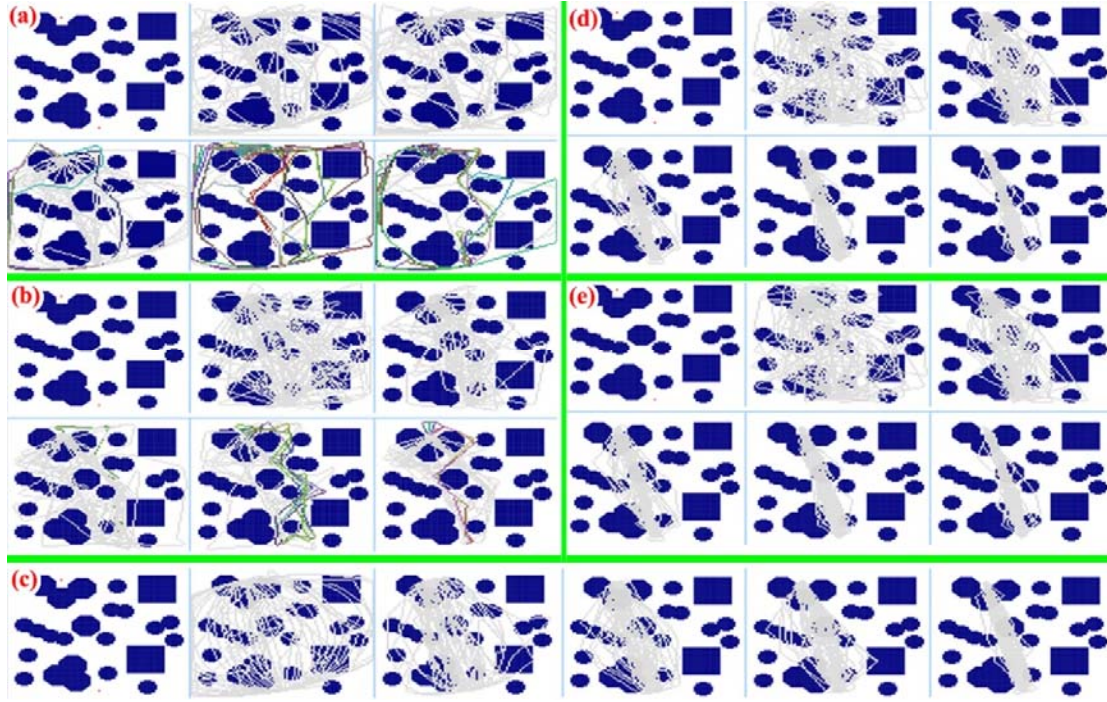
Elde edilen bulgular, scGA'nın genellikle daha yüksek uygunluk değerleri ile etkili bir yol planlama sağladığını göstermektedir. Gerçekleştirilen çeşitli testler, önerilen scGA ve cGA varyantlarının İHA yol planlama probleminde uyumunu anlamak ve gelecekteki çalışmalara yönelik ipuçları elde etmek açısından önemli bir rol oynamıştır. Sonuçların daha detaylı bir şekilde inceleneceği bir sonraki bölümde, algoritmaların performansının ötesinde çeşitli değişkenlerin performansı nasıl etkilediğini anlamak amacıyla daha bütünsel bir analiz gerçekleştirilecektir. İHA yol planlama probleminde özgü genetik algoritmaların evrimini daha kapsamlı bir şekilde anlayarak gelecekteki çalışmalara sağlam bir temel oluşturmak ve daha derinlemesine araştırmalara odaklanmak mümkün olacaktır.

Tablo 9. Hava aracı performansına dayalı amaç fonksiyonu kullanan algoritmaların tüm haritalardaki test sonuçları ortalamaları
(Averages of test results of algorithms using objective function based on aircraft performance for all maps)

