Askeri Hava Taşıma Probleminde Risk Yönelimli Eşzamanlı Rotalama–Çizelgeleme ve Çözüm Önerileri

Özgür Aksoy

DOKTORA TEZİ

Endüstri Mühendisliği Anabilim Dalı

Ekim 2018

Risk-Oriented Simultaneous Routing–Scheduling for Military Air Transportation Problem and Solution Proposals

Özgür Aksoy

DOCTORAL DISSERTATION

Department of Industrial Engineering

October 2018

Askeri Hava Taşıma Probleminde
Risk Yönelimli Eşzamanlı Rotalama–Çizelgeleme ve Çözüm Önerileri

Özgür Aksoy

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Endüstri Mühendisliği Anabilim Dalı
Endüstri Mühendisliği Bilim Dalında
DOKTORA TEZİ
Olarak Hazırlanmıştır

Danışman: Prof. Dr. Muzaffer Kapanoğlu

Ekim 2018

ONAY

Endüstri Mühendisliği Anabilim Dalı Doktora öğrencisi Özgür Aksoy'un DOKTORA tezi olarak hazırladığı "Askeri Hava Taşıma Probleminde Risk Yönelimli Eşzamanlı Rotalama- Çizelgeleme ve Çözüm Önerileri" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek oybirliği ile kabul edilmiştir.

Danışman : Prof.Dr. Muzaffer Kapanoğlu

İkinci Danışman : –

Doktora Tez Savunma Jürisi:

Üye: Prof.Dr. Muzaffer Kapanoğlu

Üye: Prof.Dr. İnci Sarıçiçek

Üye: Prof.Dr. Özden Üstün

Üye: Dr.Öğr.Üyesi R. Aykut Arapoğlu

Üye: Dr.Öğr.Üyesi İlkay Orhan

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Hürriyet ERŞAHAN
Enstitü Müdürü

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Prof.Dr. Muzaffer KAPANOĞLU danışmanlığında hazırlamış olduğum "Askeri Hava Taşıma Probleminde Risk Yönelimli Çizelgeleme–Rotalama ve Çözüm Önerileri" başlıklı DOKTORA tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim.

Bu tez çalışmasında belirtilen görüş ve yorumlar tarafıma aittir. Türk Silahlı Kuvvetleri'nin ya da diğer kamu kuruluşlarının görüşlerini yansıtmaz. 24.10.2018

Özgür AKSOY

İmza

ÖZET

Her faaliyette olduğu gibi askeri faaliyetlerde de lojistik operasyonların etkin ve minimum maliyetlerle yapılabilmesi büyük önem taşımaktadır. Hava kuvvetlerindeki lojistik faaliyetlerin en hayati bileşeni hava taşımalarıdır. Hava taşımalarına ait maliyetlerin en büyük kısmını uçaklara ait uçuş maliyetleri (yakıt, personel vb.) oluşturur. Bu nedenle, hava taşımalarında kullanılan hava araçlarına ait rotaların optimize edilmesi çok önemlidir. Bu çalışmaya konu olan Askeri Hava Taşıma Problemi (AHTP), literatürdeki talep anında taşıma (On Demand Transportation) probleminin bir varyantı olup, önemli özellikleri arasında üslere (müşterilere) ait taleplerin farklı uçaklarla taşınabilmesi (bölünebilen talepli), her üssün bir talep noktası ve/veya depo olarak davranabilmesi, eş zamanlı toplama ve dağıtma yapılabilmesi ve kapasiteli heterojen hava araçlarının kullanılması sayılabilir. Ayrıca savaş ve gerginlik gibi zamanlarda kargo uçaklarının karşılaşabilecekleri riskler de göz önünde tutulmuş ve uçakların maruz kalacağı riskler azaltılmaya çalışılmıştır. Kötü hava koşullarında iniş kalkış yapan uçakların maruz kaldığı risk, önerilen bir bulanık kontrolcü ile modellenerek rota optimizasyonunda kullanılmıştır. AHTP için bir matematiksel model geliştirilmiş olup küçük boyutlu problemler matematiksel model ile çözülmüştür. Daha büyük boyutlu problemlerin çözülebilmesi için bir Hücresel Genetik Algoritma önerilmiştir. AHTP'ye ait örnek problemler literatürde olmadığı için yeni deney seti oluşturulmuştur. Oluşturulan problemler önerilen Hücresel Genetik Algoritma ve Klasik Genetik Algoritma ile çözülmüştür. Önerilen Hücresel Genetik Algoritma ile Klasik Genetik Algoritmaların sonuçları karşılaştırılmıştır.

Anahtar Kelimeler: Askeri Hava Taşıma Problemi, Topla Dağıt Araç Rotalama Problemi, Hücresel Genetik Algoritma, Bulanık Kontrol, Tam Sayılı Programlama

SUMMARY

As in every activity, it is vital to carry out logistic operations with minimum cost in military activities. Air transportation is the most vital component of the air force logistic activities. Flight costs of aircraft are the most significant cost of military air transportation. Therefore, it is critical to optimize the aircraft routes used in air transportation. In this study, The Military Airlift on Demand (MAOD) Problem a variant of the On–Demand Transportation problem, which has the features such as split demand, each airbase can act as a delivery as well as a pickup point, simultaneous Pick and Delivery, and capacitated heterogeneous aircraft. In addition, the risks faced by the aircraft were taken into consideration and tried to reduce the risks occurred while flying through the routes during times such as war and tension. The risks of aircraft associated with landing and takeoff in severe weather conditions have been modeled by using a proposed fuzzy controller and used in route optimization. A mathematical model has been developed for the MAOD problem, and small-sized problem instances have been solved with it. A Cellular Genetic Algorithm has been proposed to solve larger sized problems. Since the instances belonging to the MAOD were not found in the literature, new instances have been proposed. The generated instances are solved by the proposed Cellular Genetic Algorithm and the Classical Genetic Algorithm. The results which are generated by the proposed Cellular Genetic Algorithm and Classical Genetic Algorithm are compared.

Keywords: Military Airlift Problem, Pickup and Delivery Problem, Cellular Genetic Algorithm, Fuzzy Control, Integer Optimization Model

TEŞEKKÜR

Doktora çalışmam boyunca bana danışmanlık yaparak beni yönlendiren, daima yapıcı ve yol gösterici olan, bilgi ve deneyimlerini benimle paylaşan, akademisyen kimliği ile bana rol model olan değerli danışmanın Prof.Dr. Muzaffer Kapanoğlu'na sonsuz teşekkürlerimi sunuyorum. Tez izleme jürimde yer alarak tezimin hazırlanmasında yardımcı olan Prof.Dr. Özden Üstün'e ve Dr.Öğr.Üyesi R. Aykut Arapoğlu'na değerli katkılarından ve göstermiş oldukları sabırdan dolayı teşekkür ederim.

Gösterdikleri anlayış ile daima yanımda oldukları için aileme sonsuz teşekkürlerimi sunuyorum.

İÇİNDEKİLER

Sayfa

ÖZET vi
SUMMARY vii
TEŞEKKÜR viii
İÇİNDEKİLER ix
ŞEKİLLER DİZİNİ xii
ÇİZELGELER DİZİNİ xiii
1. GİRİŞ VE AMAÇ 1
2. ASKERİ HAVA TAŞIMA PROBLEMİ (AHTP) 6
2.1. Problemin Tanımı 6
2.1.1. Nakliye uçaklarına ait kısıtlamalar 9
2.1.2. Pilotlara ilişkin kısıtlamalar 9
2.1.3. Askeri uçuş ağı kısıtlamaları 10
2.1.4. Risk faktörlerine ait kısıtlar 10
2.2. AHTP'nin Sınıflandırılması 11
3. LİTERATÜR ARAŞTIRMASI 18
3.1. Sezgisel Yöntemler 18
3.2. Meta Sezgisel Yöntemler 20
4. AHTP'NİN ANALİZİ VE BULANIK HAVA KOŞULLARI RİSKİ 22
4.1. AHTP için Yapılan Varsayımlar 22
4.2. AHTP'nin Analizi 23
4.2.1. AHTP'nin talep yapısı 23
4.2.2. AHTP'nin karakteristik özellikleri 26
4.2.2.2. Hasarlı pist durumu 28
4.2.2.3. Rota farklılaştırma 28
4.2.2.4. Hava savunma sistemlerine uzaklık 29
4.2.2.5. Alçak irtifa uçuş 30
4.2.2.6. Hava koşulları riski 31
4.3. Hava Koşulları Riskinin Bulanık Mantık ile Sayısallaştırılması 33
4.3.1. Bulanık mantık kavramı 33

İÇİNDEKİLER (devam)

Sayfa

4.3.2. Bulanık kontrol 34
4.3.2.1. Girdi 34
4.3.2.2. Bulanıklaştırma birimi 35
4.3.2.3. Kural tabanı 38
4.3.2.4. Sonuç çıkarım birimi 40
4.3.2.5. Durulaştırma birimi 40
4.3.2.6. Bulanık Kontrol Çıktısı 40
5. MATEMATİKSEL MODEL 42
5.1. Notasyon 42
5.2. Matematiksel Model 44
5.2.1. Amaç fonksiyonu 44
5.2.2. Kısıtlar 44
5.3. Matematiksel Model ile Örnek Bir Problemin Çözümü 46
6. MATERYAL VE YÖNTEM 49
6.1. Klasik Genetik Algoritma 49
6.2. Hücresel Genetik Algoritma 51
6.2.1. Senkronize (Paralel) hücresel genetik algoritma 53
6.2.2. Sıralı hücresel genetik algoritma 53
6.2.3. Hücresel genetik algoritmanın adımları 54
6.3. AHTP için Önerilen Hücresel Genetik Algoritma 55
6.3.1. Genetik gösterim (kodlama) 56
6.3.2. Başlangıç popülasyonunun oluşturulması 58
6.3.2.1. Clarke &Wright tasarruf algoritması 58
6.3.2.2. Rastgele rota oluşturma 59
6.3.2.3. Uçak yükleme 59
6.3.2.4. Giren çıkan ürün eşitliği 62
6.3.3. Uyum fonksiyonu 62
6.3.4. Ebeveyn seçimi 63
6.3.5. Çaprazlama 64
6.3.6. Mutasyon 65

İÇİNDEKİLER (devam)

Sayfa

6.3.6.1. Araya sokma mutasyonu 66
6.3.6.2. Değiş tokuş mutasyonu 66
6.3.6.3. Ters çevirme mutasyonu 66
6.3.7. Yer değiştirme 67
6.4. hGA'ya Ait Parametre Analizi 68
6.4.1. Tam faktöriyel deney tasarımı 69
6.4.2. İstatistiksel modelin oluşturulması 69
6.4.3. İstatistiksel analiz 71
6.4.3.1. *3–Numaralı* probleme ait analiz 71
6.4.3.2. *14–Numaralı* probleme ait analiz 72
6.4.4. Bulgular 73
7. DENEY SETİNİN OLUŞTURULMASI 75
7.1. Deney Setinin Özellikleri 75
7.1.1. Üslere ait özellikler ve konumları 76
7.1.2. Talep yoğunluğu 76
7.1.3. Taleplerdeki sapmalar 77
7.1.4. Uçak kapasitesi 78
7.2. Örnek Problemlerin Oluşturulması 78
8. BULGULAR VE TARTIŞMA 80
8.1. hGA'nın Ürettiği Çözümlerin İncelenmesi 80
8.2. Test Problemlerinin Çözümleri 85
9. SONUÇ VE ÖNERİLER 90
KAYNAKLAR DİZİNİ 94
EK AÇIKLAMALAR 100
Ek Açıklama-A: Pilotlardan Elde Edilen Verilerin Toplanmış Hali 101
Ek Açıklama-B: Üyelik Fonksiyonları ve Matlab Bulanık Kontrolcü Ara Yüzü 105
Ek Açıklama-C: IBM ILOG CPLEX Kodu 108
Ek Açıklama-D: hGA Matlab Kodu 118
ÖZGEÇMİŞ 129

ŞEKİLLER DİZİNİ

Şekil **Sayfa**

1.1 AHTP'nin Şematik Gösterimi 3
2.1 ARP'ye Ait Sınıflandırma 12
2.2 Zengin ARP İçin Sınıflandırma 13
2.3 TDARP İçin Sınıflandırma-1 14
2.4 TDARP İçin Sınıflandırma-2 15
4.1 AHTP'nin Serim Üzerinde Gösterimi 23
4.2 Radar Çalışma Prensibi 31
4.3 Bulanık Kontrolcü 35
4.4 Görüş İçin Üyelik Fonksiyonları 37
4.5 Bulanık Kontrolcüye Ait Örnek Bir Çıktı 41
6.1 Bir Kromozom Gösterimi 50
6.2 Genetik Algoritmanın Genel Akış Şeması 50
6.3 hGA 'da Komşuluk Örneği 51
6.4 hGA'ya Ait Komşuluk Örnekleri 52
6.5 Sıralı Süpürme 53
6.6 Uçak Rotası ve Uçağa Ait Yükleme/Boşaltmalar 60
6.7 Ürün Akış Dengesi 62
6.8 Von Neumann Komşuluk Yapısı 64
6.9 Çaprazlama Örneği 65
6.10 Araya Sokma Mutasyonu Örneği 66
6.11 Değiş Tokuş Mutasyonu Örneği 66
6.12 Ters Çevirme Mutasyonu Örneği 67
6.13 Tahmin Profili 74
8.1 Üslerin ve Hava Savunma Sistemlerinin Konumları 80
8.2 hGA 'da Popülasyonun Evrimi Grafiği 82
8.3 kGA 'da Popülasyonun Evrimi Grafiği 82
8.4 Ara Jenerasyonda Erişilen En İyi Çözüm (Kullanılan uçak sayısı:4) 83
8.5 Son Jenerasyonda Erişilen En İyi Çözüm (Kullanılan uçak sayısı:2) 83
8.6 Taleplerin Nasıl Taşındığına Dair Bir Örnek 84
8.7 Problem Tipi Bazında hGA ve kGA Karşılaştırması 88
8.8 Problem Özelliği Bazında Karşılaştırma 89

ÇİZELGELER DİZİNİ

Çizelge **Sayfa**

2.1 AHTP'ye Ait Örnek Talep Matrisi 8
2.2 Örnek Uçak Rotaları 8
4.1 Örnek Talep Matrisleri 24
4.2 Örnek Meteorolojik Görüş Formu 36
4.3 Kural Tabanı 39
5.1 Örnek Probleme Ait Talep Matrisi 46
5.2 Örnek Probleme Ait Üs ve Hava Savunma Sistemleri Konumları ve Bilgileri 47
5.3 Örnek Probleme Ait Uçak Bilgileri 47
5.4 Optimum Çözüme Ait Uçak Rotaları 47
5.5 Uçakların Uçtuğu Bacaktaki Taşıdığı Toplam Yük Miktarı 48
5.6 Uçaklar Tarafından Taşınan Taleplerin Miktarları 48
6.1 Örnek Rota Matrisi 57
6.2 Örnek Yükleme Matrisi 57
6.3 Uçak Yüklemeleri İçin Örnek Talep Matrisi 60
6.4 Örnek Yükleme Matrisi 60
6.5 Faktörler ve Düzeyleri 68
6.6 Deney Tasarımı ve Gözlem Sonuçları 70
6.7 Problem 3-Varyans Analizi 71
6.8 Problem 3-Parametre Tahminleri 71
6.9 Problem 14-Varyans Analizi 72
6.10 Problem 14-Parametre Tahminleri 72
7.1 Problem Tiplerine Ait Özellikler 79
8.1 Uçaklara Ait Özellikler 81
8.2 *63-Numaralı* Probleme Ait Talep Matrisi 81
8.3 Uçak-1 Rotasında Taşınan Miktarlar ve Sıraları 85
8.4 hGA ve kGA İle Elde Edilen En İyi Değerler 86
8.5 Problem Tipine Göre Karşılaştırma 88

1. GİRİŞ VE AMAÇ

Lojistik faaliyetler her alanda olduğu gibi orduların faaliyetlerinde de büyük bir önem taşımaktadır. Atatürk lojistiğin önemini aşağıdaki sözleriyle ifade etmektedir.

> "Zaferi; denizi kontrol altında tutan, ihtiyacı olan materyali, gereksinmesi olduğu zaman, istediği yere ulaştırabilen ülke kazanır."

Savaşların kazanılmasında diğer önemli bir faktör de hava gücünün etkin kullanılmasıdır. Düşman topraklarının derinliklerinde, en az yan etki ve en hızlı şekilde hedeflerin yok edilmesi ancak hava gücünün etkin kullanılması ile sağlanabilir. Etkin hava gücünün özellikleri ise esneklik, hassas zamanlama ve yüksek vuruş gücü olarak sayılabilir. Hava gücünün bu eşsiz özellikleri hızlı ve devam ettirilebilir lojistik kapasiteyi gerektirir. Lojistik ihtiyaçların hızla karşılanabilmesi, vurucu güç olan savaş uçaklarının hızla çalışabilir hale getirilebilmesi ancak hava taşıma modunun kullanılması sayesinde gerçekleştirilebilir. Bu yüzdendir ki nakliye uçakları hava kuvvetlerinin lojistik alt yapısının bel kemiğini oluşturmaktadır.

Hava kuvvetlerinin üsleri ülke geneline dağınık olarak konuşlandırılmıştır. Bu dağınık konuşlandırma ile her bölgenin kendine has jeopolitik özelliklerinden faydalanılmıştır. Üslerin bu şekilde dağınık olarak konuşlandırılması jeopolitik yönden faydalar sağlar iken diğer yandan her üs için bulundurmanız gereken personel ve malzeme miktarlarını arttırmaktadır. Üslerde vurucu güç olan savaş uçaklarının veya bu uçakların uçuşa hazır hale getirilmesi için kullanılan araç ve gereçlerin arızalanması durumunda, arızalanan parçanın hızla değiştirilerek uçakların, araç ve gereçlerin tekrar çalışabilir hale getirilmesi çok önemlidir. Uçaklar, araç ve gereçlerin hızla çalışabilir hale getirilememesi halinde savaşın kaybedilmesi kaçınılmazdır. Tarihte bu durumun birçok örnekleri mevcuttur. Arızalanan parçaların en hızlı şekilde yenisi ile değiştirilmesi için her üs, deposunda envanterinde olan araç ve gereçlere ait yedek malzeme stoku tutar. Her üs deposunda her yedek parça için tutulan bu stok miktarları, doğal olarak merkezi bir depoda tutulacak stok miktarının çok altında kalmaktadır. Düşük stok miktarları malzemelerin daha sık tükenmesine ve dolayısı ile üsler arası sık malzeme akışına neden olmaktadır. Stoksuz kalan bir üssün stokunun en kısa zamanda yenilenebilmesi hava taşımasını zorunlu

kılmaktadır. Hava taşımaları, hava kuvvetlerinde nakliye uçakları ile sağlanmaktadır. Nakliye uçakları, ülke genelinde konuşlanmış olan hava üslerinin ve diğer birimlerinin arasında ihtiyaç duyulan her türlü malzemenin, mühimmatın ve personelin taşınmasında öncelikli olarak kullanılmaktadır.

Her üs kendi envanterinde olan araç ve gereçlerin yedek parçalarına ait belirli bir miktarda stok tutar. Bir üste bir malzemeye ait stok bittiğinde veya stokta olmayan bir malzemeye ihtiyaç duyulduğunda, bu malzeme stokunda bulunan bir başka üsten talep edilir. Bunun sonucunda, her üs diğer bütün üsler için bir depo ve aynı zamanda diğer üslerdeki depolar için bir talep noktası olmaktadır. Bir üsten talep edilen malzeme, stoku olan diğer bir üsten alınarak talep üssüne iletilmektedir. Üsler arasında oluşan bu malzeme akışları nakliye uçakları ile yapılmaktadır.

Uçakların uçuş maliyetleri kara araçlarına göre daha yüksektir. Maliyetler kullanılacak uçak sayısı ve uçakların uçacağı rotaların uzunluğu ile doğru orantılı olarak artmaktadır. Bu nedenle maliyetleri azaltmak için kullanılan uçak sayısının ve oluşturulan rotaların kısaltılması gerekir.

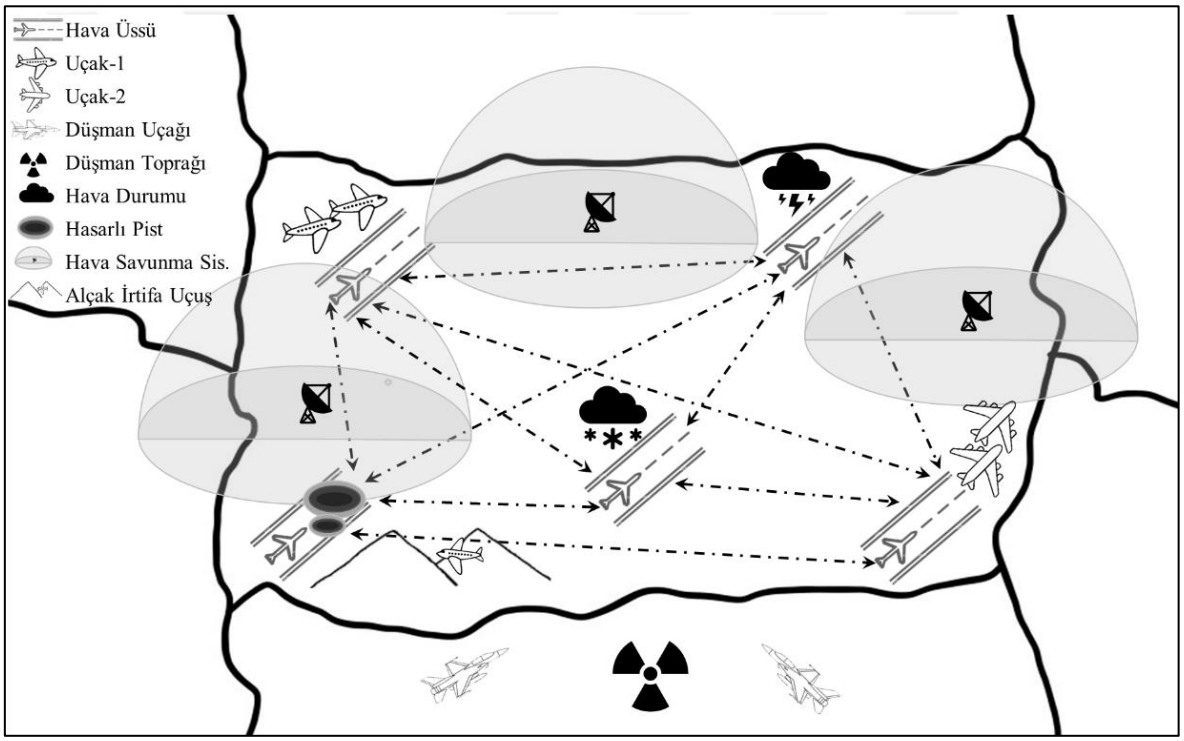
Bu tezde ele alınan problem Askeri Hava Taşıma Problemi (AHTP) olarak adlandırılmıştır. AHTP'ye ait birçok karakteristik özellik vardır. Bu özellikler kısaca;

- Farklı kapasiteye, menzile, hıza ve maliyete sahip heterojen uçaklar,
- Eş zamanlı toplama ve dağıtma yapılması,
- Taleplerin bölünebilir olması,
- Ara duraklara müsaade edilmesi,
- Pilotlar için maksimum uçuş bacağı kısıtı,
- Pilotlar için maksimum görev süresi olarak sayılabilir.

AHTP aynı zamanda, hava taşımalarının başarı ile gerçekleştirilmesi için savaş ve gerginlik anlarında karşılaşılabilecek risk faktörlerini göz önüne almaktadır. Beş farklı risk faktörü olası riskler olarak belirlenmiştir. Bu risk faktörleri,

- Hasarlı pist durumu,
- Rota farklılaştırma,
- Hava savunma sistemlerine uzaklık,
- Alçak irtifa uçuş,
- Hava koşulları riskidir.

Her risk faktörü için bir eşik değeri belirlenmiş ve belirlenen eşik değerinin altında düşük risklere sahip rotaların tercih edilmesi sağlanmıştır. AHTP için şematik bir gösterim Şekil 1.1'de verilmiştir.