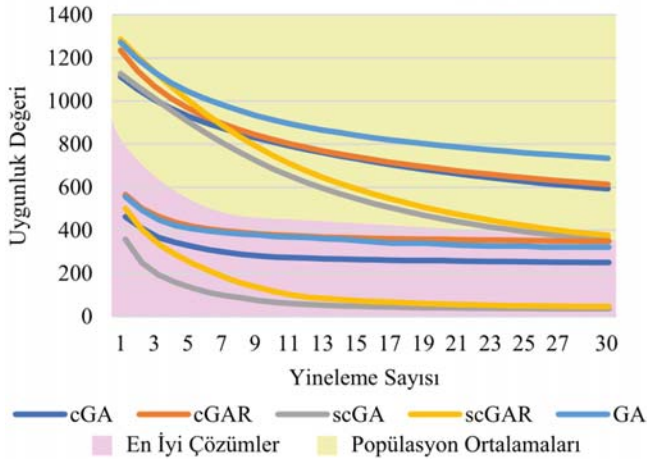
Algoritma	Ort. performansaya dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç yinelenme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Pop. Ort. Uygunluk Değeri	Algoritma	Ort. performansaya dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç bulma süresi (s)	Ort. geçerli sonuç yinelenme sayısı	Ort. en iyi geçerli sonuç bulma süresi (s)	Ort. en kötü geçerli sonuç bulma süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Pop. Ort. Uygunluk Değeri
cGA-L2	223,7	2,6	1,2	1,6	0,1	5,1	25%	6,9	749	GA ^C	201,2	1,1	0,6	1,9	0,2	1,7	28%	6,7	827
cGA-L2 ^C	108,3	1,2	0,4	2,9	0,1	1,6	21%	7,5	755	scGA-L2	86,5	3,6	0,9	2,4	0,2	3,1	11%	7	693
cGA-L3	109,2	2,5	0,8	2,7	0,1	3	21%	6,5	702	scGA-L2 ^C	86,9	1,5	0,5	2,6	0,1	1,4	11%	7,8	701
cGA-L3 ^C	177,1	1,2	0,5	1,8	0,1	2,1	22%	6,1	716	scGA-L3	62,6	3,5	0,8	2,4	0,2	2,2	10%	7,3	630
cGA-L4	200,2	2,5	1,1	1,6	0,1	4,8	25%	6	671	scGA-L3 ^C	86,2	1,5	0,4	2,4	0,1	1,3	10%	7,6	633
cGA-L4 ^C	131,6	1,2	0,5	2,8	0,1	1,6	20%	6,3	691	scGA-L4	109,1	3,6	0,9	1,8	0,2	3,2	11%	7,4	610
cGA-L5	153,7	2,6	1	2,3	0,1	4,7	22%	5,7	635	scGA-L4 ^C	110,1	1,5	0,5	2,4	0,2	1,3	10%	7,7	610
cGA-L5 ^C	155	1,2	0,5	2,5	0,1	1,6	24%	5,8	661	scGA-L5	38,7	3,6	0,7	2,5	0,2	2,3	8%	7,2	585
cGA-L6	178,2	2,6	1	1,3	0,1	4,8	25%	5,7	647	scGA-L5 ^C	63	1,5	0,4	2,5	0,1	1,1	12%	8,1	593
cGA-L6 ^C	130,3	1,3	0,5	2,4	0,1	1,9	23%	5,8	673	scGA-L6	85,4	3,7	0,9	2,1	0,2	3	11%	7,2	571
cGA-L7	131	2,5	0,9	2	0,1	3,8	21%	5,9	618	scGA-L6 ^C	62,4	1,6	0,4	2,6	0,1	1,2	12%	8,4	583
cGA-L7 ^C	176,8	1,3	0,5	1,7	0,1	2,2	23%	5,9	641	scGA-L7	110,2	3,7	0,9	2,3	0,2	2,7	11%	7	570
cGA-L8	176,6	2,6	1	1,2	0,1	4,8	22%	5,8	600	scGA-L7 ^C	61,8	1,6	0,4	2,4	0,1	1,3	11%	8,5	565
cGA-L8 ^C	108,3	1,3	0,4	2,2	0,1	1,4	17%	6	624	scGA-L8	62,7	3,7	0,8	2,5	0,2	2,7	10%	6,6	569
cGA-L9	153,6	2,6	1	1,8	0,1	4,7	23%	6,1	567	scGA-L8 ^C	109,3	1,6	0,5	2,3	0,1	1,4	11%	8,2	560
cGA-L9 ^C	130,4	1,3	0,5	2,5	0,1	2,1	22%	5,9	617	scGA-L9	85,3	3,8	0,9	2,2	0,2	2,8	10%	6,8	617
cGAR-L2	224,3	2,6	1,2	1,4	0,2	4,6	25%	6,9	743	scGA-L9 ^C	86,2	1,5	0,4	2,9	0,2	1,2	10%	8,3	614
cGAR-L2 ^C	200,8	1,3	0,6	3	0,1	1,9	26%	7	749	scGAR-L2	40,1	4,7	1,1	2,9	0,2	3,6	1%	6,1	605
cGAR-L3	201,2	2,5	1,1	1,6	0,2	3,3	27%	5,6	701	scGAR-L2 ^C	39,8	1,9	0,5	2,6	0,2	1,4	1%	6,7	614
cGAR-L3 ^C	177,7	1,3	0,6	2,1	0,1	2,1	26%	5,7	710	scGAR-L3	38,8	4,4	0,9	2,3	0,2	2,7	1%	6,5	505
cGAR-L4	223,9	2,5	1,2	1,7	0,2	4,4	26%	5,3	665	scGAR-L3 ^C	39,2	1,9	0,5	2,8	0,2	1,4	2%	6,8	538
cGAR-L4 ^C	200,4	1,3	0,6	1,9	0,1	1,8	24%	5,3	674	scGAR-L4	39,6	4,5	1	2,8	0,2	3,1	1%	6,8	491
cGAR-L5	200,5	2,6	1,2	2,4	0,2	4,3	27%	5,3	617	scGAR-L4 ^C	39,1	1,9	0,5	2,9	0,2	1,2	2%	7,4	505
cGAR-L5 ^C	201	1,3	0,6	2,4	0,1	2	27%	5,7	653	scGAR-L5	39,1	4,5	1	2,4	0,2	3,1	0%	6,8	458
cGAR-L6	200,5	2,6	1,1	1,5	0,2	4,3	30%	5,6	616	scGAR-L5 ^C	38,9	1,9	0,5	2,9	0,2	1,1	2%	7,2	500
cGAR-L6 ^C	200,4	1,4	0,6	2,4	0,1	1,9	26%	5,4	631	scGAR-L6	38,7	4,6	1	2,5	0,2	2,7	0%	7	451
cGAR-L7	176,7	2,6	1,1	2,4	0,2	3,9	26%	5,5	599	scGAR-L6 ^C	39,1	1,9	0,5	2,6	0,2	1,3	1%	7,5	474
cGAR-L7 ^C	223,1	1,4	0,7	2	0,1	1,8	28%	5,3	621	scGAR-L7	38,7	4,7	1	2,4	0,2	3,2	1%	6,6	457
cGAR-L8	176,9	2,6	1,1	2,1	0,2	3,9	27%	5,5	568	scGAR-L7 ^C	39,6	2	0,5	3	0,2	1,5	1%	6,9	495
cGAR-L8 ^C	223,5	1,4	0,7	1,6	0,1	2	27%	5,5	598	scGAR-L8	38,8	4,7	1	2,4	0,2	2,5	0%	6,7	448
cGAR-L9	177,9	2,6	1,1	2,1	0,2	3,1	24%	6	558	scGAR-L8 ^C	39,2	2	0,5	2,9	0,2	1,3	2%	7,3	484
cGAR-L9 ^C	200,3	1,5	0,7	1,9	0,1	2,1	28%	5,6	603	scGAR-L9	38,9	4,7	1	2,5	0,2	2,8	0%	7,4	437
GA	177,5	2,8	1,3	2,5	0,2	4,4	29%	5,3	753	scGAR-L9 ^C	38,9	2	0,6	3,1	0,2	1,6	2%	7,6	476

Tablo 10. Hava aracı performansına dayalı algoritmaların ortalama uygunluk değerlerine ve geçersiz sonuç yüzdelere göre karşılaştırması (Comparison of average fitness values and invalid result percentages for algorithms based on aircraft performance)

Algoritma	Ortalama uygunluk değeri				Geçersiz sonuç yüzdesi				Algoritma	Ortalama uygunluk değeri				Geçersiz sonuç yüzdesi			
	Harita-1	Harita-2	Harita-3	Harita-4	Harita-1	Harita-2	Harita-3	Harita-4		Harita-1	Harita-2	Harita-3	Harita-4	Harita-1	Harita-2	Harita-3	Harita-4
scGA-L6	42,9	23,9	35,9	50,0	0%	0%	0%	0%	cGA-L2 ^C	47,1	24,3	36,7	325	24%	4%	16%	40%
scGA-L7	42,7	23,8	36,5	50,0	0%	0%	0%	0%	cGA-L8 ^C	231	24,0	37,9	233	28%	4%	14%	32%
scGA-L9	43,0	23,9	36,3	50,2	0%	0%	0%	0%	cGA-L4 ^C	139	24,2	130	234	18%	6%	18%	36%
scGA-L8	43,0	23,9	36,0	50,7	0%	0%	0%	0%	cGA-L9 ^C	139	23,8	126	232	26%	6%	18%	36%
scGA-L7 ^C	43,0	23,9	36,0	50,9	0%	0%	0%	0%	cGA-L7	139	24,0	219	232	22%	2%	20%	38%
scGA-L9 ^C	43,1	23,7	36,0	51,1	0%	0%	0%	0%	cGA-L8	139	24,1	126	324	22%	8%	18%	36%
scGA-L5 ^C	43,2	23,7	35,8	51,1	0%	0%	0%	0%	cGA-L6 ^C	139	24,0	219	231	20%	6%	20%	40%
scGA-L6 ^C	43,1	23,9	36,3	50,9	0%	0%	0%	0%	cGA-L5	46,7	24,0	219	325	18%	12%	20%	38%
scGAR-L6	42,7	23,9	36,9	51,2	0%	0%	0%	0%	cGA-L5 ^C	139	24,1	132	324	26%	8%	18%	42%
scGAR-L8	42,8	23,7	36,5	52,0	0%	0%	0%	0%	cGA-L6	139	23,9	131	325	26%	10%	18%	44%
scGAR-L9	42,5	24,0	37,1	51,4	0%	0%	0%	0%	cGA-L3 ^C	139	24,4	219	326	24%	4%	20%	40%
scGA-L3 ^C	43,4	23,8	36,2	52,3	0%	0%	0%	0%	cGA-L7 ^C	138	24,0	219	326	24%	2%	20%	44%
scGA-L2 ^C	43,4	24,0	37,6	50,8	0%	0%	0%	0%	cGA-L9	139	23,7	219	325	22%	4%	22%	46%
scGAR-L5	42,7	23,9	37,0	52,6	0%	0%	0%	0%	cGAR-L9	140	24,1	37,2	511	38%	4%	18%	52%
scGA-L4	43,1	23,9	36,5	50,2	0%	0%	2%	0%	cGA-L4	232	24,4	219	325	24%	8%	20%	46%
scGA-L8 ^C	43,2	23,9	36,6	50,1	0%	0%	2%	0%	cGAR-L6	43,8	24,0	127	512	30%	14%	20%	58%
scGA-L5	43,0	23,9	36,4	50,5	0%	0%	0%	2%	cGAR-L2	140	24,3	128	512	36%	2%	20%	50%
scGA-L3	43,0	23,9	36,6	50,9	0%	0%	2%	0%	GA ^C	234	24,0	127	420	38%	6%	20%	48%
scGAR-L7	42,7	23,8	36,9	51,3	0%	0%	0%	2%	GA	231	23,7	127	418	38%	8%	20%	50%
scGA-L4 ^C	43,2	23,9	36,7	51,5	0%	0%	2%	0%	cGAR-L4	139	24,3	127	512	38%	8%	20%	50%
scGA-L2	43,7	24,1	37,4	52,2	0%	0%	2%	0%	cGA-L2	233	24,4	219	418	26%	12%	20%	42%
scGAR-L3	42,7	23,8	36,1	52,2	0%	0%	0%	4%	cGAR-L3 ^C	235	24,2	219	329	42%	8%	20%	50%
scGAR-L4	42,8	23,9	36,7	54,8	0%	0%	0%	4%	cGAR-L7	232	23,9	126	418	42%	10%	20%	50%
scGAR-L2 ^C	43,8	23,9	38,2	53,6	2%	0%	0%	2%	cGAR-L8 ^C	137	25,2	219	418	44%	6%	22%	52%
scGAR-L2	43,6	23,9	38,8	54,0	0%	0%	2%	2%	cGAR-L6 ^C	141	24,0	219	419	48%	10%	22%	50%
scGAR-L9 ^C	43,1	23,7	37,0	51,9	2%	0%	0%	6%	cGAR-L4 ^C	233	24,1	220	419	40%	8%	20%	50%
scGAR-L5 ^C	43,4	23,7	36,5	52,1	2%	0%	4%	2%	cGAR-L8	331	23,7	127	418	44%	14%	18%	54%
scGAR-L4 ^C	43,1	23,7	35,9	53,7	2%	0%	0%	6%	cGAR-L7 ^C	139	23,9	219	511	48%	10%	24%	54%
scGAR-L6 ^C	43,2	23,7	36,9	52,5	8%	0%	0%	2%	cGAR-L2 ^C	234	24,7	220	512	38%	4%	20%	56%
scGAR-L8 ^C	43,4	23,7	37,8	52,2	4%	0%	2%	6%	cGAR-L5 ^C	329	119,3	127	419	38%	10%	18%	54%
scGAR-L7 ^C	42,9	23,8	37,5	54,2	8%	0%	0%	4%	cGAR-L3	232	24,1	220	512	40%	10%	22%	52%
scGAR-L3 ^C	43,5	23,7	36,5	53,3	8%	0%	2%	4%	cGAR-L9 ^C	327	24,1	219	419	42%	10%	22%	52%
cGA-L3	49,2	24,1	37,9	326	22%	2%	16%	42%	cGAR-L5	326	24,1	219	512	42%	14%	20%	54%