Şekil 1.1 AHTP'nin Şematik Gösterimi

Risk faktörlerinden hava koşulları riski bir bulanık kontrolcü ile modellenmiştir. Bir üsse ait hava koşulları verilerinden olan rüzgâr, bulut alt tavanı ve görüş değerleri bulanık kontrolcüye girdi olarak verilmiştir. Çıktı olarak da o üsteki hava koşullarının uçuş için oluşturduğu risk değeri elde edilmiştir. Elde edilen hava koşulu riski, düşük riske sahip rotaların oluşturulmasında kullanılmıştır.

AHTP'nin çözümü için matematiksel bir model tasarlanmıştır. Matematiksel model "Problemdeki Üs Sayısı–2" ara durak sayısı dahil çözümler üretebilmektedir. Oluşturulan tam sayılı modelle, görece küçük boyutlu problemler makul zamanda çözülebilmiştir. Tasarlanan modelin boyutu, üs sayısına bağlı olarak üstel büyümektedir. Pentium i7-7700 işlemciye sahip bir bilgisayarda 8 üs ve 46 talepten oluşan bir örneğin çözümü günlerce sürebilmiştir.

Matematiksel modelin, mevcut bilgisayar işlem gücü ile büyük boyutlu problemleri çözmede yetersiz kalması nedeni ile AHTP'nin çözümü için hücresel bir genetik algoritma (hGA) önerilmiştir. Literatürde AHTP için örnek problemler bulunmadığı için AHTP'ye ait deney seti oluşturulmuştur. Oluşturulan problemler, önerilen hGA ve klasik genetik algoritma (kGA)ile çözülmüştür. Her iki algoritmayla elde edilen sonuçlar tartışılarak karşılaştırılmıştır.

Tezin diğer bölümleri şu şekilde organize edilmiştir. İkinci bölümde, problemin tanımı yapılmış, özellikleri anlatılmış ve kısıtları ortaya konulmuştur. Araç rotalama problemi ve çeşitlerinden bahsedilmiştir. Askeri hava taşıma probleminin araç rotalama problemleri çeşitlerinden hangisine girdiği ve bu sınıflandırma içerisinde diğer türlerden ne gibi farklılıklarının olduğu gösterilmiştir. Bu bölümde son olarak problemin kendine has yapısından dolayı yeni bir problem tipi olarak literatüre kazandırıldığından bahsedilmiştir.

Üçüncü bölümde, AHTP ve benzeri problemleri çözmek için kullanılan çözüm yöntemleri ve karşılaştırmaları yapılmıştır.

Dördüncü bölümde askeri bir problem olan AHTP'nin analizi yapılmış ve karakteristik özellikleri aktarılmıştır. Hava koşullarından dolayı oluşan risklerin sayısal bir değer olarak nasıl elde edildiği, tasarlanan bulanık kontrol mekanizmasının işleyişi ve elde edilen sonuçlar gösterilmiştir.

Beşinci bölümde AHTP'ye ait önerilen karma tam sayılı matematiksel modele yer verilmiştir. Önerilen matematiksel modelin; topla ve dağıt araç rotalama problemi matematiksel modelinden farklı olan kısıtları ve bu kısıtların geçerliliği gösterilmiştir.

Altıncı bölümde Hücresel Genetik Algoritmanın (hGA) en geniş anlamda ne olduğu ve nasıl çalıştığı anlatılmıştır. Daha sonra AHTP'nin çözümü için bir Hücresel Genetik Algoritma önerilmiştir. Önerilen hGA'nın kodlaması, başlangıç popülasyonu, uyum fonksiyonu, ebeveyn seçimi, çaprazlama, mutasyon ve yer değiştirme gibi adımlarına ait detaylar aktarılmıştır. Önerilen hGA'ya ait parametrelerin optimizasyonu için 2^4 tam faktöriyel deney tasarımı yapılmıştır. 2 farklı problem tipi için 12 tekrarlı 2^4 tam faktöriyel deney tasarımı yapılmıştır. Her problem tipi için elde edilen sonuçlara göre genel en iyi parametre değerleri elde edilmeye çalışılmıştır.

Yedinci bölümde literatürde AHTP'ye ait deney seti olmadığı için yeni deney seti oluşturulmuştur. Oluşturulan deney setinin nasıl tasarlandığı ve üretildiği anlatılmıştır.

Sekizinci bölümde oluşturulan deney seti önerilen hGA ve kGA ile çözülmüş, elde edilen sonuçlar karşılaştırılarak yorumlanmıştır.

Dokuzuncu bölümde ise yapılan çalışmaya ait genel bir değerlendirme yapılmış, elde edilen sonuçlar özetlenmiş ve ileriye dönük yapılabilecek çalışmalar aktarılmıştır.

2. ASKERİ HAVA TAŞIMA PROBLEMİ (AHTP)

2.1. Problemin Tanımı

Araç rotalama problemleri (ARP) en genel hali ile bir depo, birden fazla araç ve talepleri olan müşterilerden oluşur. Depodaki ürünlerden talepte bulunan müşterilerin talepleri, eldeki araçlar yardımı ile depodan alınıp müşterilere dağıtılarak karşılanır. Burada amaç, talep noktalarının taleplerini karşılayan araçlara ait rotaların toplam maliyetinin en aza indirilmesidir. Literatürde araç rotalama problemlerine ait birçok çalışma vardır. ARP'ye olan ilgi her geçen gün artmaktadır. ARP alanında yapılan çalışmaların sayısı üstel olarak artış göstermektedir (Eksioglu vd. , 2009). Bu büyüme oranı ARP'nin önemini ortaya koymaktadır.

Bütün araç rotalama problemleri gezgin satıcı probleminin (GSP) bir özel hali olduğu için NP-Zor olarak sınıflandırılabilir. Bunun içindir ki ARP'nin küçük boyutlu olan örneklerinin bile çözülmesi çok zordur. Bu tarz problemlerin çözülmesi, problem boyutu büyüdükçe zorlaşmaktadır. ARP'nin bir alt türü ise Topla-Dağıt Araç Rotalama Problemidir (TDARP). TDARP'nin klasik ARP'den farkı, taleplerin toplama ve dağıtma olarak ikiye bölünmesinden kaynaklanır. TDARP toplama veya dağıtmaların nasıl yapılacağına göre birçok alt problem tipine ayrılmıştır (Savelsbergh, 1995). Bu çalışmada Askeri Hava Taşıma Problemi en genel haliyle bir Topla-Dağıt Araç Rotalama Problemi olarak değerlendirilmiştir.

AHTP'de, araç rotalama problemlerindeki araçlar yerine uçaklar, talep noktaları ve depoların yerine de askeri üsler kullanılmaktadır. Üslerde bulunan jet uçakları ve diğer araç gereçler, o üssün faaliyetlerinin etkinlikle yürütülmesinde büyük rol oynar. Dolayısı ile bu uçakların ve araç gereçlerin faal tutulması veya arızalananların en kısa zamanda onarılması çok önemlidir. Arızalanan uçakların **faal edilme süresi**, o üssün **savaşa hazırlığının** bir göstergesidir.

Her üs kendi faaliyetlerinde kullanmak, arızalanan uçakları, araç ve gereçleri faal edebilmek için belirli ürünlerden stok tutar. Fakat bu stokların merkezi bir depodaki kadar yüksek tutulması mümkün değildir. Düşük stok miktarları zaman zaman bir üsteki stokun bitmesine veya çok azalmasına neden olmakta, stoku azalan veya biten ürünlerin diğer üslerden temin edilmesini gerektirmektedir. Her üssün, her ürün için bir stokunun olması aynı zamanda diğer üsler için bir depo vazifesi görebilmesini sağlar. Aynı zamanda her üs elinde olmayan malzemeler için bir talep noktasıdır. Yani her üs **hem bir talep noktası hem de bir depo** olarak davranabilmektedir. Bunun yanı sıra çeşitli görevler nedeni ile bir üsten diğerine gitmesi gereken personel de bir üssün diğer bir üsten talebi olarak değerlendirilir. Taleplerin depodan alınıp ihtiyaç duyulan üsse götürülmesi için nakliye uçakları kullanılır. Talep noktaları birer askeri hava üssüdür. Bu hava üsleri farklı tip hava araçlarının ana üsleri olarak görev yapar. Yolcu taşıyan bir yolcu uçağının her an uçuşa hazır olması, uçağın sahibi şirketin karı için ne kadar önemli ise hava kuvvetlerindeki her hava aracının ve bu araçları kullanacak olan personelin her an göreve hazır olması da hava kuvvetleri için o ölçüde önemlidir (Best, 2014). Askeri hava taşıma probleminde amaç, uçakların uçtukları rotalarından kaynaklanan toplam uçuş maliyetlerinin (birim uçuş saati sonucunda oluşan maliyet) minimuma indirilmesidir.

Üslere ait talepler önceden kesin olarak bilinmektedir. Her üs bir gün önceden ihtiyacı olan ürünlere ait taleplerini belirler. Planlayıcılar tarafından bir gün önce alınan bu taleplere göre ertesi günün uçuş planı hazırlanır. Uçuş planında hangi uçağın hangi üsten hangi miktarda malzeme alacağı ve uçakların izleyeceği rotalar belirtilir. Üsler arasındaki mesafeler iki boyutlu Öklid (Euclidean) mesafesidir ve üçgensel eşitsizliği sağlar.

Her üs, herhangi bir malzeme için depo olabilirken aynı zamanda bir başka malzeme için de talep noktası olabilir. Her bir üs hem bir talep hem de bir depo noktasıdır. Örneğin *1–numaralı* hava üssünde bir hava aracının faal edilebilmesi için ihtiyaç duyulan malzeme kendi stoklarında yoksa diğer üslere ait stoklar incelenir. Diğer üslerden *2–numaralı* hava üssünün stoku göndermek için uygun olduğu varsayılırsa *2–numaralı* hava üssünden *1–numaralı* hava üssüne bir gönderi isteği çıkarılır. Bu istek *1–numaralı* hava üssüne ait taleptir. Benzer şekilde *3–numaralı* hava üssünün de *1–numaralı* ve *2–numaralı* hava üslerinden bir talebi vardır. Bunun yanı sıra *1–numaralı* hava üssünden *2–numaralı* hava üssüne gitmesi gereken personel vardır. Bu taleplere ait örnek bir talep matrisi Çizelge 2.1'de

gösterilmiş olup taleplerin hepsinin kesin bir çıkış ve kesin bir varış noktası vardır. Talepler bir üsten diğer üste gitmesi gereken bütün malzeme ve personel toplam ağırlık olarak ele alınmıştır. Taleplere ait hacimsel kısıtlar göz ardı edilmiştir. Personel ile birlikte malzeme taşınmasında veya malzemelerin birlikte taşınmaları (mühimmat, yanıcı madde vb.) konusunda da bir kısıtlama olmadığı varsayılmaktadır. *N* adet üs olan bir problemde toplam talep sayısı *1* ile N^2–*N* arasında bir değer alabilmektedir.

Çizelge 2.1 AHTP'ye Ait Örnek Talep Matrisi

(kg)	***1–Nolu Üs***	***2–Nolu Üs***	***3–Nolu Üs***
1–Nolu Üs		80	-
2–Nolu Üs	32		-
3–Nolu Üs	12	29	

Taleplerin karşılanması için uçaklar iki üs arasında direk taşıma veya rota içerisinde ara duraklara uğramalar yapabilir. Örneğin 5 üsten oluşan bir problemde *2–numaralı* üsten *3–numaralı* üsse götürülecek bir talep olsun. Bu talebin karşılanması için uçağın izleyebileceği rotalardan bazıları Çizelge 2.2'de gösterilmiştir.

Çizelge 2.2 Örnek Uçak Rotaları

2-3	2-1-4-3	2-1-4-5-3
2-1-3	2-1-5-3	2-5-1-4-3
2-4-3	2-5-1-3	2-4-1-5-3

Probleme ait kısıtlar dört ana kategoride toplanabilir. Bunlar nakliye uçakları, pilotlar, askeri uçuş ağı ve son olarak risk faktörleri olarak kategorize edilebilir (Aksoy ve Kapanoglu, 2012). Yapılan bu sınıflandırmaya göre probleme ait kısıtlar aşağıda açıklanmıştır.

2.1.1. Nakliye uçaklarına ait kısıtlamalar

AHTP'de farklı tiplerde nakliye uçakları bulunmaktadır. Her uçak tipinin bir ana üssü vardır. Uçaklara ait ana üslerde, uçak bakımları ve arıza işlemleri yapılmaktadır. Ayrıca bu ana üssün bulunduğu bölgede o uçak tipinin pilotları ikamet etmektedir. Dolayısı ile görevini icra eden her uçak ana üssüne geri dönüş yapmaktadır. Her uçak tipinin farklı kapasitesi, hızı, menzili ve bir saatlik uçuş maliyeti vardır. Uçaklara yüklenen malzeme ve personelin toplam ağırlığı uçağın kapasitesini geçemez. Bir üsse inen uçağa malzeme yüklenip malzeme indirilmesi için belirli bir süreye ihtiyaç vardır. Bu çalışmada yükleme ve boşaltmalar için ayrılan süre 15 dakika olarak belirlenmiştir. Bir uçak kalkışını takiben menzilinden daha uzakta olan bir üsse gidemez. Bir uçağın bir üsten kalkıp diğer üsse inmesinden dolayı oluşan maliyet, denklem (2.1) ile hesaplanır. Burada TM: Toplam maliyeti, U: iki üs arasındaki uzaklığı, H: uçağın hızını, BM: uçağın birim süredeki maliyetini ifade eder.

$$TM = \frac{U}{H} \times BM \tag{2.1}$$

2.1.2. Pilotlara ilişkin kısıtlamalar

Pilotların ana üslerinden kalkıp tekrar ana üslerine inene kadar geçen zaman görev süresi olarak adlandırılır. Bir pilotun görev süresi, maksimum görev süresini geçmez. AHTP'de maksimum görev süresi, genel havacılık kurallarında olduğu gibi 15 saat olarak ele alınmıştır. Pilotların yemek yemeleri ve dinlenmeleri için belirli bir zaman görev süresine eklenir. Bir pilotun bir üsten kalkıp diğer bir üsse inmesi bir uçuş bacağı olarak adlandırılır. Bir pilotun görev süresi içerisinde uçtuğu bacak sayısı maksimum bacak sayısından fazla olamaz. Mevcut uçakları kullanabilecek yeterli sayıda pilot olduğu varsayılır.

2.1.3. Askeri uçuş ağı kısıtlamaları

Her nakliye uçağının kendine ait bir ana üssü vardır. Ana üsler uçak tiplerine göre farklılık gösterir. Ana üslerde uçaklara ait bakım ekibi ve teçhizatı bulunmaktadır. Görevini tamamlayan bir uçak ana üssüne geri dönüş yapar. Bir üssün talebinin karşılanmasında birden fazla uçak kullanılabilir. Yani bir üsse ait talep bölünerek (demand split) iki farklı uçak ile karşılanabilir. Bir uçak rotası içerisinde aynı üsse birden fazla kez uğrayabilir. Bu sayede daha az sayıda uçak ile talepler karşılanabilir. Bir üsse gün içerisinde inen ve kalkan uçak sayısı belirli miktardan fazla olamaz. Dolayısı ile bir üsse ait taleplerin karşılanmasında kullanılacak uçak sayısı belirli sayının üzerinde olamaz. Bir üsse inen bir uçağın yükleme veya boşaltma yapması için belirli sabit bir süre verilmiştir.

2.1.4. Risk faktörlerine ait kısıtlar

Muharebe şartlarında karşılaşılabilecek bazı durumlar, her türlü planlamanızı ve yapmakta olduğunuz operasyonları etkileyebilir (Gleason, 2012). AHTP askeri bir problem olduğundan muharebe şartlarında karşılaşılabilecek bazı durumlar çalışmaya dahil edilmiştir. Bu olumsuz koşullar önceden planlamalara dahil edilerek operasyonlara olan etkilerinin minimuma indirilmesi amaçlanmıştır.

Muharebe şartlarından kaynaklanan probleme etki edebilecek faktörler;

- Hasarlı pist durumu,
- Rota farklılaştırma,
- Hava savunma sistemlerine uzaklık,
- Alçak irtifa uçuş,
- Hava koşulları riski olarak ele alınmıştır.

Buradaki her bir olumsuz koşul birer risk faktörü olarak değerlendirilmiştir. Risk faktörlerinin bir eşik değeri vardır. Eşik değerleri problemde kabul edilebilir risk seviyesi olarak adlandırılmıştır. Problemde bir çözüme ait risk seviyesi, kabul edilebilir risk seviyesinin üzerinde ise bu çözümün tercih edilirliği azalmaktadır. Kabul edilebilir risk değerleri planlayıcılar tarafından belirlenmektedir. Planlayıcı, çözümde istenilen kabul edilebilir risk seviyesini değiştirerek riskten kaçınan veya riskleri göze alan farklı sonuçlara

ulaşabilir. Bu sayede planlamanın yapıldığı koşullara veya ihtiyaçlara göre arzu edilen hedefler doğrultusunda çözümler elde edilir. Risk faktörleri dördüncü bölümde detaylı olarak ele alınmıştır.

2.2. AHTP'nin Sınıflandırılması

AHTP ilk bakışta bir araç rotalama problemi olarak görülmektedir ve gezgin satıcı probleminin genelleştirilmiş bir halidir. Fakat AHTP'nin mevcut özellikleri ve talep yapısındaki farklılıklar dikkate alındığında mevcut problem tiplerinden farklılaştığı görülmektedir.

Literatürde ARP'nin birçok farklı çeşidi ile karşılaşmak mümkündür. Eksioglu vd. (2009) ARP için bir sınıflandırma çalışması yapmışlardır. Yapılan bu çalışma Şekil 2.1'de verilmiştir. Bu çalışmada ARP; çalışma tipine, senaryo, fiziksel, bilgi ve veri karakteristiğine göre sınıflandırılmıştır. Daha sonra inceledikleri makalelerin, bu sınıflandırma dallarının ve alt özelliklerinin hangisine/hangilerine girdiğini göstermişlerdir. Bu sınıflandırmaya göre Askeri Hava Taşıma Problemi 1.2.2, 2.1.1, 2.2.1, 2.3.1, 2.5.1, 2.7.1, 2.8.1, 3.1.2, 3.2.1, 3.3.3, 3.5.2, 3.7.2, 3.8.1, 3.9.3, 3.10.1, 3.11.1, 4.1.1, 4.2.1, 4.3.2, 4.4.1, 5.1.2 özelliklerini taşımaktadır.

Braekers vd. (2016) etki faktörü 1,5 üzeri dergilerden, 2009-2015 yılları arasında yayınlanmış ve yılda 5 ve daha fazla atıf almış makaleler arasından seçtikleri 277 makaleyi Şekil 2.1'deki sınıflandırmaya göre kategorize etmişlerdir. AHTP gibi gerçek dünya karakteristiklerinin dahil edildiği ARP varyantlarının gittikçe daha fazla dikkat çektiği sonucuna varmışlardır. İncelenen 277 makalenin %95'inde bir çözüm metodu önerildiği ve bunların %71'inde de meta sezgisel çözüm yöntemi kullanıldığı ortaya konulmuştur. Makalelerin %6'sında, AHTP'de de bulunan 2.2.1-Bölünmüş Dağıtım (split deliveries) özelliğine yer verilmiş olduğunu tespit etmişlerdir. AHTP'nin bu özelliğinin çok az sayıda yayında olduğu görülmektedir.

Lahyani vd. (2015) zengin ARP'nin tanımını yapmışlar ve bu problem tipi için bir sınıflandırma çalışması ortaya koymuşlardır. Senaryo ve problemin fiziksel karakteristikleri olmak üzere iki ana dal ve bu dalların alt dallarından oluşan bir sınıflandırma önermişlerdir.

Önerilen sınıflandırma Şekil 2.2'de verilmiştir. Daha sonra bu sınıflandırmaya göre bir kümeleme analizi yaparak literatürde inceledikleri problemlerden hangilerinin Zengin ARP'ye girdiğini tespit etmişlerdir. Bu sınıflandırmaya göre Askeri Hava Taşıma Problemi 1.1.1, 1.2.1, 1.3.2, 1.4.2, 1.5.1, 1.6.1, 1.7.1, 2.1.1.2, 2.1.2.1, 2.1.3.2, 2.1.4, 2.1.5.1, 2.1.6, 2.4, 2.5, 2.6.1 özelliklerini taşımaktadır. Yapılan kümeleme analizi AHTP'ye uygulandığında senaryo karakteristiğinden 4, problemin fiziksel karakteristiğinden 9 puan almış olup C_1 kümesinin bir elemanı yani Zengin ARP olarak sınıflandırılabileceği görülmüştür.

1. Çalışma Tipi
   - 1.1. Teori
   - 1.2. Uygulanan metot
     - 1.2.1. Kesin yöntemler
     - 1.2.2. Sezgiseller
     - 1.2.3. Simülasyon
     - 1.2.4. Gerçek zamanlı metotlar
   - 1.3. Belgelenen uygulama
   - 1.4. Anket, gözden geçirme ve meta arama
2. Senaryo karakteriskleri
   - 2.1. Rota üzerindeki durak sayısı
     - 2.1.1. Bilinen (Deterministtik)
     - 2.1.2. Kısmen bilinen, kısmen stokastik
   - 2.2. Yük bölme kısıtı
     - 2.2.1. Bölünebilir
     - 2.2.2. Bölünemez
   - 2.3. Müşteri servis talebi miktarı
     - 2.3.1. Deterministtik
     - 2.3.2. Stokastik
     - 2.3.3. Belirsiz[1]
   - 2.4. Yeni müşterilerin talep zamanı
     - 2.4.1. Deterministtik
     - 2.4.2. Stokastik
     - 2.4.3. Belirsiz
   - 2.5. Servis/bekleme zamanları
     - 2.5.1. Deterministtik
     - 2.5.2. Zamana bağlı
     - 2.5.3. Araca bağlı
     - 2.5.4. Stokastik
     - 2.5.5. Belirsiz
   - 2.6. Zaman penceresi yapısı
     - 2.6.1. Gevşek zaman penceresi
     - 2.6.2. Sıkı zaman penceresi
     - 2.6.3. İkisinin karışımı
   - 2.7. Zaman ufku
     - 2.7.1. Tek periyot
     - 2.7.2. Çok periyot
   - 2.8. Geri dönüş yüklemeleri
     - 2.8.1. Eşzamanlı topla ve dağıt
     - 2.8.2. Sadece alma veya sadece dağıtma
   - 2.9. Düğüm/kenar
     - 2.9.1. Öncelik ve eşleşme kısıtları
     - 2.9.2. Alt küme kapsama kısıtları
     - 2.9.3. Tekrar rotalamaya izin verilmesi
3. Problemin fiziksel karakteristikleri
   - 3.1. Taşıma ağ tasarımı
     - 3.1.1. Yönlü ağ
     - 3.1.2. Yönsüz ağ
   - 3.2. Müşterilerin yerleri
     - 3.2.1. Düğüm üzerinde müşteriler
     - 3.2.2. Kenar üzerinde müşteriler
   - 3.3. Müşterilerin coğrafik yerleri
     - 3.3.1. Kentsel (Bir desene göre dağılım)
     - 3.3.2. Kırsal (Rastgele dağılım)
     - 3.3.3. Karma
   - 3.4. Başlangıç noktası sayısı
     - 3.4.1. Tek başlangıç noktası
     - 3.4.2. Çok başlangıç noktası
   - 3.5. Yükleme/Boşaltma noktası sayısı (Depo)
     - 3.5.1. Tek depo
     - 3.5.2. Çok depo
   - 3.6. Zaman penceresi tipi
     - 3.6.1. Müşterilerde kısıtlama
     - 3.6.2. Yollarda kısıtlama
     - 3.6.3. Depolarda/Merkezlerde kısıtlama
     - 3.6.4. Sürücülerde/Araçlarda kısıtlama
   - 3.7. Araç sayısı
     - 3.7.1. *n* araç (GSP bu sınıfta)
     - 3.7.2. *n* araca kadar
     - 3.7.3. Limitsiz araç sayısı
   - 3.8. Kapasite değerlendirmesi
     - 3.8.1. Kapasiteli araçlar
     - 3.8.2. Kapasitesiz araçlar
   - 3.9. Araç Homojenliği
     - 3.9.1. Benzer araçlar
     - 3.9.2. Yüke özel araçlar[2]
     - 3.9.3. Heterojen araçlar
     - 3.9.4. Müşteriye özel araçlar[3]
   - 3.10. Seyahat zamanı
     - 3.10.1. Deterministik
     - 3.10.2. Fonksiyona bağımlı (şu anki zamanın bir fonksiyonu)
     - 3.10.3. Stokastik
     - 3.10.4. Belirsiz
   - 3.11. Taşıma maliyeti
     - 3.11.1. Seyahat zamanına bağlı
     - 3.11.2. Mesafeye bağlı
     - 3.11.3. Araca bağlı[4]
     - 3.11.4. Operasyona bağlı
     - 3.11.5. Gecikmenin bir fonksiyonu
     - 3.11.6. Maruz kalınan tehlike/riske bağlı
4. Bilgi karakteristikleri
   - 4.1. Bilginin gelişimi
     - 4.1.1. Statik
     - 4.1.2. Kısmen dinamik
   - 4.2. Bilginin kalitesi
     - 4.2.1. Belirli (deterministik)
     - 4.2.2. Stokastik
     - 4.2.3. Tahmini
     - 4.2.4. Belirsiz (Gerçek zamanlı)
   - 4.3. Bilginin geçerliliği
     - 4.3.1. Yerel
     - 4.3.2. Genel
   - 4.4. Bilginin işlenmesi
     - 4.4.1. Merkezi
     - 4.4.2. Merkezi olmayan
5. Data karakteristikleri
   - 5.1. Kullanılan data
     - 5.1.1. Gerçek dünya datası
     - 5.1.2. Sentetik data
     - 5.1.3. Hem gerçek hem sentetik data
   - 5.2. Data kullanılmamış