Şekil 6. (a) scGA (b) scGAR (c) cGA (d) cGAR (e) GA (soldan sağa) 1, 10, 20, 30 ve 40 yinelemeli çalıştırmalarda tarafından üretilen yolların gösterimi
(aa) scGA (b) scGAR (c) cGA (d) cGAR (e) GA (left to right) Representation of paths generated by 1, 10, 20, 30 ve 40 iterative runs



Şekil 7. Yinelemelere göre ortalama en iyi çözüm ve ortalama popülasyon uygunluk değerinin değişimi
(Variation of average best individuals fitness value and average population fitness value for each iteration)

4. Sonuçlar (Conclusions)

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 sağlanması beklenmektedir. Bu doğrultuda daha az yakıt tüketimi ve karbon emisyonunun azaltılması ile çevresel kazanım sağlanmaktadır. Hesaplama maliyetinde verimlilik gerçek zamanlı çalışmaya imkân tanımaktadır.

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 yarıya 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 bütünlük enerji kaynağının önemi kritiktir. Test sonuçları, yol planlama algoritmalarının güvenilirliğini ve sağlamlığını ölçerken gerçek uygulamalara yaygınlaştırılma potansiyelini belirlemektedir.

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 hüresel genetik bir algoritmadır.

Önerilen Hüresel 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. Rastgele başlangıç popülasyonu kullanan cGAR algoritmasının başarısı cGA'ya ve diğer algoritmalara kıyasla oldukça düşüktür. Bu durum, sabit başlangıç popülasyonunun bir başarısı ve standart cGA'nın ancak önerilen sabit başlangıç popülasyonu ile istenen başarıyı yakalayabileceğinin bir göstergesi olarak

Tablo 11. Tüm harita ve mahalle yapılarının algoritmalara göre ortalamaları ve algoritmaların geçerli ve geçersiz sonuç üretmedeki hesaplama süreleri

(Average results of algorithms for all maps and neighborhood structures and Computation times for algorithms to produce valid and invalid results)

Algoritma	Ort. perf. dayalı uygunluk değeri	Ort. hesaplama süresi (s)	Ort. geçerli sonuç süresi (s)	Ort. geçerli sonuç yineleme sayısı	Ort. en iyi geçerli sonuç süresi i(s)	Ort. en kötü geçerli sonuç süresi (s)	Geçersiz sonuç yüzdesi	Çeşitlilik oranı	Son Popülasyon Ort. Uyumu Değeri	Ort. geçerli yol hesap süresi (s)	Ort. geçersiz yol hesap süresi (s)	Geçerli /geçersiz yol üretme arasındaki süre farkı
scGA	38,6	2,9	0,5	2,0	0,1	1,5	0,2	8,1	468	2,41	3,21	%133,5
scGA(R)	39,1	3,3	0,9	3,1	0,3	2,4	1,4	7,0	502	3,84	3,53	%92,0
cGA	155,5	1,9	0,8	2,1	0,1	3,1	22,0	6,1	648	3,23	1,93	%59,6
cGA(R)	218,2	2,0	1,0	2,0	0,2	3,0	30,6	5,8	669	2,93	1,90	%64,9
GA	200,6	1,9	0,9	2,1	0,2	3,0	28,5	5,6	795	3,01	1,99	%66,2