[1] Belirsiz, bilginin gerçek zamanlı olarak ortaya çıktığını ifade eder
[2] Her araç ancak belirli bir tip yükü taşıyabilir
[3] Bir müşteri ancak belirli bir araç tipi tarafından ziyaret edilebilir
[4] Bir aracın işletme maliyeti göz ardı edilemez

Şekil 2.1 ARP'ye Ait Sınıflandırma

1. Senaryo karakteristikleri

1.1.Giriş datası
- 1.1.1. Statik
- 1.1.2. Dinamik
- 1.1.3. Deterministik
- 1.1.4. Stokastik

1.2.Karar yönetimi bileşenleri
- 1.2.1. Rotalama
- 1.2.2. Envanter ve rotalama
- 1.2.3. Yer seçme ve rotalama
- 1.2.4. Rotalama ve şoför çizelgeleme
- 1.2.5. Üretim ve dağıtım planlama

1.3.Depo sayısı
- 1.3.1. Tek
- 1.3.2. Çok

1.4.Operasyon tipi
- 1.4.1. Al veya dağıt
- 1.4.2. Al ve dağıt
- 1.4.3. Geri toplama
- 1.4.4. Ara ve bin

1.5.Yük bölme kısıtları
- 1.5.1. Yük bölmeli
- 1.5.2. Yük bölmeden

1.6.Planlama periyodu
- 1.6.1. Tek periyod
- 1.6.2. Çok periyod

1.7.Araçların birden çok kullanılması
- 1.7.1. Tek uğrama
- 1.7.2. Çok uğrama

2. Problemin fiziksel karakteristikleri

2.1.Araçlar
- 2.1.1. Tip
  - 2.1.1.1. Homojen
  - 2.1.1.2. Heterojen
- 2.1.2. Sayı
  - 2.1.2.1. Sabit
  - 2.1.2.2. Limitsiz
- 2.1.3. Yapı
  - 2.1.3.1. Kompartımanlı
  - 2.1.3.2. Kompartımansız
- 2.1.4. Kapasite kısıtları
- 2.1.5. Yükleme politikası
  - 2.1.5.1. Kronolojik sıralı
  - 2.1.5.2. Sırasız
- 2.1.6. Şoför kuralları

2.2.Zaman kısıtları
- 2.2.1. Müşteri kısıtlamaları
- 2.2.2. Yol kullanım kısıtlamaları
- 2.2.3. Depo kısıtlamaları
- 2.2.4. Hizmet zamanı
- 2.2.5. Bekleme zamanı

2.3.Zaman penceresi yapısı
- 2.3.1. Tek zaman penceresi
- 2.3.2. Çok zaman penceresi

2.4.Uyumsuzluk kısıtları

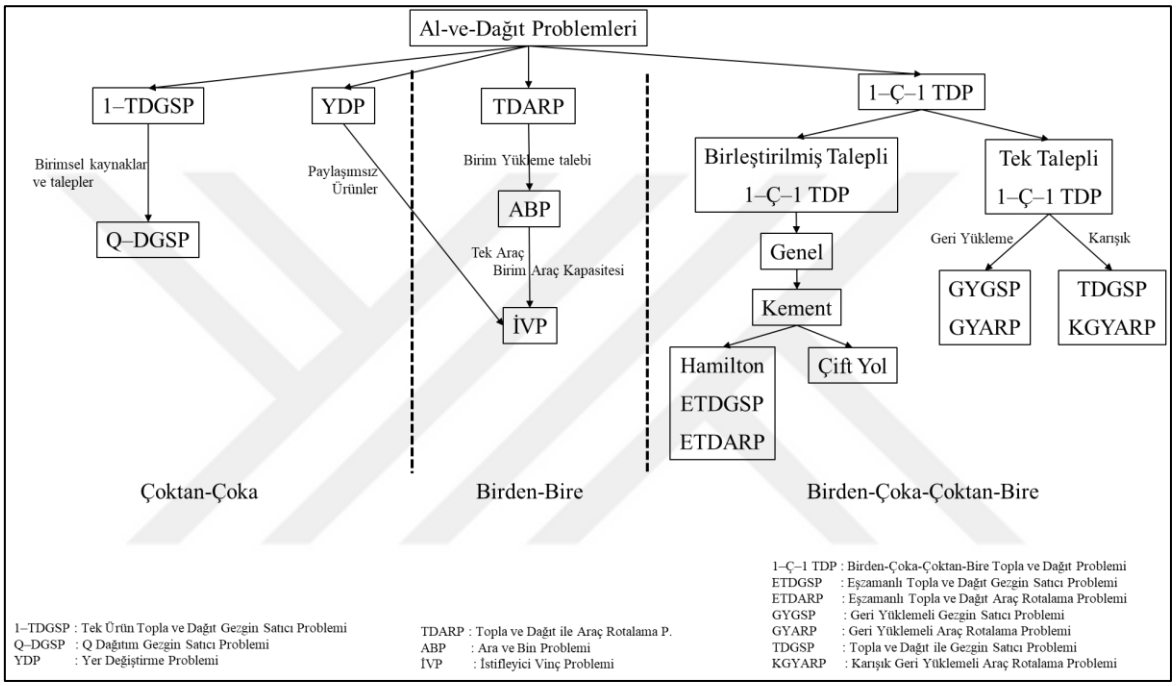
2.5.Belirli kısıtlar

2.6.Amaç fonksiyonu
- 2.6.1. Tek amaç
- 2.6.2. Çok amaç

Şekil 2.2 Zengin ARP İçin Sınıflandırma

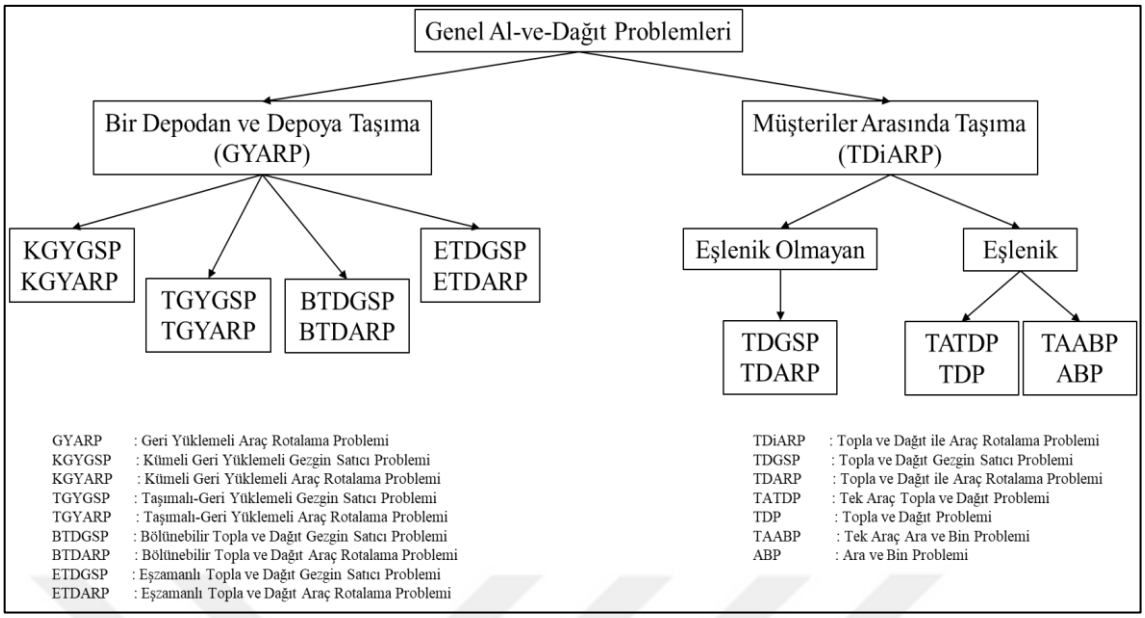
Berbeglia vd. (2007) Topla-Dağıt Araç Rotalama Problemi (TDARP) için bir sınıflandırma çalışması yapmışlardır. Yapılan sınıflandırma Şekil 2.3'de verilmiştir. Bu sınıflandırmayı üç alanlı [Yapı|Uğramalar|Araçlar] şemasını kullanarak yapmışlardır. İlk alan olan yapı ürünlerin depo ve varış yerlerini belirler. *Çoktan-Çoğa* (*Many-to-Many*) tipindeki problemlerde her düğüm ürünler için hem kaynak hem de varış noktası rolünü oynayabilir. *Birden-Çoğa-Çoktan-Bire* (*One-to-Many-to-One*) problemlerinde ürünler depodan müşterilere götürülür ve müşterilerde olan ürünler depoya götürülür. *Birden-Bire* (*One-to-One*) problem tipinde ise her ürünün bir kaynak ve varış noktası vardır. İkinci alan olan uğramalar, bir düğüme uğrandığında o düğümde alma ve dağıtma işlerinin nasıl olduğu (aynı anda topla dağıt, sadece topla veya sadece dağıt gibi) konusunda bilgi verir. Son alan

olan araçlar ise çözümde kullanılan araçların sayısını verir. Şekil 2.3'de statik tipteki Topla-Dağıt problemlerinin birbirleriyle olan ilişkilerini açıklamaya çalışmışlardır. AHTP bu sınıflandırmaya göre hem *Çoktan-Çoğa* hem de *Birden-Bire* problem sınıflarının özelliklerini taşımaktadır. *Çoktan-Çoğa sınıfında* olduğu gibi her üs hem bir depo hem de talep noktasıdır. *Birden-Bire* sınıfında olduğu gibi her ürünün belirli bir çıkış ve belirli bir varış noktası vardır. AHTP bu sınıflandırmadaki birden çok sınıfın özelliklerini taşımaktadır.



Şekil 2.3 TDARP İçin Sınıflandırma-1

TDARP için yapılan bir başka sınıflandırmada da iki ana dal oluşturulmuştur (Parragh vd. , 2008a, Parragh vd. , 2008b). İlk ana dalda ürünlerin müşteriler ile depo arasında taşınması, diğer ana dalda ise ürünlerin toplama ve dağıtma noktaları arasında taşınması incelenmiştir. Bu sınıflandırma Şekil 2.4'de gösterilmiştir. İkinci ana dal, eşlenik ve eşlenik olmayan olmak üzere iki alt dala daha ayrılmıştır. Eşlenik olmayan alt daldaki problemlerde genellikle tek bir ürün tipi mevcut olup herhangi bir noktadan alınan ürün herhangi bir noktadaki talebi karşılamak için kullanılabilir. Eşlenik dalında ise taleplerin bir çıkış ve varış noktası vardır. AHTP bu dalda değerlendirilebilir.



Şekil 2.4 TDARP İçin Sınıflandırma-2

Cordeau vd. (2007) yazdıkları bu kitap bölümünde talep üzerine taşımacılıktan bahsetmektedir. Talep üzerine taşımacılık, kullanıcıların isteği üzerine yolcuların veya eşyaların belirli varış ve çıkış noktaları arasında taşınmasıyla ilgilenir. Ara-Bin (Dial-a-Ride) problem tipine, yaşlı veya engelli kişilerin taşınması, kurye servisleri, uçak paylaşımı ve acil durum araç sevkiyatı örnekler olarak verilebilir.

Cordeau vd. (2007) talep üzerine taşımacılık probleminin bir alt kolu olan klasik Ara-Bin probleminin hava taksi servislerini temsil için yetersiz kaldığını, yeni bir olgu olan hava taksi servislerinin literatürde **yeri olmadığını veya çok az yeri** olduğunu söyleyip Uçak-Ara-Bin Problemini-UABP (Dial-a-Flight Problem-DAFP) önermişlerdir. Literatürde bu problem tipine hava taksi problemi (Air Taxi Problem) veya koltuk talebi üzerine hava taşımaları (Per Seat on Demand) denilebilmektedir. UABP talep yapısı bakımından literatürdeki diğer araç rotalama problemlerinden ayrılmaktadır. AHTP ile UABP benzer özellikler göstermektedir. Fakat AHTP, UABP'de iki olarak sınırlandırılmış olan ara durak sayısı bakımından farklılık göstermektedir.

Espinoza vd. (2008) tarafından ele alınan Statik Uçak-Ara-Bin Problemi (SUABP), bir gün içindeki yolculara ait ulaştırma talepleri ile ilgilenir. Bu problemde; her yolcu için bir çıkış, en erken çıkış zamanı, varış noktası ve en geç varış zamanı vardır. Uçaklar birbirinin aynıdır ve belirli bir kapasiteye sahiptirler. Her uçağın bir ana üssü vardır ve

rotasını tamamlayan uçak ana üssüne döner. Pilotlar günde 8 saatten fazla uçamazlar ve görev süreleri 14 saati geçemez. Bir uçağın bir noktadan belirli bir noktaya gitmesinin belirli bir maliyeti vardır. İniş gerçekleştiren bir uçak minimum 30 dakika sonra tekrar kalkabilir. Kabul edilebilir bir servis sağlanabilmesi için bir yolcunun maksimum bir ara durağa inişine (Intermediate Stop) müsaade edilir. Amaç, maliyetlerin minimuma indirilmesidir. Aşağıda UABP'ye ait örnek çalışmalara yer verilmiştir.

Espinoza vd. (2008) yaptıkları bu çalışmada, 2008 yılından sonra faaliyetlerini durduran "Day Jet" hava taksi firması için tam sayılı çok ürünlü ağ akış modelini geliştirmişlerdir. Şirkete ait örnek problemleri makul zaman içinde çözerek daha iyi çözümler elde etmişlerdir.

Lee vd. (2008) tek hizmet sağlayıcı hava taksi modelini oluşturmuşlardır. Bu modelle haftalık yolcu ve uçak akışının nasıl oluştuğunu karakterize etmişlerdir. Daha sonra yolcu ve uçak akışlarının, kazançların ve boş geri dönüş uçuşlarının miktarının ölçülmesinde modelin kullanılabileceğini göstermişlerdir.

Fagerholt vd. (2009) Norveç'te hava taksi hizmeti vermek isteyen bir şirket için önemli stratejik kararların alımını destekleyecek bir yöntem ve benzetim çalışması yapmışlardır. Bu çalışmada sadece direk uçuşlara izin vermişlerdir.

Keysan (2009) tezinde, koltuk talebi üzerine hava taşımaları (Per Seat On Demand-PSOD) operasyonlarındaki iki ana planlama problemine cevap vermiştir. İlki uçak filosunun periyodik bakım takviminin belirlenmesidir. İkinci ise üsler için yer seçimi ve bu üslerde kaç uçak bulundurulacağıdır.

Campbell (2013) tezinde, Okavango Delta-Botswana bölgesinde turistlere yönelik faaliyet gösteren bir hava yolu şirketine ait hava taksi çizelgelemesi ile ilgilenmiştir. Üç farklı sezgisel algoritma önermiştir ve bu algoritmaların işleyişlerinin değerlendirildiği sayısal çalışmalar yapmıştır.

Bahsedilen ARP'ye ait sınıflandırmalar incelendiğinde AHTP'nin tek bir sınıfa ait olduğu söylenememektedir. AHTP birçok sınıfın belirli özelliklerini içermektedir. Bunun yanında AHTP'ye ait talep matrisi dikkate alındığında yukarıda sözü geçen ARP'lerin tamamından farklıdır. Fakat AHTP, son yıllarda yeni görülen ve Cordeau vd. (2007)'ne ve Campbell (2013)'ye göre literatürde yok denilebilecek kadar az çalışılmış olan SUABP'ye benzemektedir. SUABP'de müşterilere kabul edilebilir bir taşıma hizmeti sağlayabilmek amacıyla maksimum bir ara durağa inişe müsaade edilirken AHTP'de bu kısıt gevşetilmiştir. AHTP ve SUABP birbirine benzemekle birlikte AHTP'de gevşetilen bir ara durağa iniş kısıtı ile birbirinden ayrılmaktadır. Çizelge 2.2'de gösterildiği gibi AHTP'de bir talebin karşılanabilmesi için birçok farklı yol izlenebilmektedir. Bir ara durağa iniş kısıtının gevşetilmesi probleme ait çözüm uzayının büyümesine neden olmaktadır. N adet üsten oluşan örnek bir problemde bir üsten diğer bir üsse gidecek bir müşterinin taşınabilmesi için SUABP'de kullanılabilecek rota parçalarındaki üs sayısı en fazla ***dört*** iken AHTP'de bu sayı yaklaşık $3*(N-2)!$ olmaktadır.

AHTP bulundurduğu özellikler ve özel talep yapısı nedeni ile literatürdeki problemlerden farklılıklar göstermektedir. Bu farklılıklar aşağıdaki şekilde özetlenebilir:

- Her üssün hem bir talep noktası hem de bir depo olarak davranması,
- Taleplerin bölünebilir olması,
- Eşzamanlı toplama ve dağıtma,
- Bir üsse birden fazla kez uğranabilmesi,
- Rota içerisinden sınırsız ara durağa uğranması,
- Askeri riskleri göz önünde bulundurması,
- Heterojen araçlar ve bu araçlara ait farklı ana üslerin olması,

Bu özellikler askeri hava taşıma problemini özgün bir problem haline getirmekte, literatürde yapılan sınıflandırmalara uymamaktadır. AHTP, TDARP değildir çünkü toplama ve dağıtmalara ait birden fazla nokta vardır, yer değiştirme problemi değildir çünkü her noktada birden fazla alınacak ürün vardır, UABP değildir çünkü birden fazla ara durağa uğranabilir. Sonuç olarak AHTP yeni ve özgün bir problem olarak literatüre kazandırılmıştır.

3. LİTERATÜR ARAŞTIRMASI

Gezgin satıcı problemi (GSP) belirli bir sayıdaki noktayı ziyaret eden en kısa yolun bulunmasıyla ilgilenir. Burada her bir noktaya sadece bir kez uğranması koşulu vardır. Bu problem NP-Zor bir problemdir ve bu problemi çözmek için çok fazla çalışma yapılmıştır. Araç rotalama problemlerinin çözüm yöntemleri, genellikle GSP alt problemlerinin çözümünü içerir.

ARP terimi genel olarak bir depo ve müşterilere bu depodaki ürünlerin dağıtıldığı problemleri tarif etmek için kullanılır. Literatürde ARP'nin birçok varyantı mevcuttur. Kapasiteli araç rotalama problemi, zaman pencereli araç rotalama problemi, açık uçlu araç rotalama problemi, topla ve dağıt araç rotalama problemi, ara-ve-bin araç rotalama problemi vb. araç rotalama problemi varyantlarına birkaç örnek olarak sayılabilir. AHTP ise en genel hali ile Topla ve Dağıt Araç Rotalama Problemleri (TDARP) altında değerlendirilebilir.

GSP, ARP ve TDARP doğrusal programlama modelleri ile modellenebilmekte ve çözülebilmektedir. Bu şekilde çözülen problemlerin sonuçları rassallık içermiyorsa ve iyi formüle edilmiş ise optimum çözümü verir. Fakat doğrusal programlama modellerinin çözümleri genellikle bir çeşit sayımlama olan Dal-Sınır algoritmalarına dayanmakta ve çözüm için gereken süre problemden probleme değişmektedir. Çözüm zamanı özellikle değişken ve kısıt sayısı çok olduğunda çok uzayabilmektedir.

Kesin çözüm yöntemlerinin günlük hayatta karşılaşılan problem boyutlarına uygulanmasının zorluğundan dolayı genellikle sezgisel veya meta sezgisel çözüm yöntemleri sık sık kullanılmıştır.

3.1. Sezgisel Yöntemler

Sezgisel yöntemler genel olarak problem boyutunu küçültmeye yönelik çalışarak, örneğin bir değişkeni problemden çıkararak, çözüm bulmaya çalışan yöntemler olarak tanımlanabilir. NP-Zor problemlerin çözümünde genellikle tek başlarına veya diğer yöntemlerle birlikte kullanılırlar. Sezgisel yöntemler en iyi çözümü garanti etmezler fakat

kısa zamanda en iyiye yakın çözüm üretebilmektedirler. Sezgisel yöntemler üç ana grup altında sınıflandırılabilirler. Bunlar

- Tur Kurucu Sezgisel Yöntemler,
- Tur Geliştirici Sezgisel Yöntemler,
- İki Aşamalı Sezgisel Yöntemlerdir.

Tur kurucu sezgiseller değişkenleri ve kısıtları bir araya getirerek ilk uygun çözümü oluşturmaya çalışırlar. Clark&Wright Tasarruf Algoritması, En Yakın Komşuluk Algoritması, Aç Gözlü Algoritma, Ekleme Algoritması bunlara örnek olarak sayılabilir. Tur geliştirici sezgiseller ise var olan bir çözümü geliştirmeye çalışırlar. 2-Opt/3-Opt Algoritmaları, Lin ve Kernighan Algoritması bunlara örnek algoritmalar olarak sayılabilir. İki aşamalı sezgisel yöntemler ise problemin çözüm sürecini kümeleme ve rotalama olarak ikiye ayırarak çözüm ararlar. Süpürme Algoritması, Taç Yaprağı Algoritması, Fisher ve Jaikumar Algoritması bunlara örnek algoritmalar olarak sayılabilir.

J. J. van der Bruggen vd. (1993) tek araçlı topla ve dağıt problemi için bir tur kurucu-geliştirici sezgisel algoritma tasarlamışlardır. Öncelikle bir uygun çözüm oluşturmuşlar ve daha sonra bu çözümü değişken derinlikli değiştirme yöntemi ile çözüme ait uyum değeri bozulmadan geliştirmeye çalışmışlardır. 55 adet talebe kadar olan örnekleri çözmüşlerdir.

Diana ve Dessouky (2004) statik ara-ve-bin araç rotalama problemi için pişman araya sokma (regret insertion) algoritmasını önermişlerdir. Öncelikle bütün talepleri alma zamanlarına göre azdan çoğa göre sıralamışlardır. Daha sonra rotaya sokması zor olan taleplere öncelik sırası vermişlerdir. İlk *m* talep, tohum müşteri olarak kullanılmıştır. Geri kalan bütün talepler pişman araya sokma strateji ile rotaya dahil edilmişlerdir. 1000 taşıma talebine kadar örnekleri çözebilmişlerdir.

Xiang vd. (2006) statik ara-ve-bin araç rotalama problemi için bir tur kurucu-geliştirici sezgisel algoritma tasarlamışlardır. Algoritma ön işlem, tur kurma, tur geliştirme ve yoğunlaştırma işlem adımlarından oluşmuştur. Heterojen araçlar, sürücüler ve müşteriler problemde ele alınmıştır. 2000 talebe kadar olan örnekleri çözebilmişlerdir.

Xu vd. (2001) sütun türetme tabanlı bir sezgisel algoritma önermişlerdir. Alma ve dağıtma noktalarında birden çok zaman penceresi bulunan, yükleme kısıtları olan, şoförlerin belirli çalışma zaman aralığında çalışabildiği, ürün ve araç uygunluğunun kontrol edildiği bir statik topla ve dağıt problemi üzerinde çalışmışlardır. Kolon türetme tekniği ile ana problem ticari bir doğrusal programlama çözücüsü ile çözülmüştür. Fakat alt problemleri çözmek için farklı sezgisel yöntemler kullanılmıştır. 500 talebe kadar olan problem örneklerini çözebilmişlerdir.

3.2. Meta Sezgisel Yöntemler

Meta sezgisel yöntemler, küçük modifikasyonlarla farklı optimizasyon yöntemlerine uygulanabilen belirli bir algoritmaya sahip yapılardır. Büyük boyutlu problemleri daha kısa zamanda çözebildikleri için oldukça popüler yöntemlerdir. Bu yöntemler bütünsel en iyiyi garanti etmezler. Braekers vd. (2016) inceledikleri makalelerin %71'inde, Montoya Torres vd. (2015) ise inceledikleri makalelerin %57'sinde ARP için meta sezgisel yöntemlerin kullanıldığını ortaya koymuşlardır. Genetik algoritmalar, tabu arama, parçacık sürü optimizasyonu, karınca koloni optimizasyonu ve tavlama benzetimi meta sezgisel yöntemlere örnek olarak verilebilir.

Genetik algoritma araç rotalama problemini çözmek için en çok kullanılan yöntemlerden biridir. Karakatič ve Podgorelec (2015) "araç rotalama problemi genetik algoritma" arama terimini kullanarak yaptıkları araştırmaya göre, ARP için genetik algoritmanın kullanım oranının her yıl arttığını ortaya koymuştur. Yaptığı incelemeden elde ettiği sonuçlara göre, genetik algoritmanın araç rotalama türü problemleri çözmek için çok uygun olduğunu göstermiştir.

Baker ve Ayechew (2003) klasik ARP'nin çözümü için başlangıç popülasyonunun üretiminde süpürme algoritması kullanılan bir genetik algoritma önermiştir. Daha sonra önerdikleri bu genetik algoritmayı en yakın komşu arama yöntemi ile melezlenerek ikinci bir algoritma geliştirmişlerdir. Geliştirdikleri ikinci algoritmanın çözüm kalitesi bakımından daha iyi sonuçlar ürettiğini ortaya koymuşlardır.

Laporte (2009) ARP'nin 50'nci yılında, geçmişte ARP için geliştirilen çözüm yöntemlerinden literatüre büyük katkısı olanları derlemiştir. Çözüm yöntemlerini kesin algoritmalar, klasik sezgiseller ve meta sezgiseller olarak sınıflandırmıştır.

Koç vd. (2016) heterojen araçlı ARP'nin, Koç ve Laporte (2018) ise geri dönüş yüklemeli (Backhaul) ARP'nin incelemesini yapmışlardır. Her iki çalışmada da ilgili problemin varyantları ve varyantlarına ait çözüm yöntemleri ortaya konulmuştur.

Montoya Torres vd. (2015) çok depolu ARP (ÇDARP) için en son gelişmeleri yansıtan bir araştırma yapmışlardır. 1984–2014 arasında yayınlanan makaleleri incelemişlerdir. İncelenen tek amaçlı ÇDARP'lerin %21'inde heterojen araç, %38'inde kapasite, %11'inde topla ve dağıt, %5'inde bölünmüş dağıtım kısıtlarının olduğunu ve %57'sinin de meta sezgisel yöntemlerle çözüldüğünü ortaya koymuşlardır. Yukarıda bahsedilen özelliklerin tamamı AHTP'de bulunmakta ve incelenen makalelerin %57'sinin çözümünde kullanılan meta sezgisel yöntemlerden birisi olan genetik algoritmada AHTP'nin çözümünde kullanılmıştır.

4. AHTP'NİN ANALİZİ VE BULANIK HAVA KOŞULLARI RİSKİ

AHTP askeri lojistik faaliyetleri içerisinde yer alan bir problemdir. Üsler arasında yapılan hava taşımalarına ait rotaların optimizasyonu ile ilgilenir. AHTP'nin talep yapısı, literatürdeki araç rotalama problemlerinin tüm varyantlarından farklı bir yapıya sahiptir. Ayrıca AHTP, askeri bir problem olmasından kaynaklanan riskleri göz önünde bulundurarak çözüm arar. AHTP'ye ait bu talep yapısı, içerdiği riskler ve yapılan varsayımlar bu bölümde detaylı olarak ele alınmıştır.

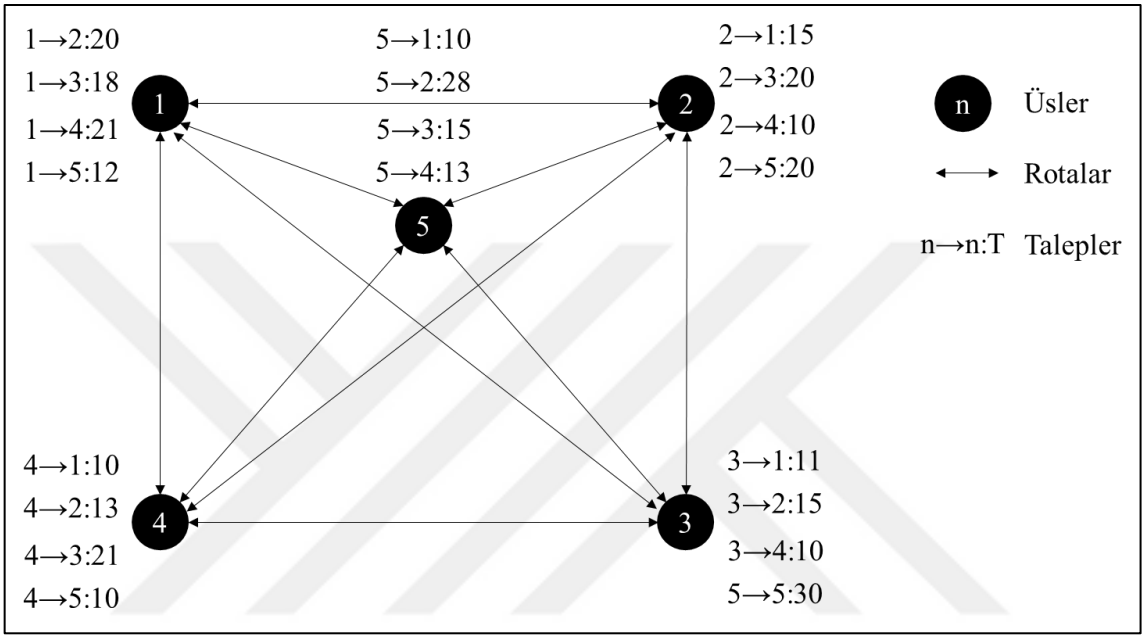
4.1. AHTP için Yapılan Varsayımlar

Askeri hava taşıma problemi ele alınırken yapılan varsayımların neler olduğu bu kısımda anlatılmakta olup maddeler halinde verilmiştir.

- AHTP tek dönem olarak ele alınmıştır. Uçuşlar bir gün önceden planlanmaktadır. Alınan taleplerin tamamı uçuşların icra edildiği gün içerisinde tamamlanmakta ve ertesi güne bırakılmamaktadır.
- Mevcut uçakları kullanacak ekiplerin sayısı planlanabilecek bütün görevleri icra edebilecek miktardadır.
- Taleplerin tamamı problemde verilen uçaklarla ile taşınabilmektedir.
- Uçaklara ait hazırlık, bakım vb. maliyetler dikkate alınmamıştır. Uçak maliyetleri uçakların bir saatlik uçuş maliyeti üzerinden değerlendirilmiştir. Uçuş maliyetleri sabit olarak ele alınmış olup sürat, irtifa ve yüke bağlı olarak değişmemektedir.
- Problemde verilen uçaklar, bir gün içerisinde uçulacak görevleri bakım yönünden icra edebilir durumdadır.
- Pilotların maksimum görev sürelerini aşması durumunda ekip değişimi yapılarak uçakların ana üslerine geri dönmesi sağlanmaktadır.
- AHTP problemi çok amaçlı bir problem olmasına karşın burada uçuşlar sonunda gerçekleşen uçuş maliyetlerinin en aza indirilmesi tek amaç olarak ele alınmıştır. Bununla birlikte diğer amaçlar birer ceza fonksiyonu ile amaç fonksiyonu içerisine dahil edilerek incelenmiştir.

4.2. AHTP'nin Analizi

Askeri hava taşıma probleminin beş üsten oluşan yönsüz tam bağlı bir çizge üzerindeki gösterimi Şekil 4.1'de verilmiştir. Burada düğümler üsleri ifade etmektedir. Her bir üs diğer bir üsse bağlıdır ve diğer üslere bir gönderisi yani talebi olabilir.



Şekil 4.1 AHTP'nin Serim Üzerinde Gösterimi

Askeri hava taşıma probleminin kendine has olan özellikleri ve riskleri problemin daha iyi anlaşılabilmesi için detaylı olarak incelenmiştir. Öncelikle farklı olan talep yapısının nasıl olduğu ve farklılıklarından bahsedilmiştir. Daha sonra problemin askeri bir problem olmasından kaynaklanan riskler ele alınmıştır.

4.2.1. AHTP'nin talep yapısı

AHTP genel olarak bir ARP'dir. Bir üsten alınan bir malzeme diğer bir üsse götürülerek buraya bırakılır. Daha sonra bu üsten alınan bir diğer malzeme diğer bir üsse bırakılır. Bu durum AHTP'nin TDARP olarak tanımlanabileceğini gösterir. Fakat AHTP'de üslere ait talep yapısı ele alındığında, her üssün diğer üslerden bir talebi olabildiği gibi diğer bütün üslerin de bu üsten talebi olabilmektedir. Dokuz düğümlü bir problem için Çizelge 4.1 (a)'da AHTP'ye ve (b)'de klasik TDARP'ye ait örnek talep matrisleri verilmiştir. AHTP'ye

ait matriste ilk satır ve sütun düğümleri gösterirken matristeki rakamlar ise düğümler arasındaki taşıma talep miktarlarını göstermektedir. TDARP'ye ait matriste ilk sütun düğümleri gösterirken diğer rakamlar bu düğümlere ait toplama ve dağıtma talep miktarlarını göstermektedir. Bu örnekten de görülebileceği gibi klasik TDARP'de talep sayısı, N düğümlü bir problem için en fazla $2 \times N$ olmaktadır. Fakat AHTP'de talep sayısı $N^2 - N$ gibi büyük bir rakam olabilmektedir. 100 düğümlü bir TDARP örneğinde talep sayısı 200 iken, 15 düğümlü bir AHTP örneğinde talep sayısı 210 olabilmektedir. Bu durum küçük boyutlu AHTP örneklerinin bile literatürdeki büyük TDARP örneklerine benzer problem büyüklüğüne sahip olduğunu göstermektedir. Buradan hareketle küçük boyutlu AHTP'lerin bile çözümünün hayli zor olacağı görülmektedir.

Talep yapısından görülmektedir ki her üssün diğer bir üsten talebi veya diğer bir üsse gönderisi olabilir. Bir üsse inen bir uçağın o üssün taleplerini bırakırken bu üsten gidecek olan gönderileri de alması gerekmektedir. Bir üsse inen bir uçakta, öncelikle bu üsse ait talepler boşaltılmakta ve daha sonra bu üsse ait gönderiler yüklenmektedir. Yani toplama ve dağıtma işlemleri aynı anda yapılmaktadır. Bu işlemler için AHTP'de yaklaşık 30 dakikalık bir süre öngörülmektedir.

Çizelge 4.1 Örnek Talep Matrisleri

Üsler	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
1		5		4	1		7	2	
2	4		6	5	9	4	5		4
3	9	7		7	9		3	4	7
4	2	5	2		8	9	3	2	8
5		4	5	8		4	7	5	1
6	1	2		5	7		7	8	6
7	3	4	3	6	4	4		9	
8	8	8	5	7	5		9		6
9		3	4		3	8	1	7	

(a) AHTP

Üsler	Toplama	Dağıtma
1	3	3
2	5	4
3	7	6
4	9	9
5	5	4
6	3	5
7	1	8
8	2	6
9	5	7

(b) TDARP

AHTP'de uçaklar bir üsse hiç talep bırakmadan sadece bu üsten yüklenecek olan talepleri almasına müsaade edilmektedir. Yani bir uçak bir üsse bırakılacak talebi olmasa bile bu üsse ait yükleme talebinin yüklenmesi için uğrayabilir.

Bir üsse ait taleplerin etkin ve minimum maliyet ile taşınabilmesi için uçakların bir üsse inmeden önce uğradığı üslerden bu üsse ait gönderilerin hepsinin yüklenmiş olması gereklidir. Örneğin bir uçağın *1-2-3-4-5-6-7-8-1* üslerinden oluşan bir rotayı izlediği düşünülsün. Uçak *5-numaralı* üsse indiğinde, rotada daha önce inilen üslerden *5-numaralı* üsse ait olan gönderilerin yani *5-numaralı* üssün *1,2,3* ve *4 numaralı* üslerden olan taleplerinin bu uçağa yüklenmiş olması gereklidir. Taleplerin toplanmasıyla birlikte aynı anda diğer üslere gönderilecek malzemelerin de dağıtılması gereklidir. Örneğin uçak *5-numaralı* üsse indiğinde, *1,2,3* ve *4 numaralı* üslerden olan taleplerin indirilmesi, *6* ve *1 numaralı* üslere olan gönderilerin ise yüklenmesi gerekmektedir.

Taleplerin bu şekilde yüklenmesi, aynı üsse aynı uçak tarafından rotada birden fazla kez uğranmasını gerektirebilmektedir. *1* ve *2 numaralı* üsler arasında talepler olduğu yani *1-numaralı* üsten *2-numaralı* üsse gidecek ve *2-numaralı* üsten de *1-numaralı* üsse gidecek malzemeler olduğu varsayılsın. Bir uçağın bu talepleri karşılayabilmesi için uçağa ait rotanın *1-2-1* veya *2-1-2* şeklinde olması gerekir. Diğer bir örnekte de *1-numaralı* üsten 2-*numaralı* üsse, *2-numaralı* üsten *1,2,3 numaralı* üslere ve 4-*numaralı* üsten 2-*numaralı* üsse gitmesi gereken malzemeler olduğu durum ele alınsın. *1-2-3-4-1* üslerinden oluşan bir rotası olan uçağın bu talepleri karşılayabilmesi için rotasını *1-2-3-4-2-1* olarak icra etmesi gerekir. Sonuç olarak, bu taleplerin karşılanabilmesi için, bir uçağa ait rota içerisinde aynı üsse birden fazla kez uğranılması kaçınılmazdır.

Talep konsolidasyonu yani bir üsse gelecek taleplerin tek bir uçak ile karşılanması, taşımaların efektif ve düşük maliyet ile icra edilmesi açısından oldukça önemlidir. Bu şekilde bir üssün talepleri tek bir uçakla karşılanabilmekte, birden fazla uçakla hizmet verilmesi engellenmekte ve uçuş trafiği yoğunluğunun azaltılması sağlanabilmektedir. Talep konsolidasyonu büyük miktardaki taleplerin tek bir uçakta taşınmasına ve zaman zaman uçak kapasitesinin aşılmasına neden olmaktadır.

Rotası *1-2-3-1* olan bir uçak ve rotası *4-5-2-6-3-4* olan ikinci bir uçak olduğu varsayılsın. *3-numaralı* üssün *1* ve *2 numaralı* üslerden olan taleplerinin birinci uçak tarafından taşınması gereklidir. Birinci uçak *2-numaralı* üsse inip *3-numaralı* üssün *2-numaralı* üsten olan taleplerini almak istediğinde kapasitesini aştığı varsayılsın. Bu durumda, birinci uçağa bu talebin ya hiç yüklenmemesi ya da taşıyabildiği kadarının yüklemesi gerekmektedir. İkinci seçenek uygulandığında, yani taleplerin bölünmesine ve birden fazla uçak ile taşınmasına izin verildiğinde ***3*** üssünün talebinin bir kısmı birinci uçak ile diğer kısmı ise ikinci uçak ile taşınacaktır. AHTP'de talepler bölünebilir olarak ele alınmaktadır ve bir üsse ait talep birden fazla uçak ile karşılanabilmektedir. Sonuç olarak AHTP'nin yukarıda bahsedilen özellikleri şu şekilde özetlenebilir:

- Üsler hem bir talep noktası hem de bir depo olarak davranır (Multi-Depot).
- Bir rota içerisinde bir üsse birden fazla uğranmasına müsaade edilir (Multi-Trip).
- Toplama ve dağıtma işlemleri aynı anda yapılır (Simultaneous Pickup and Delivery).
- Bir üsse ait talebin birden fazla uçak ile bölünerek taşınmasına müsaade edilir (Split-Delivery).

4.2.2. AHTP'nin karakteristik özellikleri

Bir ülkenin operasyonel askeri gücünün kabiliyeti birçok parametreye bağlıdır. Bu parametrelerden en büyük öneme sahip olanlardan birisi de "Askeri Üslerdir". Her ülkenin farklı amaçlara hizmet eden farklı kabiliyetleri ve araçları bulunan birçok askeri üssü vardır. Bu üslerden havacılık ile ilgili olanlar o ülkeye ait en büyük kuvvet çarpanlarındandır. Yakın zamanda yaşanan operasyonlar göstermiştir ki hava gücü çok büyük öneme haizdir ve diğer kuvvetlerin yapacağı operasyonları büyük ölçüde kolaylaştırmaktadır. Bu derece büyük öneme sahip hava gücünün operasyonel devamlılığı ancak uygun bir lojistik alt yapı ile sağlanabilir. AHTP, bu lojistik alt yapıya hızlı ve maliyet etkin bir çözüm bulmaya çalışan askeri bir problemdir.

Askeri problemlerde düşmandan kaynaklanan olumsuz etkilerin problem içerisinde değerlendirilmesi gereklidir. Muharebe şartlarında, düşmanın olumsuz etkilerinden dolayı karşılaşılabilecek durumlar, her türlü planlamayı etkileyecektir (Gleason, 2012). Bir savaş esnasında düşmana ait bütün uçakların vurulması çok zordur. Fakat bu uçakların faaliyetlerinin devamlılığını sağlayan unsurların ortadan kaldırılması veya uçakların yapacakları görevlerin etkinlikle yapmasının engellenmesi halinde bütün uçaklar yok edilmişçesine yüksek etkilere ulaşılabilir. Bunun tersi düşünüldüğünde, düşman unsurlar da dost unsurlara ait faaliyetleri etkileyecek veya harekât çarpanlarını kullanılmaz hale getirmeye çalışacaktır. Düşman faaliyetlerinden kaynaklanan etkilerin minimuma indirilmesi için alınacak önlemlerin, yapılan görevler üzerinde farklı olumsuz etkileri olabilmektedir. Hem düşmandan kaynaklanan etkilerin hem de alınan önlemlerin olumsuz etkilerinin minimuma indirilmesi, dost kuvvetlerin bekası için çok önemlidir. Bu önlemlere genel olarak pasif korunma tedbirleri denir. Bu pasif önlemlerden bazıları:

- Alçak irtifa uçuş,
- Farklı rotaların kullanılması,
- Hava savunma sistemlerine yakınlık,
- Hasarların en kısa zamanda onarılması olarak sıralanabilir.

Faaliyetlerin emniyet ile icrasında bir diğer önemli etken ise hava koşullarıdır. Planlamalarda göz ardı edilen hava koşullarının, savaşların kaybedilmesine kadar gidebilecek yan etkileri olabilir. Barış şartlarında da hava araçları için hava koşulları son derece önemlidir ve olumsuz hava koşullarının pilotların görevlerini emniyetle icra etmeleri üzerinde büyük etkileri vardır.

AHTP'nin en büyük özelliklerinden birisi de yukarıda bahsedilen muharebe şartlarında karşılaşılabilecek askeri güçlükleri içerisinde barındırmasıdır. AHTP'de bu güçlükler probleme birer risk faktörü olarak eklenmiştir ve bu risklerin hava taşımalarına olan etkisinin minimuma indirilmesi amaçlanmıştır. AHTP'de rotaların oluşturulmasından sonra bu rotalara ait risk değerleri hesaplanır. Her bir riske ait maksimum kabul edilebilir risk değeri vardır ve bu değerler planlayıcı tarafından belirlenebilmektedir. Planlayıcı maksimum değerleri değiştirerek maruz kalınacak risk miktarını değiştirebilmektedir. AHTP'de hesaplanan risk değerlerinin, kabul edilebilir risk değerlerinin altında kalmasını

sağlayacak kısıtlar tasarlanmıştır. Kısıtları sağlamayan yani kabul edilebilir seviyenin üzerinde risklere sahip çözümlerin uyum değerine belirli bir ceza puanı eklenir. Cezalandırılan çözümler çözüm kümesi içerisinde daha az tercih edilen çözümler haline gelir. AHTP için 5 adet risk faktörü tasarlanmış olup aşağıda bu risk faktörlerinin tanımları ve problem içindeki yerleri anlatılmıştır.

4.2.2.2. Hasarlı pist durumu

Savaş zamanlarındaki öncelikli hedeflerden birisi de hava üstünlüğünün elde edilmesidir. Hava üstünlüğünün elde edilebilmesi ancak hava unsurlarının vurulması veya bu unsurların faaliyet göstermelerini sağlayan faktörlerin engellenmesi ile sağlanır. Hava unsurlarının faaliyetlerini etkileyecek en önemli faktörlerden birisi de uçakların havalanması için tek yer olan pistlerdir. Bir üsse ait pistin zarar görmesi o üsse ait tüm uçakların havalanmasını engelleyecek ve hava üstünlüğünün sağlanmasına olumlu katkı sağlayacaktır. Dolayısı ile bir üssün bombalanması ve üsteki pistin zarar görmesi düşmanın hava üstünlüğünü sağlamak için baş vuracağı yöntemlerden birisidir.

Zarar gören pistler, pist çabuk onarım ekipleri tarafından onarılabilmektedir. Onarım süresi hasarın büyüklüğüne göre değişir (Gleason, 2012). Bu çalışmada, bir üsse ait pistin hasar görmesi durumunda onarım süresince bu üsse iniş planlanmamasını sağlayacak kısıtlar tasarlanmıştır. Ayrıca pisti zarar gören üs, ana üs ise bu üssü ana üs olarak kullanan uçak tipi ya da tiplerinin belli bir süre çözümde kullanılmasını engelleyen kısıtlar eklenmiştir. Bir üssün pistinin hasarlı olup olmadığı planlama öncesi kesin olarak bilinmektedir. Düşman tarafından hasarlanan pistler modellenerek, hava taşımalarına olan etkilerinin minimuma indirilmesi hedeflenmiştir.

4.2.2.3. Rota farklılaştırma

Tehlikeli maddelerin taşınmasında riskin tek bir bölgede yoğunlaşmasını önlemek veya bir rotanın sık kullanılması sonucu bu rotadaki uçakların düşman için kolay hedef haline gelmesinin engellenmesi amacıyla, bir bacağın kullanılma sayısının azaltılması gereklidir (Akgun vd. , 2000). Tehlikeli maddeleri taşıyan uçakların aynı rotaları kullanması,

bu rota üzerindeki bölgeler için risk arttıran bir durumdur. Benzer bir şekilde bir rota ne kadar çok kullanılırsa, o rotada uçan uçakların hareketleri önceden tahmin edilebilir ve bu uçaklar düşman için kolay bir hedef haline gelir. Bu çalışmada iki üs arasındaki bir uçuş bacağının bütün uçaklar tarafından iki yönlü olarak belirli bir sayıdan daha fazla kullanılmasını engelleyecek kısıtlar tasarlanmıştır. Bir ayrıtın maksimum kaç defa kullanılacağı planlayıcı tarafından belirlenir. Planlayıcı tarafından bir ayrıtın maksimum kullanım sayısı değiştirilerek, oluşabilecek risklerin dağıtılması sağlanmış ve düşman tarafından önceden kolay kestirilebilir rotaların oluşturulması engellenmiştir.

4.2.2.4. Hava savunma sistemlerine uzaklık

Hava savunma sistemleri, yaklaşan düşman uçaklarının etkisiz hale getirilmesinde ya da manevra kabiliyetlerinin kısıtlanmasında kullanılır. Bu sistemler bulunduğu konumdan belirli bir mesafeye ve yüksekliğe kadar etkili olabilmektedir. Bir radarın havadaki bir uçağı tespit kabiliyeti uçak ile radar arasındaki mesafe ile ters orantılıdır (Zabarankin vd. , 2006). Dolayısı ile dost uçakların rotalarının bu sistemlerin menzilleri içerisinde olması, hatta mümkün olduğunca bunlara yakın rotalardan uçması düşman uçaklarına karşı savunulmasında büyük fayda sağlayacaktır. Bu çalışmada, dost uçakların mümkün olduğunca savunma kalkanı içerisinde uçmalarını sağlayacak bir yöntem geliştirilmiştir.