değerlendirilebilir. scGAR'de ise durum oldukça 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. 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 nesiller boyu popülasyon çeşitliliğini devam ettirdiği fark edilebilir. Görüldüğü gibi, cGA ve scGA(R), keşif özelliklerini korur ve algoritma çalışmasını tamamladığında GA'ya göre çok daha fazla Pareto cephe çözümü üretir. 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 bulunduğu 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 sağladığını açıkça göstermektedir.

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.

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. Önceki bölümde 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 çözüm kalitesinde test sonuçlarına da yansıyan bir farklılığa sebep olmaktadır.

Testlerin, sadece 4 gerçek ve 4 sanal süreç desteği olan mütevazı bir işlemcide yapılmış olması çoklu işlem çözüm süresinin sınırlayıcı faktördür. (Ryzen 2400G CPU, bu çalışma hazırlandığı dönemde piyasadaki çoklu işlemde en iyi işlemcilerle kıyasla masaüstünde %425, iş istasyonunda %1525 daha düşük performansa sahiptir [85, 86].) 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. Veritabanı kullanımıyla aynı ağ üzerinde çalışan çok sayıda bilgisayarın paralel işlem gücü ile elde edilebilecek yüksek problem çözüme yeteneği sonraki çalışmalar için umut vadetse de bu çalışmada bir dezavantaj olarak yer almıştır.

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 veri alışverişinin 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.

scGA(R), neredeyse tüm senaryolar için hem uygunluk değerinde hem de hesaplama süresinde daha iyi performans göstermiştir. Hesaplama süresinde GA'nın önde olduğu görülmektedir. Ancak geçersiz sonuçların hesaplama hızı GA'da %66,2 daha kısadır (Tablo 11). GA'nın ürettiği yolların da %28,50'si geçersizdir. Geçerli ve geçersiz yolların üretilmesindeki hız farkının en önemli nedeni hatalı yakınsamadır. Genellikle, geçersiz en iyi yol ki bu aynı zamanda en kısa yoldur, iki nokta arasındaki düz çizgidir (Şekil 6). Bu nedenle geçersiz yoğunluklu popülasyon genelinde yollar kısalmakta ve hesaplama süresi büyük oranda düşmektedir. scGA'da ise durum bunun tam tersidir. Popülasyon genelindeki çeşitlilik çalışma süresinin sonuna kadar korunur ve geçerli bir aday çözüme istenmeyen bir 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. scGA'nın hesapladığı geçersiz yol sayısının çok az olması ve bu senaryoların genel olarak zor ve çözüm sürelerinin yüksek olması nedeniyle Tablo 11'de scGA için gösterilen %133,5 oranının gerçekçi olmadığı söylenebilir. Bunun yanında scGAR'de değerlendirme yapmak açısından yeterli veri bulunduğundan scGA'nın da geçersiz yollarda hesaplama süresi artışının %92 oranına yakın bir değerde olduğu değerlendirilebilir.

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 güzergâhlar içermesi bir avantajdır.

Farklı senaryoları içeren çeşitli konfigürasyonlarda testler gerçekleştirilmiştir. İHA'ya özgü bir model kullanılarak aracın dinamiklerine uygun optimale yakın yollar üretilmiştir. 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. Geçersiz yol üretme başarısı, geçerli bir çözüme en hızlı ulaşabilme ve hesaplama tamamlandığında üretilen yollardaki yüksek çeşitlilik; karşılaştırılan algoritmalar arasında scGA'yı zirveye taşımıştır.

Bu çalışmanın 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; hız modeli, yakıt modeli, tırmanma oranı ve alçalma oranı modelleri, tırmanış programına göre model, eğrisel yörünge için yakıt modeli oluşturulması tercih edilir. 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 sınırlanmıştır. Çoklu İHA tarafından yürütülen görevler açısından optimize edilmesi önemli bir geliştirme alanıdır.
- Test edilen tüm algoritmalar CPU üzerinde uygulanmıştır. Yazılım, GPU'lar üzerinde çalışabilecek şekilde düzenlenebilir ve daha yüksek bir performansa ulaşılabilir.
- Algoritmanın bir betik dilden ziyade C/C++ gibi daha düşük seviyeli bir dil ile yazılması hesaplama süresini önemli ölçüde azaltabilir.