Uçakların uçtuğu rotalar üsler arasındaki ayrıtlardan oluşmaktadır. Bir ayrıtın orta noktasının her bir hava savunma sistemine olan uzaklığının ortalaması, bu ayrıtın hava savunma sistemlerine olan ortalama uzaklığını verir. Öncelikle, her bir ayrıtın hava sistemlerine olan ortalama uzaklığı hesaplanır. Daha sonra, bir rota içerisindeki ayrıtların hava savunma sistemlerine ortalama uzaklıklarının toplamı, kullanılan ayrıt sayısına bölünmesi ile bu rotaya ait hava savunma sistemlerine ait ortalama uzaklık bulunur. Rotanın hava savunma sistemlerine olan ortalama uzaklığı ne kadar küçük ise o ölçüde bu rotayı kullanan uçak hava savunma sistemlerinin korunması içerisindedir. Bu çalışmada, uçakların rotalarının hava savunma sistemlerine olan ortalama uzaklığı bir risk faktörü olarak belirlenmiştir. Risk faktörünün belirli bir değerden daha küçük olmasını sağlayacak kısıtlar tasarlanmıştır. Planlayıcı tarafından bu risk faktörü değiştirilerek, uçaklarının hava savunma sistemlerinin koruma kalkanının ne kadar içerisinde kalacakları belirlenebilmektedir. Kullanıcı bu değeri küçülttükçe uçaklar koruma kalkanı içerisinde merkeze daha yakın

uçacak, büyülttükçe de korunma kalkanından uzakta yani düşman savaş uçaklarının tehdidine daha çok maruz kalacaktır.

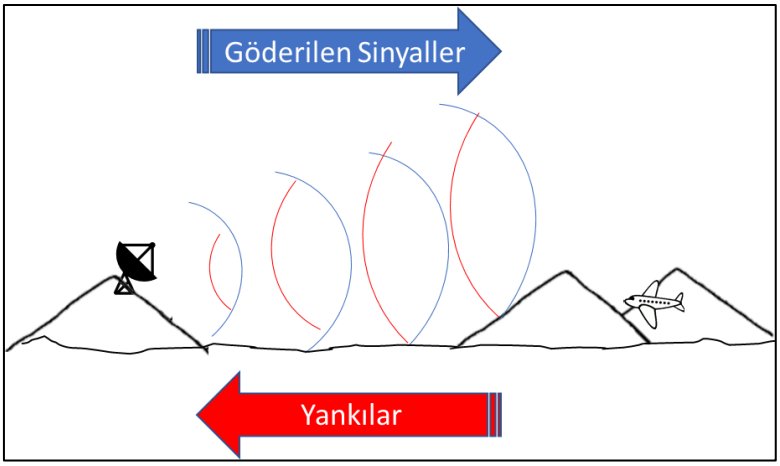
4.2.2.5. Alçak irtifa uçuş

Normal şartlarda kargo uçaklarının ve yüksek değere sahip hava araçlarının korunması için savaş uçakları (Fighter Jets) koruma aracı (Escort) olarak kullanılır. Fakat savaş esnasında av uçaklarının öncelikli görevleri değişmekte, koruma aracı olarak kullanılamamakta ve bunun sonucu olarak kargo uçakları kolay hedefler haline gelebilmektedir. Bu gibi durumlarda alçak irtifa uçuş (yeryüzüne yakın arazi yapısı izlenerek yapılan uçuş) gibi pasif savunma tedbirlerinin kullanımı çok önem taşımaktadır.

Radarlar çevrelerine elektromanyetik sinyal darbeleri gönderirler. Gönderilen radyo dalgalarının taşıdığı enerji herhangi bir nesneye ulaştıktan sonra nesneden yansıyarak geri döner. Yansıyan sinyallere yankı denir. Bir radar için yankının şiddeti yönü ve geri gelme zamanı çok önemlidir. Bu parametreler kullanılarak çeşitli hesaplamalar yapılır ve bu hesaplar doğrultusunda nesnenin yönü, hızı ve özellikleri saptanabilir (Bole vd. , 2014).

Yeryüzü şekillerine çarpan radar sinyalleri bu yüzeylerden geri yansır ve radar ekranında bir nesne olarak algılanmaz. Alçak irtifa uçuşta, pilotlar mümkün olduğunca alçak uçarak arazideki yükseltilerin altında kalmaya çalışır. Bu şekilde alçak uçularak düşman radarlarına ait sinyallerin arazideki yükseltilere çarpması ve uçağa ait yankıların oluşmaması hedeflenir. Şekil 4.2'de radar çalışma prensibi ve alçak irtifada düşman radarlarına görünmez olan bir uçak temsili olarak gösterilmiştir.

Alçak uçuşun düşman radarlarına karşı görünmez olunması gibi yararları olduğu kadar farklı zorlukları ve yan etkileri de vardır. Özellikle kötü görüş şartlarında alçak uçuş çok tehlikelidir. Alçak irtifa uçuş, pilotların çevre kontrolüne azami dikkat etmesini gerektirir (Foyle vd. , 1992). Dikkatin kokpit dışında tutulması, kokpit içi kontrolün ve koordinenin azalmasına neden olur. Aynı zamanda aşırı dikkat yoğunluğu, pilotların çok daha çabuk ve daha fazla yorulmasına neden olur. Dolayısı ile alçak irtifa uçmaktan kaynaklanan risklerin azaltılabilmesi ve pilotların daha az yorulmasının sağlanabilmesi için alçak uçuş süresinin kısa tutulması tercih edilir.



Şekil 4.2 Radar Çalışma Prensibi

AHTP'de planlayıcı bir üsse hangi sektörden (bölgeden) yaklaşılırken veya uzaklaşılırken alçak irtifa uçulması gerektiğini belirleyebilir. Bir uçak bu sektörden geçerek bir üsse gidiyorsa veya bu sektörden geçerek bu üsse inmeyi planlıyorsa bu iki üs arasındaki bacağı alçak irtifada uçması zorunludur. Bir pilotun bu sektörden geçerek yapacağı alçak irtifada uçuş, alçak irtifa uçuş süresi olarak kaydedilir. Bu şekilde bir rota içerisinde pilotların gerçekleştireceği alçak irtifa uçuşların süresi toplanarak o pilota ait toplam alçak uçuş süresi bulunur. Pilotların bir gün içerisinde toplam belirli bir süre alçak irtifa uçuş yapmasına müsaade edilir. Bu süre planlayıcı tarafından belirlenir ve kabul edilebilir alçak irtifa uçuş riski olarak değerlendirilir. Planlayıcı bu süreyi değiştirerek, bir pilotun maksimum uçabileceği alçak irtifa uçuş süresini belirleyebilir. AHTP'de belirlenen alçak irtifa uçuş süresi altında süreye sahip rotaların oluşturulmasını sağlayacak kısıtlar tasarlanmıştır.

4.2.2.6. Hava koşulları riski

Hava koşulları ileriye dönük olarak gerçeğe yakın doğrulukta tahmin edilebilmektedir. Uçuş planlamaları pilotlar tarafından bu hava tahminlerine göre yapılmaktadır. Bu planlamanın yapılmasının amacı, pilotların kötü hava koşullarından dolayı karşılaşacağı riskleri minimuma indirmektir. Harp şartlarında ihtiyaç duyulan malzemelerin, kötü hava koşulları yüzünden ihtiyaç yerlerine ulaştırılamaması istenmeyen

bir durumdur. Dolayısı ile hava taşımalarına ait uçuş rotası planlanmalarında hava koşulları mutlaka göz önüne alınmalıdır.

Bir üsse inişte veya kalkışta pilotlar için hava koşullarına ait üç önemli faktör vardır. Bunlar **görüş**, **bulut alt tavanı** ve **rüzgârdır**. Bu faktörler, inişin emniyetle yapılabilmesini etkileyen en önemli faktörlerdir (FAA, 2010). Görüş, bulut alt tavanı ve rüzgâr faktörlerinin tahmin değerleri kullanılarak her üssün iniş kalkış için elverişlilik miktarı ya da başka bir deyişle iniş ve kalkış için o üsse ait risk miktarının elde edilebilmesi için bu üç farklı değişkenin kullanılarak tek bir risk değerinin elde edilebilmesi gerekmektedir. Bu çalışmada hava koşulları birer bulanık sözel değişken olarak modellenmiş ve daha sonra bu bulanık sözel değişkenler bulanık kontrol yapısı içerisinde kullanılarak üslerin hava durumlarına ait tek bir risk değeri sayısal olarak ifade edilebilmiştir. Tasarlanan bulanık kontrol mekanizması Bölüm 4.3'de detaylı olarak anlatılmıştır.

Elde edilen risk değeri hava koşulları açısından o üsse ait risk değerini yani o üsteki hava koşullarının uçuş açısından elverişlilik miktarını belirler. Bir uçak, rotası boyunca ne kadar çok kötü hava koşulları olan üsse inip kalkar ise o kadar çok riskli bir rota uçmuş olur. Kötü hava koşullarında iniş kalkış gerçekleştiren pilotların yorgunluğu, alçak irtifa uçuşta olduğu gibi artar. Hem karşılaşılan risklerin azaltılması hem de pilotların yorgunluk miktarının azaltılması için mümkün olduğunca daha iyi hava koşullarında iniş kalkış yapılması tercih edilir. Bir uçağın rotası boyunca hava koşulları bakımından riskli üslere inip kalması tercih edilen bir durum değildir ve hava koşulları kötü olan üslere ait taleplerin mümkün olduğunca farklı uçaklarla taşınarak risklerin azaltılması ve dağıtılması istenir.

Bir üsse ait hesaplanan hava koşulları riski, bir uçağın o üsse inip kalmasından sonra bu uçağa eklenir. Bu şekilde bir uçağın rotası boyunca inip kalktığı üslere ait hava koşulları riskleri birikimli olarak toplanır ve o uçağa ait toplam hava koşulu riski bulunur. Toplanan hava koşulu riski, uçağın rotasındaki toplam bacak sayısına bölünerek o uçağa ait ortalama hava koşulları riski elde edilmiş olunur. Ortalama hava koşulları riski, bir uçağın rotası boyunca karşılaşacağı riskin azaltılmasında kullanılmıştır. AHTP'de ortalama hava koşulları riskinin belirli bir seviyenin altında kalmasını sağlayacak kısıtlar tasarlanmıştır. Belirlenen bu seviye maksimum hava koşulu riskidir ve planlayıcı tarafından belirlenir. Planlayıcı, maksimum hava koşulu riskini değiştirerek bir uçağın rotası içerisinde ne kadar kötü hava

koşuluna sahip üsse iniş kalkış yapabileceğini belirler. Hava koşulları riski değeri azaltıldıkça kötü hava koşulları olan üslere farklı uçakların iniş kalkış yapması ve uçakların karşılaşacakları risklerin dağıtılması amaçlanır.

4.3. Hava Koşulları Riskinin Bulanık Mantık ile Sayısallaştırılması

Hava koşullarından görüş mesafesi, bulut alt tavanı ve rüzgâr bir uçağın inişinin emniyetle yapılmasında en önemli faktörlerdir. Bu koşulların etkilerinin minimuma indirilebilmesi için aletle uçuş kuralları ve ekipmanları geliştirilmiştir. Fakat yine de uçak kazalarının büyük bir kısmına hava koşullarının ya doğrudan ya da dolaylı olarak etkisi olmaktadır. İniş veya kalkış esnasında hava koşullarının uçaklar için oluşturduğu riskler, bulanık kontrolle modellenmiştir (Aksoy ve Kapanoğlu, 2014). Bulanık kontrolcünün çıktıları hava koşullarından oluşan risklerin sayısal ifadesidir ve bu değerler AHTP'de hava koşulları risk verisi olarak kullanılmıştır.

4.3.1. Bulanık mantık kavramı

Klasik mantıkta bir önermenin doğruluk değeri, doğrular için 1 ve yanlışlar için 0 olarak kullanılır. E evrensel kümesindeki bir A kümesi, matematiksel olarak $x_A: E \rightarrow \{0,1\}$ fonksiyonu ile karakterize edilir. Burada, A kümesine ait elemanlara 1 değerini, ait olmayan elemanlara ise 0 değerini veren x_A fonksiyonuna A kümesinin karakteristik fonksiyonu denir (Klir ve Yuan, 1995).

Hâlbuki gerçek dünya, hiç de öyle 0 ve 1'den ibaret değildir. 0 ile 1 arasında sonsuz miktarda sayı vardır. Genellikle konuşma dilinde ifade edilen, çoğu zaman sınıflandırmada kullanılan, kesin sınırlarla tanımlanamayan ve kişiden kişiye farklı yorumlanan "çok güzel", "fazla uzun", "aşırı sıcak", "hafif pahalı", "biraz tatlı" gibi bulanık kavramlar, klasik mantığın öngördüğü şekilde incelenemezler. "Ayşe çok güzel", "Hava aşırı sıcak", "Amcam epeyce yaşlı" gibi terimlerle ifade edilen kavramlar, kesin hüküm belirtmediğinden klasik mantık tarafından önerme olarak kabul edilmezler. Bu kavramlarla klasik manada küme tanımlaması yapılamaz. Bu tür önermelere bulanık önermeler ve bunlarla uğraşan mantığa da bulanık mantık denir.

Matematiksel olarak, E evrensel kümesindeki bir bulanık U kümesi $\mu_U(x): E \rightarrow [0,1]$ şeklinde karakterize edilir. Buradaki μ_U fonksiyona bulanık U kümesinin üyelik fonksiyonu denir. Bulanık U kümesi, E'deki her elemanın üyelik derecesiyle birlikte oluşturduğu ikililer kümesidir. U'ya ait bu ikililer kümesi $U = \{(x, \mu_U(x)): x \epsilon E, \mu_U(x) \epsilon [0,1]\}$ ile ifade edilir. Buradaki $\mu_U(x)$ değeri, x'in U kümesine üyelik (aitlik) derecesini gösterir ve [0,1] aralığında herhangi bir değer alabilir (Harris, 2006).

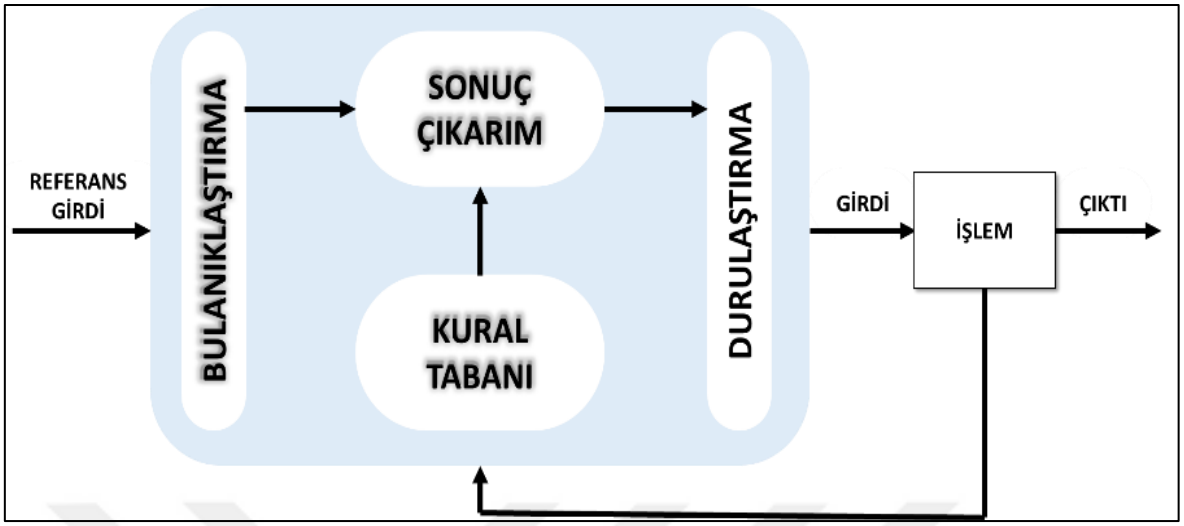
4.3.2. Bulanık kontrol

Bulanık mantık kontrolcüleri, klasik ve modern kontrol teorisinde olduğu gibi kesin ve tam matematik modellere ihtiyaç duymaz. Bulanık kontrol, bulanık mantığın en çok kullanılan alanlarından biridir. Matematiksel eşitlikler yerine sözel ifadelerle kontrol imkânı sağlar. Bu sebepten de bulanık kontrol insanın düşünce yapısına daha yakındır (Soy, 2006).

Bulanık mantık kontrolcülerin dayandığı temel nokta, uzman bir sistem operatörünün bilgi, deneyim, sezgi ve kontrol stratejisini, kontrolcü tasarımında bilgi tabanı olarak oluşturmaktır. Kontrol işlemi uzmanlardan alınan bilgi ve deneyimin bulanık olarak işlenmesi sonucu gerçekleştirilir. Bulanık kontrol sistemi dört temel birimden oluşmaktadır. Bunlar; Bulanıklaştırma birimi (Fuzzification), Kural tabanı (Rule Base), Sonuç çıkarım birimi (Inference Engine) ve Durulaştırma birimidir (Defuzzification). Bu dört temel sistemin bir bulanık kontrolcü diyagramındaki yerleri Şekil 4.3'de verilmiştir (Passino ve Yurkovich, 1998). Aşağıda her birimin işlevi anlatılmıştır.

4.3.2.1. Girdi

Girdi, bulanık kontrolcüye kesin değerlerin birer veri olarak girildiği birimdir. Havacılıkta genel olarak iki çeşit hava durumu tahmini kullanılır. İlki meydan rutin hava raporudur (Meteorological Terminal Air Report-METAR) ve kısa vadeli tahminleri gösterir. İkincisi ise meydan hava tahminidir (Terminal Aerodrome Forecast-TAF) ve orta/uzun süreli hava tahminleri gösterir. Her iki raporda da ilgili meydandaki rüzgâr hızı, bulut alt tavanı (bulutların yerden yüksekliği), görüş mesafesi (bulunulan yerden en uzak görülebilen noktanın mesafesi), sıcaklık ve hava hadiseleri (yağmur, kar, dolu vb.) gibi bilgileri içerir.



Şekil 4.3 Bulanık Kontrolcü

AHTP'de bulanık kontrol sisteminin girdisi olarak meteoroloji uzmanları tarafından üretilen TAF raporu kullanılır. Bu raporda tahmin edilen rüzgâr hızı, bulut alt tavanı ve görüş mesafesi, kontrol sisteminin kesin (Crisp) değerlerdir. Planlayıcı üslere ait bu meteorolojik verileri ilgilenilen zaman dilimi için alır ve bulanık kontrolcüye birer veri olarak girer.

4.3.2.2. Bulanıklaştırma birimi

Bulanıklaştırma birimi kesin değerleri, bulanık sözel değişkenlere dönüştüren birimdir. Bulanıklaştırma aşamasında rüzgâr, bulut alt tavanı ve görüş kesin değerleri bulanık sözel değişkenlere dönüştürülür. Her bir bulanık sözel değişkenin 5'er adet ayrı sözel terimi vardır. Görüş ve bulut alt tavanı için "*çok düşük*", "*düşük*", "normal", "*iyi*", "*çok iyi*" sözel terimleri, rüzgâr için "*çok sakin*", "*sakin*", "*orta*", "*şiddetli*", "*çok şiddetli*" sözel terimleri, risk için "*risksiz*", "*düşük riskli*", "*normal riskli*", "*riskli*", "*yüksek riskli*" sözel terimleri kullanılmıştır.

Sözel değişkenlere ait üyelik fonksiyonları çok uzmanlı direkt metot kullanılarak oluşturulmuştur (Klir ve Yuan, 1995). Çalışmada gönüllü olan toplamda 37 pilotun uzmanlıklarından faydalanılmıştır. Her sözel değişkene ait her sözel terimin olası bütün değerlerinin pilotlar tarafından değerlendirilmesi istenmiştir. Yapılan değerlendirmede sözel

terime ait bir değerin, bu sözel terime ait olup olmadığının cevaplanması istenmiştir. “100 m görüşü çok düşük görüş olarak değerlendirir misiniz?” sorusu, değerlendirmeye bir örnek olarak verilebilir. Pilotlardan alınacak cevapların kolay elde edilebilmesi ve daha sonra kolayca değerlendirilebilmesi için Çizelge 4.2’de verilen form tasarlanmıştır. Her sözel değişken için ayrı bir form oluşturulmuştur. Formda bir sözel değişkene ait sözel terimler ilk satıra ve sözel terimin alabileceği değerler ilk sütuna yerleştirilmiştir. Formlar, sözel terime ait değerlerin ilk satırdan son satıra kadar hangi sözel terime ait olduğu işaretlenerek doldurulur.

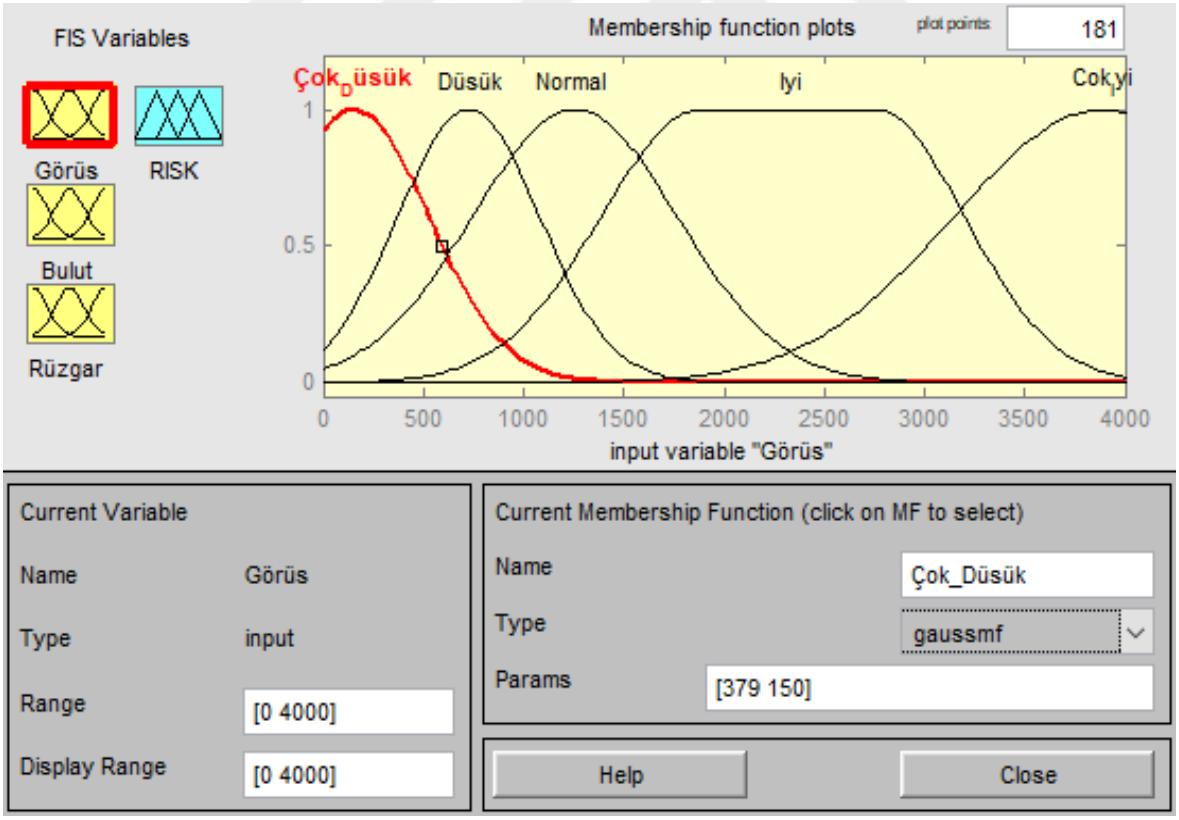
Formlar pilotlara verilmeden önce çalışmanın amacı ve formun nasıl doldurulması gerektiği anlatılmıştır. Daha sonra her bir sözel değişken için ilgili değerleri sözel terimlere göre değerlendirmesi istenmiştir. Her bir sözel değişken için 37 pilottan elde edilen verilerin toplanmış hali Ek Açıklama-A’da verilmiştir.

Çizelge 4.2 Örnek Meteorolojik Görüş Formu

METEOROLOJİK GÖRÜŞ FORMU						
	GÖRÜŞ (m)	ÇOK DÜŞÜK	DÜŞÜK	NORMAL	İYİ	ÇOK İYİ
1	100	X				
2	200	X				
3	300		X			
4	400		X			
5	500		X			
6	600			X		
7	700			X		
8	800			X		
9	900				X	
10	1000					X

Elde edilen veriler ile sözel değişkenlere ait üyelik değerleri denklem (4.1) kullanılarak hesaplanmıştır. Burada A bulanık küme, *A(x)* ise *x*'in *A* kümesine olan üyelik değeridir. "Sözel değişken değeri sözel terime ait midir?" sorusu "*i"* uzmanı tarafından evet olarak cevaplandı ise $a_i(x)=1$ diğer durumda $a_i(x)=0$'dır. "*n*" ise toplam uzman sayısıdır. Bu şekilde hesaplanan üyelik değerleri, MATLAB programının eğri uydurma aracına veri olarak girilmiştir. Üyelik değerlerini en iyi tanımlayan eğri fonksiyonu, üyelik fonksiyonu olarak elde edilmiştir (Chaturvedi, 2008). Görüş sözel değişkenine ait her bir sözel değişkene ait örnek üyelik fonksiyonları Şekil 4.4'de verilmiştir. Diğer sözel değişkenlere ait üyelik fonksiyonları ve MATLAB'de kullanılan bulanık kontrolcü ara yüzü Ek Açıklama-B'de sunulmuştur. Elde edilen üyelik fonksiyonları ile kesin olan meteorolojik tahmin değerleri birer bulanık değere dönüştürülmüştür.

$$A(x) = \frac{\sum_{i=1}^{n} a_i(x)}{n} \tag{4.1}$$



Şekil 4.4 Görüş İçin Üyelik Fonksiyonları

4.3.2.3. Kural tabanı

Kural Tabanı bulanık kontrolün kalbini oluşturur ve uzmanlar tarafından belirlenmiş, bulanık mantık denetleyicinin davranışlarını tespit eden denetim kurallarını içerir. Bulanık kontrol kuralları genellikle bir uzman bilgisinden üretilir. Bu kurallar sistemin giriş ve çıkışları arasındaki mantıksal ilişkileri açıklar.

Bir bulanık kontrol kuralı genellikle "Eğer-İse" formundadır ve birçok girişli tek çıkışlı (Many Input Single Output) sistem için bulanık kontrol kural tabanı aşağıdaki gibidir:

Kural 1. ***EĞER*** $X_1 = A_{11}$ ve $X_2 = A_{12}$ ……. ve $X_m = A_{1m}$ ***İSE*** $Y = B_1$

Kural 2. ***EĞER*** $X_1 = A_{21}$ ve $X_2 = A_{22}$ ……. ve $X_m = A_{2m}$ ***İSE*** $Y = B_2$

.
.
.

Kural n. ***EĞER*** $X_1 = A_{n1}$ ve $X_2 = A_{n2}$ ……. ve $X_m = A_{nm}$ ***İSE*** $Y = B_n$

Burada X_i sistem giriş değişkenlerine, Y çıkış değişkenine, A_{ij} ise bulanık kümelere (sözel değişkenlere) karşılık gelmektedir. Bulanık Eğer-İse kuralları, uzmanların ilgili alandaki bilgisini ifade ederken kullanılabilecek çok kullanışlı bir yöntemdir (Wang, 1994). Bu formdaki örnek bir kural aşağıdaki gibidir:

- Eğer Görüş ***düşük***, Bulut Alt Tavanı ***normal*** ve Rüzgâr ***sakin*** ise İniş/Kalkış ***risklidir***.

"*n*" giriş değişkeni sayısı, "*m*" giriş değişkenindeki sözel terim sayısı olmak üzere kural tabanındaki kuralların sayısı en fazla m^n tanedir. Bu çalışmada kural tabanının bütünselliğinin (Completeness) sağlanabilmesi için 5^3=125 tane kural tanımlanmıştır (Pedrycz, 1993). Kural tabanı, bir pilot olan yazar tarafından oluşturulmuştur. Oluşturulan kural tabanı Çizelge 4.3'de verilmiştir.

Çizelge 4.3 Kural Tabanı

GÖRÜŞ	BULUT ALT TAVANI	RÜZGÂR				
		ÇOK SAKİN	SAKİN	ORTA	ŞİDDETLİ	ÇOK ŞİDDETLİ
ÇOK DÜŞÜK	ÇOK DÜŞÜK	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
ÇOK DÜŞÜK	DÜŞÜK	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
ÇOK DÜŞÜK	NORMAL	RİSKLİ	RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
ÇOK DÜŞÜK	İYİ	RİSKLİ	RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
DÜŞÜK	ÇOK İYİ	RİSKLİ	RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
DÜŞÜK	ÇOK DÜŞÜK	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
DÜŞÜK	DÜŞÜK	RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
DÜŞÜK	NORMAL	RİSKLİ	RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
DÜŞÜK	İYİ	NORMAL RİSKLİ	NORMAL RİSKLİ	RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ
DÜŞÜK	ÇOK İYİ	NORMAL RİSKLİ	NORMAL RİSKLİ	RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ
NORMAL	ÇOK DÜŞÜK	RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
NORMAL	DÜŞÜK	NORMAL RİSKLİ	NORMAL RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
NORMAL	NORMAL	DÜŞÜK RİSKLİ	DÜŞÜK RİSKLİ	NORMAL RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ
NORMAL	İYİ	RİSKSİZ	DÜŞÜK RİSKLİ	NORMAL RİSKLİ	RİSKLİ	RİSKLİ
NORMAL	ÇOK İYİ	RİSKSİZ	DÜŞÜK RİSKLİ	NORMAL RİSKLİ	RİSKLİ	RİSKLİ
İYİ	ÇOK DÜŞÜK	RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
İYİ	DÜŞÜK	NORMAL RİSKLİ	NORMAL RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
İYİ	NORMAL	DÜŞÜK RİSKLİ	DÜŞÜK RİSKLİ	NORMAL RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ
İYİ	İYİ	RİSKSİZ	RİSKSİZ	DÜŞÜK RİSKLİ	NORMAL RİSKLİ	RİSKLİ
İYİ	ÇOK İYİ	RİSKSİZ	RİSKSİZ	DÜŞÜK RİSKLİ	NORMAL RİSKLİ	RİSKLİ
ÇOK İYİ	ÇOK DÜŞÜK	RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
ÇOK İYİ	DÜŞÜK	NORMAL RİSKLİ	NORMAL RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ	YÜKSEK RİSKLİ
ÇOK İYİ	NORMAL	DÜŞÜK RİSKLİ	DÜŞÜK RİSKLİ	NORMAL RİSKLİ	RİSKLİ	YÜKSEK RİSKLİ
ÇOK İYİ	İYİ	RİSKSİZ	RİSKSİZ	DÜŞÜK RİSKLİ	NORMAL RİSKLİ	RİSKLİ
ÇOK İYİ	ÇOK İYİ	RİSKSİZ	RİSKSİZ	DÜŞÜK RİSKLİ	NORMAL RİSKLİ	RİSKLİ

4.3.2.4. Sonuç çıkarım birimi

Bulanık çıkarımda, bulanık mantık işlemleriyle bulanık kural tabanındaki *EĞER-İSE* kuralları birleştirilerek, giriş ve çıkış bulanık uzayı arasında bir bağıntı oluşturulur. Kural tabanındaki girdilere göre koşul değerlendirmeleri yapılır. Bir kural için koşullar sağlanıyorsa o kural aktive edilir. Daha sonra aktive edilen tüm kurallar, bir operatör ile birleştirme işlemine tabi tutulur. Mamdani ve Sugeno olmak üzere iki tip sonuç çıkarım birimi literatürde çok yaygın olarak kullanılmaktadır. Bu çalışmada Mamdani sonuç çıkarım birimi, "*ve*" operatörü için "*prod*", "*veya*" operatörü için "*max"*, "*birleştirme*" operatörü için "*sum"* metotları kullanılmıştır.

4.3.2.5. Durulaştırma birimi

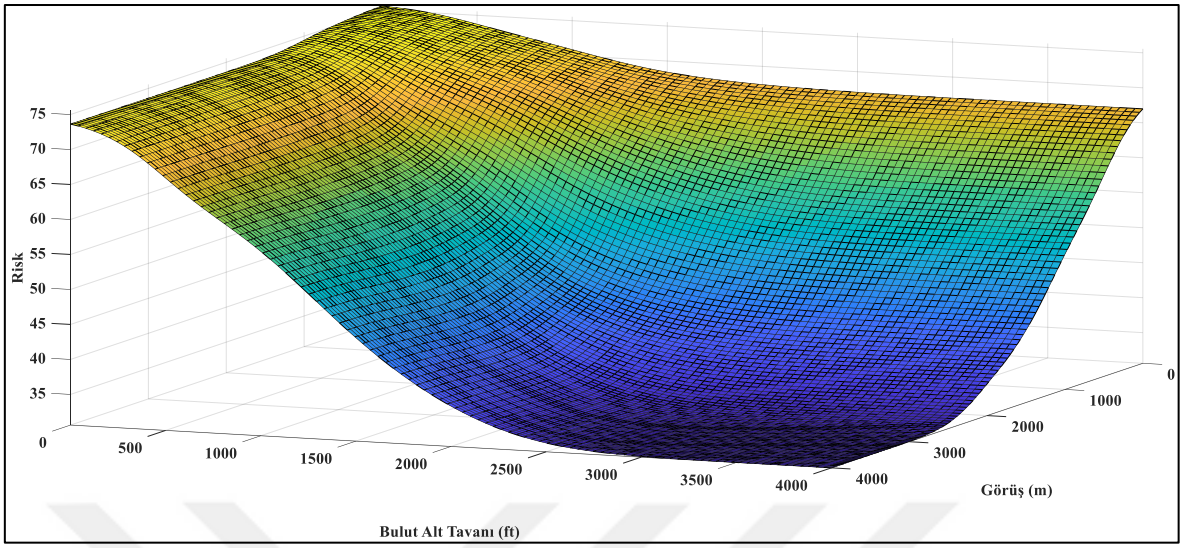
Durulaştırma işlemiyle çıkarım biriminden alınan bulanık değerler gerçek sayı değerlerine dönüştürülür. Literatürde yaygın olarak kullanılan üç çeşit durulaştırma yöntemi şunlardır:

- Ağırlık merkezi yöntemi
- Toplam alan merkezi yöntemi
- Maksimumların en büyüğü yöntemi

Bu çalışmada, durulaştırma yöntemi olarak ağırlık merkezi yöntemi kullanılmıştır. Bu yöntem model için daha uygundur. Çünkü girdilerdeki küçük değişiklikler çıktıda da küçük değişikliğe neden olmaktadır (Devamlılık (Continuity) özelliği). Bununla birlikte kontrol yüzeyinin ya da çıktı grafiğinin eğimi düşük kalmaktadır. Benzer şekilde düşük eğimde de kontrol çıktılarının değişimleri sıçramalar göstermez (Hellendoorn ve Thomas, 1993).

4.3.2.6. Bulanık Kontrol Çıktısı

Elde edilen çıktılar, üslerdeki hava koşullarından dolayı oluşan o üsse ait riskleri göstermektedir. Bu değerler uçakların bir üsse iniş kalkış yaparken maruz kalacağı riskleri belirli bir değer altında tutmak için kullanılmıştır. Bir üsse ait hava koşulu riskinin elde edilmesi için geliştirilen bulanık kontrolcüye ait örnek bir çıktı Şekil 4.5 gösterilmiştir.



Şekil 4.5 Bulanık Kontrolcüye Ait Örnek Bir Çıktı

Bulanık kontrolcüye ait risk grafiği sabit bir rüzgâr değeri (20K) üzerinden hesaplanmıştır. Rüzgâr değeri sabit iken görüş ve bulut alt tavanı değişkenlerine göre elde edilen risk değerini göstermektedir. Beklenildiği üzere görüş veya bulut alt tavanı küçüldükçe risk değeri büyümektedir.

5. MATEMATİKSEL MODEL

AHTP için önerilen karma tam sayılı matematiksel model aşağıda verilmiştir. Bu modelde bir uçak bir üsse bir kez uğrar yani alt turlara izin verilmemiştir. Model görece küçük boyutlu örnekleri çözebilmektedir. Modeldeki kısıt ve değişken sayısı, çok çabuk mevcut bilgisayar işlem gücü ile çözülemeyecek boyutlara ulaşmaktadır. Matematiksel modele ait IBM ILOG CPLEX kodu ve örnek bir probleme ait data (veri) dosyası Ek Açıklamalar -C'de verilmiştir.

5.1. Notasyon

Kümeler:

$\boldsymbol{V}$ → Üsler kümesi,

$\boldsymbol{K}$ → Uçaklar kümesi,

$\boldsymbol{P_{A_k}}$ → Pisti hasarlı olan ana üsler kümesi,

$\boldsymbol{P}$ → Pisti hasarlı olan üsler kümesi (ana üsler hariç),

$\boldsymbol{AU}$ → Bu üsse gelirken veya bu üsten giderken alçak uçulması gereken üsler kümesi.

Parametreler:

$\boldsymbol{N}$ → Toplam üs sayısı,

$\boldsymbol{i, j} \in V$ → Üsler,

$\boldsymbol{k} \in K$ → Uçaklar,

$\boldsymbol{D_{ij}}$ → Üsler arasındaki mesafe matrisi (Deniz mili-NM),

$\boldsymbol{C_k}$ → *k* tipindeki uçağın birim zamandaki uçuş maliyeti,

$\boldsymbol{A_k}$ → *k* uçağının ana üssü,

$\boldsymbol{s_{ij}}$ → *j* üssünün *i* üssünden talep miktarı (kg),

$\boldsymbol{K_k}$ → *k* tipindeki uçağın kapasitesi (kg),

$\boldsymbol{B}$ → Bir uçağın dolaşabileceği maksimum ayrıt sayısı (maksimum bacak sayısı),

$\boldsymbol{H_k}$ → *k* uçağının seyir sürati (Knot),

$\boldsymbol{G}$ → Maksimum görev süresi (saat),

$\boldsymbol{T_k}$ → *k* uçağının menzili (NM),

$\boldsymbol{UR_i}$ → *i* üssünün hava koşulları risk değeri,

$\boldsymbol{HS_{ij}}$ → Her bir ayrıtın orta noktasının hava savunma sistemlerine olan uzaklığının ortalaması (Deniz mili-NM),

$\boldsymbol{S_1}$ → Bir üsten diğer bir üsse gidebilmek için kaç ara durağa (üsse) uğranabileceğini gösterir ve *N–2* ile hesaplanır,

$\boldsymbol{S_2}$ → Bir üsten diğer bir üsse gidebilmek için uğranabilecek ara duraklara (üslere) ait kaç sıralı kombinasyon olabileceğini gösterir,

$\boldsymbol{asd_{k<i-j>S_1S_2}}$ → **Alt Sıra Dizini:** *k* aracının *i* üssünden *j* üssüne olan talebi taşırken gidebileceği rotalara ait kombinasyonları S_1 ve S_2 sayılarını gözeterek tanımlar. Asd minimum 2 kenardan (1'den 2'ye gidebilmek için 1-4-2 rotasının izlenmesi gibi), maksimumda n-1 kenardan (5 üslü bir problemde 1'den 2'ye gidebilmek için 1-3-4-5-2 rotasının izlenmesi gibi) oluşur. Bu dizide araca ait ana üs sadece rotanın başında veya sonunda olabilir (5 üslü problemde 1 üssü araca ait ana üs ise 2'den 4'e gidebilmek için maksimum n–2 kenar kullanılabilir 2-3-5-4 gibi.). Eleman sayısı (5.1) denklemi ile bulunur,

$$k \times \begin{pmatrix} 2 \times (N-1)\left({}_{(N-2)}P_1 + {}_{(N-2)}P_2 + \cdots + {}_{(N-2)}P_{(N-2)}\right) \\ +(N-2)(N-1)\left({}_{(N-3)}P_1 + {}_{(N-3)}P_2 + \cdots + {}_{(N-3)}P_{(N-3)}\right) \end{pmatrix} \quad (5.1)$$

Karar Değişkenleri:

$\boldsymbol{x_{ijk}} \in (0,1)$ → *i* üssünden *j* üssüne *k* uçağı gidiyorsa 1, aksi durumda 0,

$\boldsymbol{y_{ijk}}$ → *i* üssünden *j* üssüne giden *k* uçağı üzerindeki toplam yük miktarı (kg),

$\boldsymbol{u_{ijk}}$ → *i* üssünden *j* üssüne giden talebin *k* uçağı ile taşınan miktarı (kg),

$\boldsymbol{a_{ik}}$ → Alt tur engelleme kısıtına ait değişkenler,

Risk Faktörleri:

$\boldsymbol{F_1}$ → Maksimum kabul edilebilir hava koşulları riski,

$\boldsymbol{F_2}$ → Hasarlı pistten dolayı bir üsse iniş ya da kalkış yapılamayacak süre,

$\boldsymbol{F_3}$ → Aynı ayrıttan maksimum geçiş sayısı,

$\boldsymbol{F_4}$ → Hava savunma sistemlerine olan maksimum uzaklık,

$\boldsymbol{F_5}$ → Maksimum alçak irtifa uçuş süresi,

5.2. Matematiksel Model

5.2.1. Amaç fonksiyonu

Amaç fonksiyonu (5.2) toplam maliyeti minimize eder. Buradaki maliyet, *k* uçağının *i* üssünden *j* üssüne gitmesinden oluşan maliyettir. Her uçak için birim zaman maliyeti farklıdır. *i* üssünden *j* üssüne *k* uçağının gitmesinin maliyeti, *k* uçağının *i-j* üsleri arasındaki mesafeyi kat etme süresinin, *k* uçağının birim zamandaki maliyeti ile çarpılması sonucu bulunur. Sonuç olarak amaç fonksiyonu hem kullanılan uçak sayısını hem de kat edilen toplam mesafeyi minimize eder.

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} \frac{D_{ij}}{H_k} C_k x_{ijk} \tag{5.2}$$

5.2.2. Kısıtlar

$$\sum_{i=A_k} \sum_{j \in V} x_{ijk} \leq 1 \quad \forall\, k \in K \tag{5.3}$$

$$\sum_{j \in V} \sum_{i \in A_k} x_{jik} \leq 1 \quad \forall\, k \in K \tag{5.4}$$

$$\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0 \quad \forall\ i \in V,\ k \in K \tag{5.5}$$

$$\sum_{k \in K} u_{ijk} = s_{ij} \quad \forall\ i,j \in V \tag{5.6}$$

$$y_{ijk} \leq K_k \quad \forall\ i,j \in V,\ k \in K \tag{5.7}$$

$$\sum_{j \in V} (y_{jik} - u_{jik}) = \sum_{j \in V} (y_{ijk} - u_{ijk}) \quad \forall\ i \in (V \backslash A_k),\ k \in K \tag{5.8}$$

$$\sum_{j \in V} (y_{ijk} - u_{ijk}) = \sum_{j \in V} (y_{jik} - u_{jik}) \quad \forall\ i \in A_k,\ k \in K \tag{5.9}$$

$$M \times x_{ijk} \geq y_{ijk} \quad \forall\ i,j \in V,\ k \in K \tag{5.10}$$

$$a_{ik} - a_{jk} + N \times x_{ijk} \leq N - 1 \quad \forall\ i,j \in (V \backslash A_k),\ k \in K \tag{5.11}$$

$$(u_{ijk} \neq 0) \geq (x_{ijk} = 1) + \left(\sum_{\substack{r \in S_1 \\ t \in S_2}} \left(\sum_{l,m \in asd_{k,i,j,r,t}} x_{lmk} = r + 1 \right) = 1 \right) \quad (5.12)$$

$$\forall\ i,j \in V, k \in K$$

$$\sum_{i \in V} \sum_{j \in V} x_{ijk} \ \leq B \qquad \forall\, k \in K \quad (5.13)$$

$$\sum_{i \in V} \sum_{j \in V} x_{ijk} \times \left(\frac{D_{ij}}{H_k} + \frac{1}{4} \right) + \frac{1}{2} \leq G \qquad \forall\, k \in K \quad (5.14)$$

$$x_{ijk} \times D_{ij} \leq T_k \qquad \forall\ i,j \in V,\ k \in K \quad (5.15)$$

$$\sum_{i \in V} \sum_{j \in V} (x_{ijk} \times UR_i) - F_1 \times \sum_{i \in V} \sum_{j \in V} x_{ijk} \leq\ 0 \quad \forall\ \ k \in K \quad (5.16)$$

$$\sum_{i \in P_{A_k}} \sum_{j \in V} \sum_{k \in K} x_{ijk} = 0 \quad (5.17)$$

$$x_{ijk} = \begin{cases} = 0, & \frac{D_{A_k i}}{H_k} \leq\ F_2 \\ \geq 0, & d.d. \end{cases} \qquad \forall\, i \in\ P, k\ \in K \quad (5.18)$$

$$\sum_{k \in K} (x_{ijk} + x_{jik}) \leq F_3 \qquad \forall\ i,j \in V \quad (5.19)$$

$$\sum_{i \in V} \sum_{j \in V} (x_{ijk} \times HS_{ij}) - F_4 \times \sum_{i \in V} \sum_{j \in V} x_{ijk} \ \leq 0 \qquad \forall\, k\ \in K \quad (5.20)$$

$$\sum_{i \in V \setminus AU} \sum_{j \in AU} (x_{ijk} * \frac{D_{ij}}{H_k} + x_{jik} * \frac{D_{ji}}{H_k}) + \sum_{i,j \in AU} x_{jik} * \frac{D_{ji}}{H_k} \leq F_5 \quad \forall\, k \in K \quad (5.21)$$

$$x_{ijk}\ \in \{0,1\} \quad \forall\ i,j \in V,\ k \in K \quad (5.22)$$

$$u_{ijk}\ \geq 0 \quad \forall\ i,j \in V,\ k \in K \quad (5.23)$$

$$y_{ijk}\ \geq 0 \quad \forall\ i,j \in V,\ k \in K \quad (5.24)$$

$$a_{ik}\ \geq 0 \quad \forall\ i \in V,\ k \in K \quad (5.25)$$

Eşitlik (5.3) ve (5.4) uçakların kendi üslerinden kalkıp geri kendi üslerine inmesini sağlar. Eşitlik (5.5) bir üsse inen uçağın o üsten kalmasını sağlar. Eşitlik (5.6) taleplerin karşılanmasını sağlar. Eşitlik (5.7) uçak kapasitesinin aşılmamasını sağlar. Eşitlik (5.8) ve (5.9) bir üsse giren ve çıkan; ana üsse giren ve ana üsten çıkan ürün miktarlarının eşit olmasını sağlar. Eşitlik (5.10) x ve y değişkenleri arasındaki bağlantı kısıtıdır. Eşitlik (5.11) alt tur engelleme kısıtıdır. Eşitlik (5.12) en genel tanımıyla üs taleplerinin uygun bir rotayla

karşılanmasını sağlar. Bu gösterim, geleneksel bir gösterim olmayıp CPLEX optimizasyon yazılımından uyarlanarak alınmıştır. Bu eşitlik, *i* ve *j* üssü arasında bir taşıma olduğunda ($u_{ijk}\neq 0$), *k* uçağının ara bir durağa uğramadan *i*'den *j*'ye direk gittiğini ($x_{ijk}=1$) veya diğer bütün olası ara duraklı taşımalardan yalnızca birisinin kullanılmasını $\left(\sum_{\substack{r \in S_1 \\ t \in S_2}} \left(\sum_{l,m \in asd_{k,i,j,r,t}} x_{lmk} = r+1\right) = 1\right)$ sağlar. Eşitlik (5.13) bir pilotun uçtuğu bacak sayısının maksimum uçuş bacağı sayı *B*'den az olmasını sağlar. Eşitlik (5.14) toplam görev süresinin maksimum görev süresi G'den az olmasını sağlar. Eşitlik (5.15) uçağın menzilinin geçilmemesini sağlar. Eşitlik (5.16) bir uçağın rotası boyunca maruz kalacağı ortalama hava koşulları riskinin F_1 seviyesinin altında kalmasını sağlar. Eşitlik (5.17) ve (5.18) pisti hasarlı olan ana üsten kalkış olmamasını veya hasarlı pisti olan bir üsse F_2 süresi kadar iniş yapılmamasını sağlar. Eşitlik (5.19) bir ayrıtın maksimum geçiş sayısı olan F_3'ten fazla kullanılmamasını sağlar. Eşitlik (5.20) bir rotanın hava savunma sistemlerinden ortalama uzaklığının F_4'den küçük olmasını sağlar. Eşitlik (5.21) bir rotadaki toplam alçak irtifa uçuş süresinin F_5'den küçük olmasını sağlar. Eşitlik (5.22), (5.23), (5.24) ve (5.25) değişken ve işaret (negatif/pozitif) tanımlarını yapar.