Kaynaklar (References)

1. The Regulatory Horizons Council, The Regulation of Drones: An exploratory study, Department for Business, Energy & Industrial Strategy (BEIS), UK, 2021.
2. Atr-Aircraft. Fuel Saving. https://www.atraircraft.com/userfiles/files/Fuel_Saving_2011.pdf Yayın tarihi Ocak 30, 2011. Erişim tarihi Ağustos 9, 2016.
3. Mayer, R., A flight trajectory model for a pc-based airspace analysis tool, AIAA Modeling and Simulation Technologies Conference and Exhibit, 1-10, 2002.
4. Eurocontrol. Base of aircraft data. <https://www.eurocontrol.int/model/bada> Yayın tarihi 16.01.2009. Erişim tarihi 20.03.2022.
5. Li, D.C., Cheng, N., Cheng, P., Song, J.Y., 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, 2012.
6. Labonte, G., Simple formulas for the fuel of climbing propeller driven airplanes, Advances in aircraft and spacecraft science, 2 (4), 367-389, 2015.
7. Roberge, V., Tarbouchi, M., Labonté, G., Fast genetic algorithm path planner for fixed-wing military UAV using GPU, IEEE Transactions on Aerospace and Electronic Systems, 54(5), 2105-2117, 2018.
8. Langelaan, J.W., Gust energy extraction for mini and micro uninhabited aerial vehicles, Journal of guidance, control, and dynamics, 32 (2), 464-473, 2009.
9. Langelaan, J., Long distance/duration trajectory optimization for small UAVs, AIAA Guidance, Navigation and Control Conference and Exhibit, 1-14, 2007.
10. Williams, A., Yakimenko, O., 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, 2018.
11. Fujii, K., Higuchi, K., Rekimoto, J., 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, 2013.
12. Qi, Y.C., Zhao, Y.J., Energy-efficient trajectories of unmanned aerial vehicles flying through thermals, Journal of Aerospace Engineering, 18 (2), 84-92, 2005.
13. Kniffin, C.A., Dogan, A., Blake, W.B., Formation flight for fuel saving in coronet mission-part a: Sweet spot determination, AIAA Atmospheric Flight Mechanics Conference, 1-14, 2016.
14. Russell, S., Norvig, P., Intelligence artificielle: Avec plus de 500 exercices, Pearson Education, Paris, 2010.
15. Li, S., Sun, X., Xu, Y., Particle swarm optimization for route planning of unmanned aerial vehicles, Information Acquisition, 2006 IEEE International Conference on, IEEE, 1213-1218, 2006.
16. Gao, D., Gong, G., Wang, J., Han, L., Multi-resolution path planning for Miniature Air Vehicles with wind effect, Industrial Informatics (INDIN), 2012 10th IEEE International Conference on, IEEE, 167-171, 2012.
17. Yılmaz N., Gencer C., Integration of sensor vision capabilities on UAV flight route optimization: a linear model and a heuristic algorithm proposal, Journal of the Faculty of Engineering and Architecture of Gazi University, 34 (4), 1917-1928, 2019.
18. Holland, J.H., Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT press, Massachusetts, 1992.
19. Fu, X., Gao, X., Genetic algorithm with adaptive immigrants for dynamic flight path planning, 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems, IEEE, 630-634, 2010.
20. Durmaz E.D., Şahin R., NSGA-II and goal programming approach for the multi-objective single row facility layout problem, Journal of the Faculty of Engineering and Architecture of Gazi University, 32 (3), 941-955, 2017.
21. Okay, F.Y., Özdemir, S., Improving coverage in wireless sensor networks using multiobjective evolutionary algorithms, Journal of the Faculty of Engineering and Architecture of Gazi University, 30 (2), 143-153, 2015.
22. Xiao, J., Michalewicz, Z., Zhang, L., Trojanowski, K., Adaptive evolutionary planner/navigator for mobile robots, IEEE transactions on evolutionary computation, 1 (1), 18-28, 1997.
23. Yıldırım, M., Akay, R., Fast path planning in multi-obstacle environments for mobile robots, Journal of the Faculty of Engineering and Architecture of Gazi University, 36 (3), 1551-1564, 2021.
24. Nikolos, I.K., Valavanis, K. P., Tsourveloudis, N. C., Kostaras, A. N., Evolutionary algorithm based offline/online path planner for UAV navigation, IEEE Trans Syst Man Cybern B Cybern, 33 (6), 898-912, 2003.
25. Roberge, V., Tarbouchi, M., Labonte, G., Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, IEEE Transactions on Industrial Informatics, 9 (1), 132-141, 2013.
26. Fu, Y.-G., Zhou, C.-P., Hu, H.-P., Research on Differential Evolution Algorithm for Path Planning for Unmanned Aerial Vehicle in Ocean Environment, Acta Armamentarii, 3, 1-9, 2012.
27. Fu, S.-Y., Han, L.-W., Tian, Y., Yang, G.-S., 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, 2012.
28. Zheng, R., Feng, Z.-M., Lu, M.-Q., Application of Particle Genetic Algorithm to Path Planning of Unmanned Aerial Vehicle, Computer Simulation, 6, 88-92, 2011.
29. Alba, E., Dorronsoro, B., Cellular genetic algorithms, 42, Springer Science & Business Media, Heidelberg, 2009.
30. Whitley, L.D., Cellular genetic algorithms, Proceedings of the 5th International Conference on Genetic Algorithms, 1-12, 1993.
31. Manderick, B., Spiessens, P., Fine-grained parallel genetic algorithms, Proceedings of the third international conference on Genetic algorithms, Morgan Kaufmann Publishers Inc., 428-433, 1989.
32. Robertson, G.G., Parallel implementation of genetic algorithms in a classifier system, Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms, Hillsdale, NJ: L. Erlbaum Associates, Massachusetts