5.3. Matematiksel Model ile Örnek Bir Problemin Çözümü

Önerilen matematiksel modelin boyutu üs sayısı arttıkça büyümektedir. Küçük boyutlu jenerik bir problem üzerinde matematiksel modelin işleyişi ve elde ettiği çözümler anlatılmıştır. Örnek problem beş üsten oluşmaktadır. Talep matrisi Çizelge 5.1'de, üslere ve hava savunma sistemlerine ait konumlar ve her üsse ait hava koşulları riskleri Çizelge 5.2'de ve problemde kullanılan uçaklara ait özellikler Çizelge 5.3'de verilmiştir.

Çizelge 5.1 Örnek Probleme Ait Talep Matrisi

Üsler	1	2	3	4	5
1	0	100	30	400	200
2	200	0	100	0	100
3	100	50	0	50	2000
4	150	50	0	0	100
5	0	100	0	0	0

Çizelge 5.2 Örnek Probleme Ait Üs ve Hava Savunma Sistemleri Konumları ve Bilgileri

Üs No	x Koordinatı	y Koordinatı	Hava Koşulları Riski
1	178	187	10
2	20	243	70
3	485	204	20
4	14	326	60
5	3	97	50
Hava Savunma Sistemi			
1	372	117	
2	437	2	
3	476	232	
4	121	267	
5	289	65	

Çizelge 5.3 Örnek Probleme Ait Uçak Bilgileri

Uçak No	Ana Üs	Kapasite	Maliyet	Hız	Menzil
1	1	2500	6000	300	2160
2	2	3000	5000	250	2000

Örnek problemin matematiksel model ile çözümü sonucu elde edilen optimum çözüm 46567,133 birim olarak bulunmuştur. Optimum çözümde bir üsse ait talep iki uçakla taşınabilmiştir. Taleplerin taşınmasında birden fazla ara noktaya iniş yapılmasına müsaade edilmiştir. Optimum çözüm sonucunda oluşan uçak rotaları Çizelge 5.4'de verilmiş olup, *1–numaralı* üsten *5–numaralı* üsse gidecek olan bir talep rota başında uçağa yüklenmiştir. Bu yük uçak *5–numaralı* üsse varıncaya kadar üç ara durağa inmiştir. Bir uçağın bir üsten kalkıp diğer bir üsse giderken üzerinde taşıdığı toplam yük miktarı Çizelge 5.5'te ve bir üsse ait talebin ne kadarının hangi uçak tarafından taşındığı bilgisi ise Çizelge 5.6'da verilmiştir. Görüldüğü üzere matematiksel model talep bölünmesine, taleplerin taşınmasında birden fazla ara durağa inilmesine ve bütün üslerin birbiri arasında bir talebi bulunmasına imkân sağlamaktadır.

Çizelge 5.4 Optimum Çözüme Ait Uçak Rotaları

1. Uçak	1	3	4	2	5	1
2. Uçak	2	3	5	2		

Çizelge 5.5 Uçakların Uçtuğu Bacaktaki Taşıdığı Toplam Yük Miktarı

Uçağın Kalkış Üssü	Uçağın İniş Üssü	1. Uçağın Taşıdığı Toplam Miktar	2. Uçağın Taşıdığı Toplam Miktar
1	3	730	0
2	3	0	100
2	5	1450	0
3	4	1500	0
3	5	0	1400
4	2	1350	0
5	1	450	0
5	2	0	100

Çizelge 5.6 Uçaklar Tarafından Taşınan Taleplerin Miktarları

Talebin Çıkış Üssü	Talebin Varış Üssü	1. Uçağın Taşıdığı Miktar	2. Uçağın Taşıdığı Miktar
1	2	100	0
1	3	30	0
1	4	400	0
1	5	200	0
2	1	200	0
2	3	0	100
2	5	100	0
3	1	100	0
3	2	50	0
3	4	50	0
3	5	600	1400
4	1	150	0
4	2	50	0
4	5	100	0
5	2	0	100

Önerilen matematiksel model gerçek hayatta karşılaşılabilecek problem boyutlarını mevcut bilgisayar işlem gücüyle makul bir zamanda çözememektedir. Bu nedenle AHTP'nin çözümü için meta sezgisel bir yöntem olan genetik algoritmaya ait iki farklı varyant olan hücresel genetik algoritma ve klasik genetik algoritma önerilmiştir.

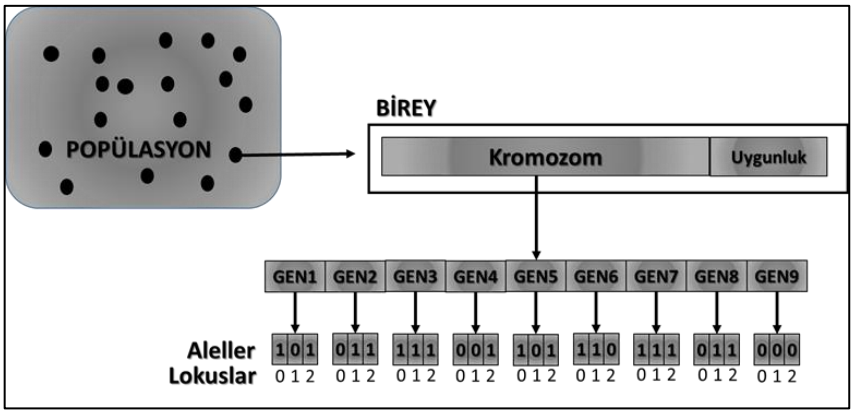
6. MATERYAL VE YÖNTEM

AHTP'nin çözümü için tasarlanan matematiksel model gerçek hayat problemlerinin çözümü için yetersiz kalmaktadır. Bu nedenle AHTP'nin çözümü için genetik algoritmanın iki varyantı olan klasik genetik algoritma ve hücresel genetik algoritma önerilmiştir. Bu bölümde çözüm yöntemi olarak önerilen Klasik Genetik Algoritma (kGA) ve hücresel genetik algoritma açıklanmıştır. kGA genel olarak tanıtılmış ve algoritma adımları gösterilmiştir. hGA'nın tanımı yapılmış ve çeşitlerinden bahsedilmiştir. hGA'ya ait adımlar anlatılmış ve AHTP'nin çözümü için kullanılan uyum fonksiyonu, genetik gösterim, çaprazlama ve mutasyon operatörleri detaylarıyla anlatılmıştır. Daha sonra parametre analizi yapılarak parametrelere ait uygun değerler bulunmuştur.

6.1. Klasik Genetik Algoritma

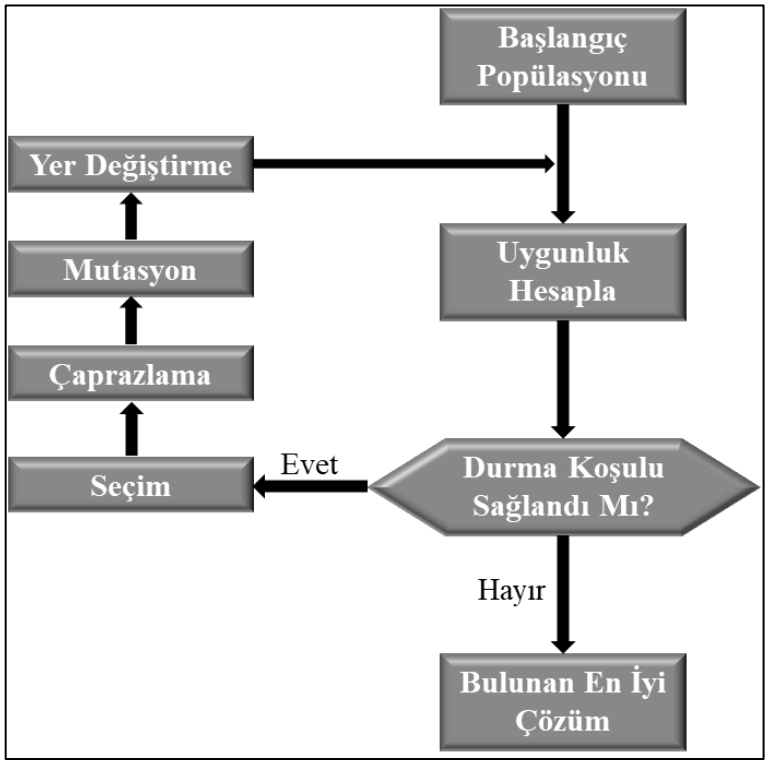
Klasik Genetik Algoritma doğada olan evrimsel süreci taklit ederek çalışan bir optimizasyon yöntemidir. Çözüm uzayında bulunan çözümleri bir birey olarak ele alır ve "en iyi olan hayatta kalır" evrimsel olgusuna göre iyi olan çözümleri çözüm kümesinde tutarak bütünsel en iyi çözümü arar. Genetik algoritmanın ilk bilimsel örneklerinden biri, John Holland'ın, Darwin teorisini bilgisayar ortamında taklit etmeyi amaçlaması sonucunda ortaya çıkmıştır (Holland, 1975).

Klasik Genetik Algoritma tek bir çözümden ziyade birbirinden tamamen bağımsız farklı çözümlerden oluşan bir çözümler grubu kullanarak çalışır. Bu grup, farklı bireylerden oluşan bir topluma benzetilebilir. Toplumdaki her bir birey bir çözümü temsil eder. Bireylerin çiftleşerek yeni bireyler meydana getirmesi ve daha iyi özellikleri olan bireylerin doğada hayatta kalma olasılığının daha yüksek olması olgusunu bilgisayar ortamında taklit eder. Çözümlerden oluşan topluluğa popülasyon, popülasyon içerisindeki her bir çözüme kromozom, kromozomların ölçülen değerine uyum değeri, bir çözümü oluşturan değişkenlere gen, genlerin kromozom içindeki konumlarına lokus, genlerde saklanan değerlere alel denilmektedir. Bu ifadelerin temsili gösterimi Şekil 6.1'de verilmiştir (Alba ve Dorronsoro, 2008).



Şekil 6.1 Bir Kromozom Gösterimi

Genetik Algoritmanın genel akış şeması Şekil 6.2'de verilmiştir. İlk üretilen çözümleri içeren gruba başlangıç popülasyonu, popülasyon içinden yeni bireyler oluşturmak için iki bireyin seçilmesine seçim, iki bireyin bir araya gelerek yeni bir birey üretmesine çaprazlama, üretilen bireylerin bir başkalaşım geçirerek yeni bir birey üretmesine mutasyon, üretilen iyi bireylerin bir sonraki jenerasyonun oluşturulmasında yer almasına yer değiştirme, algoritmanın belirli bir koşulu sağladığında sonlandırılmasına durma koşulu denir (Melanie, 1999).

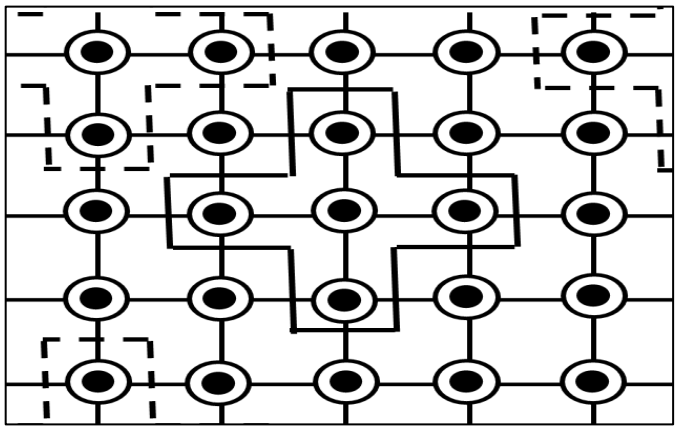


Şekil 6.2 Genetik Algoritmanın Genel Akış Şeması

AHTP'nin çözümü için MATLAB'de bir klasik genetik algoritma kodu oluşturulmuştur. Bu kodun oluşturulmasında hücresel genetik algoritma için tasarlanan kod kullanılmıştır. Bu kodlardaki farklılık iki algoritma arasındaki farktan kaynaklanmaktadır. kGA ile hGA arasındaki ana farklar popülasyon yapısında ve çaprazlamada kullanılacak bireylerin seçimlerinde bulunmaktadır. kGA'daki popülasyonda bir komşuluk yapısı yoktur. Ebeveyn olarak seçilecek bireyler, popülasyon içerisinden rulet tekerleği yöntemi ile belirlenir. Deney setleri kGA ile çözülmüş ve elde edilen sonuçlar hGA ile karşılaştırılmıştır.

6.2. Hücresel Genetik Algoritma

Hücresel Genetik Algoritma (hGA), klasik Genetik Algoritmanın bir alt kümesidir. Evrim sürecini bir bireyin bakış açısından ele alır. Toplumda bireyler yakın çevresindeki kişiler ile etkileşim halindedir ve bu bireyler ile bir araya gelip yeni bireyler meydana getirirler. Bireylerin potansiyel eşleri olabilecek bireyler o bireyin komşuları olarak adlandırılır. Bir bireye komşu bireylerin belirlenmesinde bireylerin birbirine olan uzaklıkları rol oynar. Komşular, bireye belirli bir uzaklıktan daha yakın bireyler arasından belirlenir. Bu şekilde belirlenen komşuluk yapısına bir örnek Şekil 6.3'de verilmiştir (Dorronsoro ve Alba, 2006).

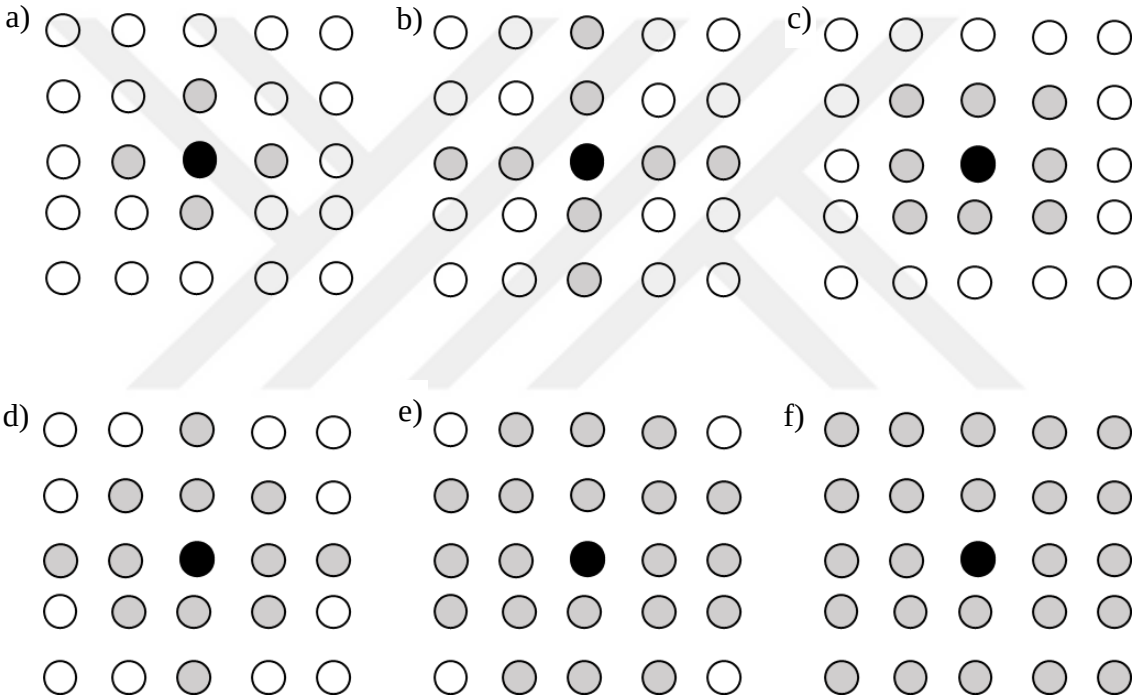


Şekil 6.3 hGA 'da Komşuluk Örneği

hGA'da görece küçük bir komşuluk kullanılır ve bireyler ancak bu komşulukları içerisindeki bireylerle etkileşime geçebilir. Üst üste geçen bu küçük komşuluklar çözüm uzayının daha iyi aranmasına yardımcı olur. Küçük komşuluklar içerisinde çözümlerin yavaş

geçişi popülasyonun daha iyi keşfedilmesini, komşuluk içindeki genetik operasyonlar aramanın bu bölgede yoğunlaşmasını sağlar.

Popülasyonun tarif edildiği serim üzerindeki belirli bir noktanın komşuları, popülasyon içindeki bu bireyden diğer bireylere olan Manhattan uzaklığı ile tanımlanır. Serim üzerindeki her bir noktanın komşulukları diğer noktaların komşulukları ile iç içe geçer ve her bir komşuluk aynı boyuta ve şekle sahiptir. hGA ‘da en çok kullanılan komşuluklara ait örnekler Şekil 6.4’te verilmiştir (Alba ve Dorronsoro, 2008). Bu çalışmada Şekil 6.4 a)’da seçeneğinde verilen Von Neumann komşuluk tipi kullanılmıştır.



Şekil 6.4 hGA’ya Ait Komşuluk Örnekleri

Üreme döngüsünün bireylere nasıl uygulanacağına göre belirlenen iki farklı hGA türü vardır. İlkinde döngü bireylere aynı anda uygulanır ve bir sonraki jenerasyona ait popülasyonda aynı anda yaratılır. Bu prosedüre senkronize hGA denir. Diğerinde bireyler belirli bir sıra ile güncellenir ve bu prosedüre sıralı hGA denir (Alba vd. , 2005). hGA bireylerin güncelleme prosedürüne göre farklı özellikler gösterir. Schönfisch ve de Roos (1999) yaptıkları çalışmada senkronize ve sıralı güncelleme prosedürlerinin etkilerini incelemişler ve her iki prosedürün niteliksel olarak farklı dinamiklere yol açtığını tespit etmişlerdir.

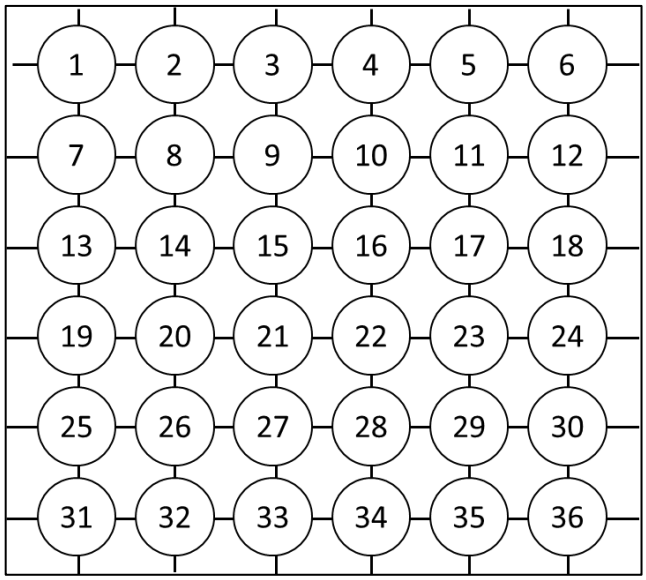
6.2.1. Senkronize (Paralel) hücresel genetik algoritma

Popülasyondaki bütün bireyler ile çaprazlama gerçekleştirildikten sonra üretilen çocuklar ile popülasyondaki kötü bireyler aynı anda yer değiştirilir ise algoritma senkronize genetik algoritma olur. Burada sonraki jenerasyona ait bireylerin tamamı eş zamanlı olarak yaratılır (Schönfisch ve de Roos, 1999).

6.2.2. Sıralı hücresel genetik algoritma

Sıralı hücresel genetik algoritmada, üreme döngüsü ile yer değiştirme işlemleri belirli bir sırayla uygulanır. Burada üreme döngüsü ve bir sonraki popülasyonun yaratılması aynı anda ve sırayla gerçekleşir. Sıralı hGA'da sık kullanılan dört yer değiştirme prosedürleri aşağıda verilmiştir.

- **Satır Süpürme:** Popülasyondaki bireyler satır satır sıra ile güncellenir. Satır süpürme prosedürüne bir örnek Şekil 6.5'de verilmiştir. Buradaki hücrelere ait numaralar bireylerin güncellenme sırasını göstermektedir.



Şekil 6.5 Sıralı Süpürme

- **Sabit Rastgele Süpürme:** Yer değiştirme uygulanacak bir sonraki hücre, yerine koymadan düzgün dağılıma göre seçilir ve bu seçim $(c_1^j, c_2^k, \ldots, c_n^m)$ şeklinde güncelleştirme sırası verir. c_q^p teriminde, p hücre numarasını, q güncelleme zamanını ve (j, k, …, m) ise n hücrenin bir permütasyonunu gösterir. Elde edilen bu permütasyon bütün jenerasyonlar için kullanılır.
- **Yeni Rastgele Süpürme:** Her jenerasyon sonundaki güncelleme işleminden sonra yeni bir rastgele permütasyonun kullanılması dışında sabit rastgele süpürme gibi çalışır.
- **Düzgün Dağılım Süpürme:** Güncellenecek hücre yerine koyarak düzgün dağılıma göre belirlenir.

6.2.3. Hücresel genetik algoritmanın adımları

Hücresel genetik algoritmanın içerisinde kullanılan adımlar en genel haliyle kısaca aşağıda açıklanmıştır.

- **İlk Popülasyon:** Probleme ait rastgele üretilen bireylerden oluşturulan ilk popülasyondur. Bu adımda problem bilgisi veya diğer bilgiler, bireylerin üretilmesinde kullanılabilir.

- **Uyum Fonksiyonu:** İlk popülasyona ait bireyler üretildikten sonra bu bireylerin eldeki problem için ne kadar iyi bir çözüm olduğu, uyum değeri hesaplanarak bulunur.

- **Seçim:** Popülasyon içerisinde yer alan daha iyi uyum değerine sahip olan bireylerin ebeveyn olarak seçilmesi ve yeni bireylerin üretilmesinde kullanılmasıdır. Bu sayede "daha iyi olan hayatta kalır" evrim kanununun uygulanması sağlanır. Jebari ve Madiafi (2013) yaptıkları çalışmada literatürde çok kullanılan altı seçim yöntemini, bunların çözüm kalitesi ve genetik farklılık bakımından genetik algoritma performansına olan etkilerini incelemişlerdir.

- **Çaprazlama:** Ebeveynlerin iki veya daha fazla parçasını alır ve bunları yeni bireylerin (çocuk çözümler) üretilmesinde kullanır. Algoritmanın performansını etkileyen en önemli faktörlerden birisi de çaprazlamadır (Błażej vd. , 2016). İdeal durumda çocuk çözümler ebeveynleri ile aynı değildir fakat ebeveynlerine ait iyi özellikleri taşır.

- **Mutasyon:** Bir birey üzerinde rastgele değişiklikler yaratılmasıdır. Bu değişiklikler doğada mutasyona uğramış canlıları göz önüne alır ve çözüm uzayındaki aramayı çeşitlendirmeyi amaçlar.

- **Yer değiştirme:** Çaprazlama ve mutasyona tabi tutulan bireylerden türetilen yeni bireyler (çocuk çözümler), belirli kıstaslara göre popülasyondaki ebeveynlerinin yerini alır ve bir sonraki jenerasyonu oluşturur. Bu şekilde popülasyondaki iyi bireylerin popülasyon içerisinde tutulması sayesinde genetik evrim modellenmiş olur. Genellikle en iyi uyum değerine sahip olan birey tüm jenerasyonlar boyunca toplumda tutulur. Bu durum elitizm olarak adlandırılır (Chambers, 2001).