- Institute of Technology, Cambridge, 140-147, 1987.
33. Wright, S., Isolation by distance, *Genetics*, 28 (2), 114-138, 1943.
 34. Alba, E., Dorronsoro, B., Solving the vehicle routing problem by using cellular genetic algorithms, *Evolutionary Computation in Combinatorial Optimization*, Springer, Heidelberg, 2004.
 35. Alba, E., Alfonso, H., Dorronsoro, B., Advanced models of cellular genetic algorithms evaluated on SAT, *Proceedings of the 2005 conference on Genetic and evolutionary computation*, ACM, 1123-1130, 2005.
 36. Folino, G., Pizzuti, C., Spezzano, G., 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, 1998.
 37. Murata, T., Gen, M., Cellular genetic algorithm for multi-objective optimization, *Proc. of the 4th Asian Fuzzy System Symposium*, 538-542, 2002.
 38. Mantere, T., Image ordering by cellular genetic algorithms with TSP and ICA, *Evolutionary Computation*, 2009. *CEC'09. IEEE Congress on*, IEEE, 822-829, 2009.
 39. Gezer, A., Turan, Ö., Baklacioğlu, T., UAV path planning using segmented cellular evolutionary algorithm, *International Journal of Sustainable Aviation*, 2 (3), 222-234, 2016.
 40. Gezer, A., Turan, Ö., Baklacioğlu, T., Hücresel Genetik Algoritma ile İnsansız Hava Araçlarında Yol Planlama, *Proceeding 3rd International Conference on Applied Engineering and Natural Sciences*, Konya, Türkiye, 1346-1350, 2022.
 41. Gezer, A., Turan, Ö., Baklacioğlu, T., İnsansız Hava Aracı Performansına Dayalı Yol Planlama, *Proceeding 3rd International Conference on Applied Engineering and Natural Sciences*, Konya, Türkiye, 2055-2061, 2022.
 42. Boulos, M.N.K., 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, 2005.
 43. Schmidt, M., Weiser, P., *Web mapping services: development and trends*, Online maps with APIs and WebServices, Springer, 2012.
 44. Özalp, N., Sahingoz, O.K., 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, 2013.
 45. Spiessens, P., Manderick, B., A massively parallel genetic algorithm: Implementation and first analysis, *Proceedings of the Fourth International Conference on Genetic Algorithms*, 279-286, 1991.
 46. Alba, E., Troya, J.M., Improving flexibility and efficiency by adding parallelism to genetic algorithms, *Statistics and Computing*, 12(2), 91-114, 2002.
 47. Petty, C., *Diffusion (cellular) models*, *Handbook of evolutionary computation*, Oxford Univ. Press, Oxford, 1997.
 48. Sarma, J., De Jong, K., 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, 1996.
 49. Lee, K.-Y., Han, S.-N., Roh, M.-I., An improved genetic algorithm for facility layout problems having inner structure walls and passages, *Computers & Operations Research*, 30 (1), 117-138, 2003.
 50. Nogueras, R., Cotta, C., Using distribution-based operators in genetic algorithms, *Proceedings of the 6th Metaheuristic International Conference*, págs, 707-712.
 51. Chiarella, C., Tò, T.-D., The multifactor nature of the volatility of futures markets, *Computational Economics*, 27 (2), 163-183, 2006.
 52. Shim, Y., Husbands, P., Incremental embodied chaotic exploration of self-organized motor behaviors with proprioceptor adaptation, *Frontiers in Robotics and AI*, 2, 1-7, 2015.
 53. Homaiyar, A., Qi, C.X., Lai, S.H., Constrained optimization via genetic algorithms, *Simulation*, 62 (4), 242-253, 1994.
 54. Xia, S., Zhang, X., Constrained path planning for unmanned aerial vehicle in 3D terrain using modified multi-objective particle swarm optimization, *Actuators*, Multidisciplinary Digital Publishing Institute, 255-265, 2021.
 55. Mcmanus, I.A., Walker, R.A., Multidisciplinary approach to intelligent unmanned-airborne-vehicles mission planning, *Journal of Aircraft*, 43 (2), 318-335, 2006.
 56. Li, S.D., Liu, S., Ai, Q., Li, J., Fast 3D Path Pre-Planning Method for UAVs, *Applied Mechanics and Materials*, 278, 1676-1681, 2013.
 57. Zheng, C., Ding, M., Zhou, C., Real-Time Route Planning for Unmanned Air Vehicle with an Evolutionary Algorithm, *International Journal of Pattern Recognition and Artificial Intelligence*, 17 (01), 63-81, 2011.
 58. Lin, L., Goodrich, M.A., UAV intelligent path planning for wilderness search and rescue, *Intelligent Robots and Systems*, 2009. *IROS 2009. IEEE/RSJ International Conference on*, 709-714, 2009.
 59. Sun, T.Y., Huo, C.L., Tsai, S.J., Liu, C.C., Optimal UAV flight path planning using skeletonization and particle swarm optimizer, 2008 *Ieee Congress on Evolutionary Computation*, Vols 1-8, 1183-1188, 2008.
 60. Holland, G.J., McGeer, T., Youngren, H., Autonomous aerodeses for economical atmospheric soundings anywhere on the globe, *Bulletin of the American Meteorological Society*, 73 (12), 1987-1998, 1992.
 61. Sliwinski, J., Gardi, A., Marino, M., Sabatini, R., Hybrid-electric propulsion integration in unmanned aircraft, *Energy*, 140, 1407-1416, 2017.
 62. Al-Hababeh, O.M., Configuration of hybrid fuel-electric airplane model based on full flight path performance, *Jurnal Teknologi*, 84 (3), 59-67, 2019.
 63. Lobit Cerrato, A., *Simulación de vuelo de múltiples UAVs de ala fija*, Universidad de Sevilla, Sevilla, 2016.
 64. Bayraktar, O., Özdemir, F., Çetin, Ö., Yılmaz, G., İnsansız Hava Araçları İçin Otonom İniş Sistemi Simülâtörü Tasarımı, *Bilişim Teknolojileri Dergisi*, 5 (2), 1-8, 2012.
 65. Kıvanç, Ö.C., Mungan, T.E., Atila, B., Tosun, G., An integrated approach to development of unmanned ground vehicle: design, analysis, implementation and suggestions, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34 (4), 1957-1973, 2019.
 66. Dynamics, U. *Aerosim Toolbox*. <http://www.u-dynamics.com/aerosim/> Yayın tarihi Şubat 1, 2014. Erişim tarihi Şubat 1, 2014.
 67. Yingxun, W., Zongji, C., Genetic algorithms (GA) based flight path planning with constraints, *J. Beijing Univ. Aeronaut. Astronaut.(China)*, 25 (3), 355-358, 1999.
 68. Schopferer, S., Pfeifer, T., Performance-aware flight path planning for unmanned aircraft in uniform wind fields, *Unmanned Aircraft Systems (ICUAS)*, 2015 *International Conference on*, IEEE, 1138-1147, 2015.
 69. Yakovlev, K., Makarov, D., Baskin, E., Automatic path planning for an unmanned drone with constrained flight dynamics, *Scientific and Technical Information Processing*, 42 (5), 347-358, 2015.
 70. Myers, D., Batta, R., Karwan, M., 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, 2016.
 71. Faa, *Pilot's handbook of aeronautical knowledge*, Skyhorse Publishing Inc., New York, 2009.
 72. Madaan, R., Gyde, N., Vemprala, S., Brown, M., Nagami, K., Taubner, T., Cristofalo, E., Scaramuzza, D., Schwager, M., Kapoor, A., *Airsim drone racing lab*, *NeurIPS 2019 Competition and Demonstration Track*, PMLR, 177-191, 2020.
 73. Perry, A.R., The flightgear flight simulator, *Proceedings of the USENIX Annual Technical Conference*, 1-53, 2004.
 74. Spencer, J., Lee, J., Paredes, J.A., Goel, A., Bernstein, D., An Adaptive PID Autotuner for Multicopters with Experimental Results, *ArXiv*, 2109 (12797), 1-7, 2021.
 75. Wright, T., West, A., Licata, M., Hawes, N., Lennox, B., *Simulating Ionising Radiation in Gazebo for Robotic Nuclear Inspection Challenges*, *Robotics*, 10 (3), 86-102, 2021.
 76. Henriques, L.C.F., De Souza, C.P., Development of an Embedded Longitudinal Flight Control Based on X-Plane Flight Simulator, *IEEE Latin America Transactions*, 19 (10), 1684-1691, 2021.
 77. Mueller, M., Casser, V., Lahoud, J., Smith, N., Ghanem, B., *Ue4sim: A photo-realistic simulator for computer vision applications*, *arXiv*, 1708 (05869), 1-18, 2017.
 78. Elsadig, M.A., Elbakri, M.A., Design of autopilot platform using HILS approach, *Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, 2017 *International Conference on*, IEEE, 1-6, 2017.
 79. Huang, H., Xu, Y.-P., Deng, Z.-W., Real-time visual flight simulation system based on flightgear simulator, *Journal of System simulation*, 19 (19), 780-785, 2007.
 80. Sorton, E., Hammaker, S., Simulated flight testing of an autonomous unmanned aerial vehicle using flightgear, *Infotech@ Aerospace*, 1-11, 2005.
 81. Joscelyne, R., Young, T., 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, 2011.
82. Romaniuk, S., Gosiewski, Z., Ambroziak, L., A Ground control station for the UAV flight simulator, *acta mechanica et automatica*, 10 (1), 28-32, 2016.
83. Zhao, X., Wang, Y., Zhang, W., Zhang, X., Personal Desktop-Level Jet Fighter Simulator for Training or Entertainment, *International Conference on Intelligent Robotics and Applications*, Springer, 417-427, 2017.
84. Garcia-Najera, A., Bullinaria, J.A., A multi-objective density restricted genetic algorithm for the vehicle routing problem with time windows, 2008 UK Workshop on Computational Intelligence, 47-52, 2008.
85. Userbenchmark. UserBenchmark: AMD Ryzen 5 2400G vs Intel Core i9-12900KS. <https://cpu.userbenchmark.com/Compare/Intel-Core-i9-12900KS-vs-AMD-Ryzen-5-2400G/m1821519vsm433194> Yayın tarihi Mayıs 4, 2022. Erişim tarihi Mayıs 4, 2022.
86. Cpu-Monkey. AMD Ryzen Threadripper PRO 3995WX vs AMD Ryzen 5 2400G. https://www.cpu-monkey.com/en/compare_cpu-amd_ryzen_threadripper_pro_3995wx-1899-vs-amd_ryzen_5. Erişim tarihi Mayıs 4, 2022.