6.3. AHTP için Önerilen Hücresel Genetik Algoritma

Hava taşımaları probleminin ancak çok küçük boyutlu ve bazı kısıtlarının gevşetilmiş hali matematiksel modelleme kullanılarak çözülebilmiştir. Bu tarz NP–Zor problemlerin büyük boyutlu örnekleri sadece farklı optimizasyon teknikleri kullanılarak çözülebilmektedir. Bu tez çalışmasında askeri hava taşımaları probleminin çözümü için bir hücresel genetik algoritma tasarlanmış ve MATLAB yazılımı kullanılarak kodlanmıştır.

Genetik algoritma kesin yöntemler gibi optimal çözümü garanti edemez. Fakat kısa zamanda oldukça iyi çözümler bulabilir. Özellikle problem boyutları büyüdükçe GA daha efektif hale gelmekte ve kısa zamanda bulduğu çözümler diğer yöntemlerin çözümleri ile kıyaslandığında daha iyi bir algoritma olduğu gözlenmektedir (Karakatič ve Podgorelec, 2015).

hGA'nın çözüm uzayını araştırma kapasitesi kGA 'ya göre daha yüksektir. Bu özellik, gerçek ve karmaşık problemler için tercih nedenidir (Alba ve Dorronsoro, 2008). Alba ve Dorronsoro (2006) çalışmasında hGA kullanarak, dokuz adet kapasiteli araç

rotalama problemi için bilinen en iyi çözümleri bulmuştur. Yukarıda bahsedilen nedenlerden dolayı hava taşımaları probleminin çözümü için kGA ve hGA kullanılmıştır. Önerilen hGA ve kGA için yazılan MATLAB kodunun tamamı http://dx.doi.org/10.17632/ddx6gjy7fy.1 internet adresinde örnek kısımları ise Ek Açıklamalar-D'de verilmiştir (Aksoy ve Kapanoğlu, 2018a). AHTP'nin çözümü için önerilen hücresel genetik algoritmanın tasarım aşamaları ve yapılan uyarlamalar aşağıdaki kısımlarda açıklanmıştır.

6.3.1. Genetik gösterim (kodlama)

Askeri hava taşımaları probleminde, birbirinden farklı uçak tipleri vardır. Her uçak tipi birbirinden farklı özelliklere sahiptir. Problemdeki talepler ise diğer araç rotalama problemlerinden farklılık göstermektedir. Her üs diğer bir üsse malzeme gönderebilir veya bir üs diğer üslerden malzeme talep edebilir. Dolayısı ile her bir üs hem talep noktası hem de depo olabilir. Bir çözümün temsil edilmesinde, farklı uçak tipi ve farklı talep yapısı, her bir uçağın bir üsten diğer üsse olan talebin ne kadarını taşıdığı bilgisini içermesini zorunlu kılmıştır. Ayrıca bir çözüm, rotada üslerin hangi sıra ile izlenmesi gerektiğini de göstermelidir. Bu yüzden problemin kodlanması için birbirine bağlı iki ayrı matris kullanılmıştır.

İlk matris rota matrisi, ikinci matris taşıma matrisidir. İlk matrisle rotalar belirlenmiş diğer matris ile de bu rota boyunca yapılan yüklemelere/boşaltmalara ait bilgiler saklanmıştır. Rota matrisinde, bir rotada uğranan üslerin sırasını belirten permütasyon kodlaması kullanılmıştır. Her bir üs bir rakamla temsil edilir. Bu rakamların yan yana sıralanışı üslerin rotada izlenme sırasını temsil eder. Örneğin *2-3-5-6* permütasyonu, uçağın *2–numaralı* üsten kalkacağını ve daha sonra sırası ile *3, 5* ve son olarak da *6–numaralı* üsse uğrayacağını gösterir. Rota matrisinde her satır bir uçağa ait rotayı temsil eder. Toplamda 3 uçak olan bir çözüm, 3 satırlı bir matrisle kodlanmıştır. Çizelge 6.1'de üç farklı uçak tipi içeren bir çözüm için örnek bir rota matrisi verilmiştir. Bir uçağa ait rota, o uçağın ana üssünden başlamalı ve ana üssünde bitmelidir. Bir çözümde her uçağa ait rota, sabit bir uzunluğa sahip değildir. Buradaki rota uzunluğu o rotada bulunan bacak sayısını ifa eder ve bacak sayısı ne kadar çok olursa o rota daha uzun bir rota olarak değerlendirilir. Her uçağın rotası farklı uzunlukta olabilir. Problemdeki maksimum bacak sayısı kısıtı bir rotanın

içerisinde uğranabilecek maksimum üs sayısını göstermekte olup rota uzunlukları "3 ≤ *rota uzunluğu* ≤ *maksimum bacak sayısı"* arasında herhangi bir değer alabilmektedir.

Çizelge 6.1 Örnek Rota Matrisi

1. Uçak	1	6	5	4	3	1			
2. Uçak	1	4	2	4	7	3	2	5	1
3. Uçak	3	5	1	5	7	4	6	3	

İkinci matris ise bir üsse ait talebin hangi uçak ile hangi sırada yükleneceği bilgisini saklar. Her bir uçağa ait sütun üç adet bilgi içerir. İlk yazılan sayı, taşınan ürünün çıkış üssünü diğer bir deyişle deposunu; ikinci yazılan sayı, ürünün hangi üsse taşınacağı bilgisini içerir. Üçüncü sayı ise taşınan ürün miktarını gösterir. Yukarıdan aşağı doğru sıralanan taşımalar, bu uçağa taleplerin yüklenme sırasını gösterir. İkinci matris için üç uçaktan oluşan bir örnek Çizelge 6.2'de verilmiştir. Burada birinci uçak *6–numaralı* üsten *1–numaralı* ve *3–numaralı* üsse, *5–numaralı* üsten *4–numaralı* üsse giden ürünleri taşır. Birinci uçağa ilk önce *6–numaralı* üsten *1–numaralı* ve *3–numaralı* üsse giden talepler yüklenir. Daha sonra uçak *5–numaralı* üsten *4–numaralı* üsse giden talep yüklenir. Uçak rotada ilerledikçe yüklenen talepler ilgili üslere bırakılır. Taşıma matrisi aynı zamanda uçaklara yükleme sırasını gösterir. Uçakların soldan sağa sıralanış sırası aynı zamanda uçaklara ait yükleme sırasıdır. Uçaklara farklı yükleme sırası farklı çözümler üreteceği için bu bilginin saklanması gerekmektedir.

Çizelge 6.2 Örnek Yükleme Matrisi

1. UÇAK	3. UÇAK	2. UÇAK
6→1: 3	**3**→4: 3	**1**→2: 5
6→3: 7	**5**→6: 37	**1**→4: 32
5→4: 3	**5**→7: 8	**1**→7: 8
	7→4: 7	**4**→2: 5
	4→6: 8	**4**→3: 8
		2→3: 4
		2→5: 4
		2→7: 8
		7→2: 4
		3→1: 4
		5→1: 2

6.3.2. Başlangıç popülasyonunun oluşturulması

Başlangıç popülasyonunun oluşturulmasında iki farklı yöntem izlenmiştir. İlkinde "Clark&Wright Savings" algoritması (Toth ve Vigo, 2001) ilk popülasyonun %10'luk kısmının oluşturulmasında kullanılmıştır. İlk popülasyonun geri kalan kısmında ise rastgele rotalar oluşturularak bireyler üretilmiştir. Bütün talepleri taşıyamayan çözümler ilk popülasyona dahil edilmemiştir.

6.3.2.1. Clarke &Wright tasarruf algoritması

ARP için en çok kullanılan sezgisellerden birisi de Clark&Wright tasarruf algoritmasıdır. İlk popülasyonun %10'luk kısmı bu algoritma kullanılarak oluşturulmuştur. Bu bireylerden elde edilen yapı taşlarının, hücresel genetik algoritma içerisinde kullanılması ve daha iyi sonuçların erken elde edilmesi amaçlanmıştır.

Clark&Wright tasarruf algoritması "*Depo – 1 numaralı üs – Depo"* ile "*Depo – 2 numaralı üs – Depo"* rotalarının "*Depo – 1 numaralı üs – 2 numaralı üs – Depo"* olarak birleştirilmesi sonucu elde edilen kazancın kullanılması prensibine dayanır. Tasarruflar (6.1) denklemi ile hesaplanır. Burada s_{ij} elde edilen kazancı, c_{iA_k} i üssünden k uçağının ana üssüne (A_k) gitme maliyetini, c_{A_kj} k uçağının ana üssünden (A_k), j üssüne gitme maliyetini, c_{ij} ise i üssünden j üssüne gitmenin maliyetini gösterir. AHTP'de her uçak tipi için ana üs yani farklıdır. Dolayısıyla her araç için farklı tasarruf değerleri hesaplanmıştır. Daha sonra hesaplanan tasarruf değerleri büyükten küçüğe doğru sıralanır.

$$s_{ij} = c_{iA_k} + c_{A_kj} - c_{ij} \tag{6.1}$$

Problemdeki uçaklar kümesinden rastgele bir uçak seçilir. Bu uçağa ait hesaplanan tasarruflar listesinin ilk %10'luk kısmındaki elemanları arasından rastgele bir tasarruf değeri seçilir. Seçilen tasarrufa ait üsler bu uçağa ait rotada daha önce kullanılmadı ise, üsler sırası ile rotanın sonuna eklenir. Bu şekilde bütün tasarruf değerleri tamamlanana kadar devam edilir. Daha sonra diğer uçağa geçilir ve bu uçak için işlemler tekrarlanır. Bu işlemler tüm uçaklar tamamlanıncaya kadar devam edilir.

Bir çözümdeki bütün uçaklar için rotalar oluşturulduktan sonra aşağıda anlatılan uçak yükleme prosedürü ile uçakların taşımaları nasıl icra edeceği belirlenir. AHTP'de bütün taleplerin taşınmadığı bireyler popülasyona dahil edilmez. Bu yüzden bütün talepleri taşıyamayan bireyler tamir edilerek popülasyona dahil edilir.

Birey tamir işlemi, taşınamayan taleplerin çıkış ve varış üsleri, rotada uygun noktalara eklenerek yapılır. Tamir işleminin en az ekleme yapılarak tamamlanması için en çok hangi üssün talebi taşınamamış ise o üs öncelikle rotaya eklenir.

6.3.2.2. Rastgele rota oluşturma

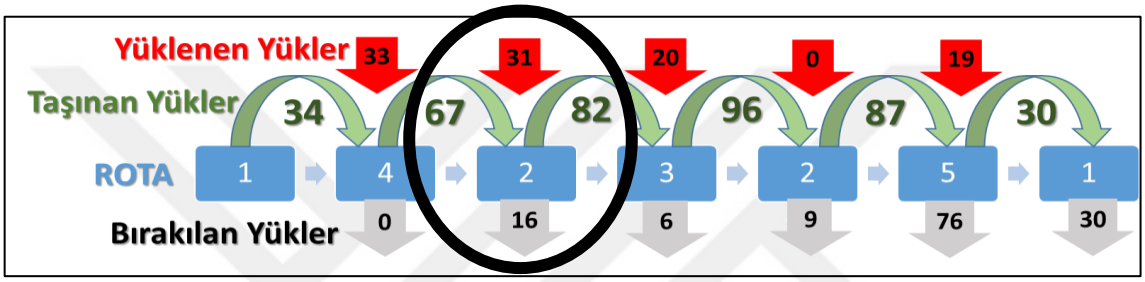
Çözülen probleme ait maksimum uçak sayısı, talepler, maksimum bacak ve uğrama sayısı bilgileri alınır. Oluşturulan bireydeki uçak sayısı "*1– maksimum uçak sayısı*" arasında bir değerden rastgele belirlenir. Uçak tipine ait ana üs o uçağın rotasının ilk üssü olarak atanır. Bu uçağa ait rota uzunluğu "*3– maksimum bacak sayısı*" arasında bir değerden rastgele belirlenir. Ana üsten sonra uğranılacak üs, talep matrisinde ana üsten diğer üslere gidecek ürün var ise bu üstler arasından, yok ise gönderisi olan üstler arasından rastgele seçilir. Seçilen üs rotaya eklendikten sonra bu üs için "maksimum uğrama sayısı" kontrolü yapılır. Bu üsse bu rota içerisinde maksimum uğrama sayısı kadar uğranmış ise bu üs yerine rotanın devamına eklenmek üzere başka bir üs seçilir. Bu şekilde rastgele belirlenen rota uzunluğu kadar üs, rotaya teker teker eklenir. Belirlenen rota uzunluğuna ulaşıldığında bu uçağa ait ana üs son nokta olarak atanır. Belirlenen uçak sayısı kadar rota yukarıdaki şekilde oluşturulur. Örnek bir rota Şekil 6.6'da verilmiştir. Bireye ait rotanın oluşturulmasından sonra bu rota ile taşımaların nasıl yapılacağı, uçak yükleme prosedürü ile belirlenir. Bütün talepleri taşıyamayan çözümler ilk popülasyona dahil edilmez.

6.3.2.3. Uçak yükleme

Bir bireydeki uçak rotlarından rastgele bir uçak rotası seçilir. Rotanın ilk üssünden diğer üslere taşınacak talepler talep matrisinden alınır. Yukarıda seçilen rota ile bu taleplerin taşınabilenleri bu uçağa yüklenir, kapasite kontrolü yapılır ve rotadaki bir sonraki üsse geçilir. Bu üsse bırakılacak talepler bırakılır, bu üsten diğer üslere gönderilecek taleplerden rotanın geri kalan kısmında taşınabilenler uçağa yüklenir, kapasite kontrolü yapılır ve diğer

üsse geçilir. Bu şekilde rotadaki son üs olan ana üsse kadar devam edilir. Seçilen rotadaki yüklemeler tamamlandıktan sonra rastgele bir uçak rotası daha seçilir ve yukarıda anlatılan yükleme işlemi tekrarlanır. Bireydeki tüm uçaklar için yüklemeler tamamlanana kadar yukarıdaki prosedürler tekrarlanır.

Örnek bir talep matrisi Çizelge 6.3'de, yükleme matrisi Çizelge 6.4'de, uçağa ait rota, uçak yükleme ve boşaltmaları Şekil 6.6'da gösterilmiştir. Yükleme matrisi aynı zamanda taleplerin uçağa yükleme sırasını da gösterir.



Şekil 6.6 Uçak Rotası ve Uçağa Ait Yükleme/Boşaltmalar

Çizelge 6.3 Uçak Yüklemeleri İçin Örnek Talep Matrisi

	1	**2**	**3**	**4**	**5**
1	X	12	-	-	22
2	-	X	6	-	25
3	11	9	X	-	-
4	-	4	-	X	29
5	19	-	-	-	X

Çizelge 6.4 Örnek Yükleme Matrisi

Depo Üs	**Talep Üssü**	**Miktar**
1	2	12
1	5	22
4	2	4
4	5	29
2	3	6
2	5	25
3	1	11
3	2	9
5	1	19

Yükleme esnasında yapılan kapasite kontrolünde, yükleme sonrasında taşınacak yük kapasiteden büyük ise bu talep için talep bölme işlemi uygulanır. Rotada kapasitenin aşıldığı üsse ait taleplerden, bu rota ile taşınabilen en büyük olanı seçilir. Taşınan en büyük talebin, kapasite fazlası yük miktarından büyük veya küçük olmasına göre iki farklı talep bölme işlemi yapılır. Bu iki farklı talep bölme işlemi aşağıda açıklanmıştır:

- **Taşınan En Büyük Talep ≥ Kapasite Fazlası Yük Miktarı:** Bu durumda, taşınabilen en büyük talebin kapasitenin üzerindeki miktarı uçağa yüklenemez. Böylelikle kapasitenin aşıldığı üsteki diğer taleplerin ve taşınabilen en büyük talebin bir kısmının taşınması sağlanır. En büyük talebin taşınan miktarı denklem (6.2)'ye göre hesaplanır. Bu denklemde "TM" taşınan miktarı, "BT" taşınabilen en büyük talebi, "KM" kapasite aşıldığındaki toplam yük miktarı, "UK" uçak kapasitesini ifade eder.

$$TM = BT - (KM - UK) \tag{6.2}$$

Örneğin bir uçağın kapasitesi 100 birim olsun. Kapasitenin aşıldığı anda uçağa yüklenen toplam talep ise 120 birim olsun. Uçağa yüklenen taleplerden en büyük olanı ise 50 birim olsun. Uçağın kapasitesinin üzerindeki toplam yük 20 birimdir. Bu miktar en büyük talepten küçüktür "50>20". En büyük talebin ne kadarının taşınacağı denklem (6.2)'den TM= 50-(120-100) = 30 birim olarak bulunur. En büyük talebin 30 birimi bu uçağa yüklenir ve geri kalan kısmı diğer uçaklar tarafından taşınmak üzere üste bırakılır.

Bu talep bölmedeki amaç taşınan en büyük talebin bölünerek, uçak kapasitesinin daha küçük yükler ile daha etkin kullanılmasını sağlamaktır. Eğer bu büyük talep bölünmez ise bu talebin tek bir uçak ile taşınmasında, kapasite aşılması yönünden problemlerle karşılaşılması kaçınılmazdır. Bu talep bölünerek kapasite kısıtının sağlanmasına yardımcı olunur.

- **Taşınan En Büyük Talep < Kapasite Fazlası Yük Miktarı:** Bu durumda, uçak yükleme listesinden en baştan itibaren sıra ile eklendiğinde kapasiteyi aşan talep bulunur. Bu talebe bölme işlemi yapılır. Bu talebin kapasite dolduktan sonra kalan kısmı bu rotada taşınmaz.

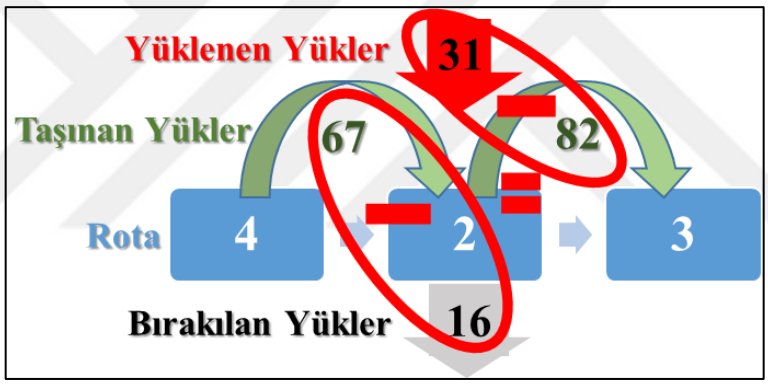
6.3.2.4. Giren çıkan ürün eşitliği

Bir uçağa ait rotanın herhangi bir noktasındaki üste, ürünlere ait akış dengesi sağlanmak zorundadır. Uçak bir önceki üsten gelirken üzerinde bulunan toplam ürün miktarıyla (GeÜ) bu üsse bırakılan ürün miktarının (BÜ) farkı; bu üsten kalkarken üzerinde bulunan toplam ürün miktarıyla (GiÜ) bu üste yüklenen ürün miktarının (YÜ) farkı birbirine eşittir. Şekil 6.6'teki örnek yüklemelerin rotanın üçüncü üssü olan *2-numaralı* üsteki akış dengesi Şekil 6.7'da gösterildiği gibidir. Akış dengesi eşitlik (6.3)'de verilmiştir.

$$Ge\text{Ü} - B\text{Ü} = Gi\text{Ü} - Y\text{Ü} \tag{6.3}$$

$$67 - 16 = 82 - 31$$

$$51 = 51$$



Şekil 6.7 Ürün Akış Dengesi

6.3.3. Uyum fonksiyonu

Uyum fonksiyonunun hesaplanmasında bir bireydeki uçakların toplam kat ettiği mesafe dikkate alınır. AHTP'de sadece bütün taleplerin taşınamadığı çözümler, çözümsüz olarak değerlendirilir. Probleme ait diğer kısıtlar sağlanmadığında, problem çözümsüz olarak değerlendirilmemiştir ve sağlanamayan kısıtlar uyum fonksiyonuna bir ceza değeri olarak yansıtılmıştır. Ceza değeri planlayıcı tarafından, plan yapılırken içinde bulunulan şartlara göre değiştirilerek istenilen özelliklere daha uygun rotaların oluşturulması sağlanır. Örneğin pilotlara ait görev sürelerinin kesinlikle aşılması istenmiyorsa çalışma süresi cezası çok büyük bir değer verilerek görev süresini aşmayan rotaların oluşturulması sağlanabilir. Ya da bunun tersi düşünüldüğünde, planlayıcı çalışma süresi cezasına küçük bir değer

vererek bu kısıdın ihlal edilebileceği çözümlerin oluşturulmasını sağlayabilir. Bu çalışmada, 1000 ile 6000 arasında değişen ceza değerleri kullanılmıştır. Uyum fonksiyonuna eklenen cezalar aşağıda sıralanmıştır.

- **Çalışma Süresi Cezası:** Pilotların toplam görev süresi, maksimum görev süresi üzerinde ise eklenir.

- **Menzil Cezası:** Uçakların rotasındaki iki üs arasındaki mesafe uçağın menzilinden uzunsa eklenir.

- **Hava Riski Cezası:** Uçakların rotası boyunca maruz kaldıkları hava riskinin ortalaması, maksimum hava koşulları riskinin üzerinde çıkarsa eklenir.

- **Hasarlı Pist Cezası:** Uçakların rotasının ilk iki bacağında bombalanmış piste sahip bir üs varsa ve bu üslere iniş süresi pist onarım süresinden küçük ise eklenir.

- **Alçak Uçuş Cezası:** Bir pilotun toplam alçak uçuş süresi, maksimum alçak irtifa uçuş süresi üzerinde ise eklenir.

- **Hava Savunma Cezası:** Bir uçak rotasının hava savunma sistemlerine ortalama uzaklığı, hava savunma sistemlerine maksimum uzaklık değeri üzerinde ise eklenir.

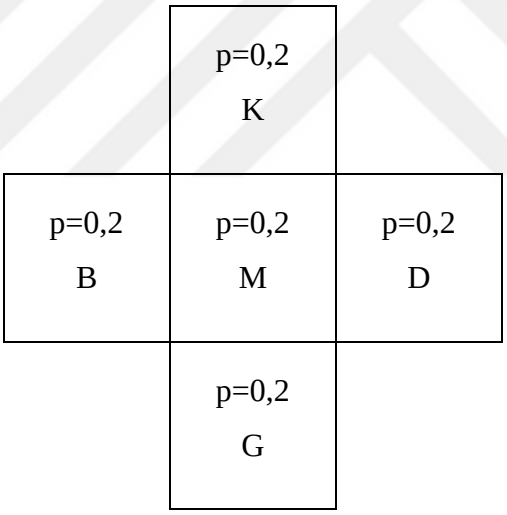
- **Maksimum Uğrama Cezası:** Bir ayrıt, maksimum geçiş sayısından fazla kez kullanıldıysa eklenir.

6.3.4. Ebeveyn seçimi

Hücresel genetik algoritmada bir sonraki nesli oluşturmak için ebeveynler seçilir. Ebeveynler komşu hücreler arasından seçilir. Komşu hücreler, popülasyon içerisinden belirli kriterlere göre oluşturulur. Komşuluk tanımı ve yapısı algoritmanın etkinliği için çok önemlidir.

Algoritmada seçim baskısının göstergelerinden biri de ele geçirme zamanıdır (Takeover Time). Ele geçirme zamanı, popülasyon içerisindeki en iyi bireyin kendi kopyalarını tüm popülasyona kopyalama zamanı olarak tanımlanır. Bu çalışmada Von Neumann komşuluk yapısı kullanılmıştır. Von Neumann ya da L5 komşuluk yapısı en uzun ele geçirme zamanına sahip komşuluk yapılarından birisidir (Sarma ve De Jong, 1996). Bu sayede algoritmanın çeşitlendirme (diversification) yapması hedeflenmiştir.

Von Neumann komşuluk yapısında bir merkez hücre ve bu hücreye dört yönde (K: kuzey, G: güney, D: doğu, B: batı) 1 birim uzaklıkta bulunan komşu hücreleri vardır. Bu komşuluk yapısında merkez hücre ve komşuları olmak üzere toplam 5 adet hücre vardır. 5 hücreden 2 tanesi ebeveyn olarak seçilir. Her bir hücrenin ebeveyn olarak seçilmesi için eşit şansı vardır ve seçilme olasılıkları p=0,2'dir. Von Neumann komşuluk yapısı Şekil 6.8'de gösterilmiştir (Simoncini vd. , 2006).



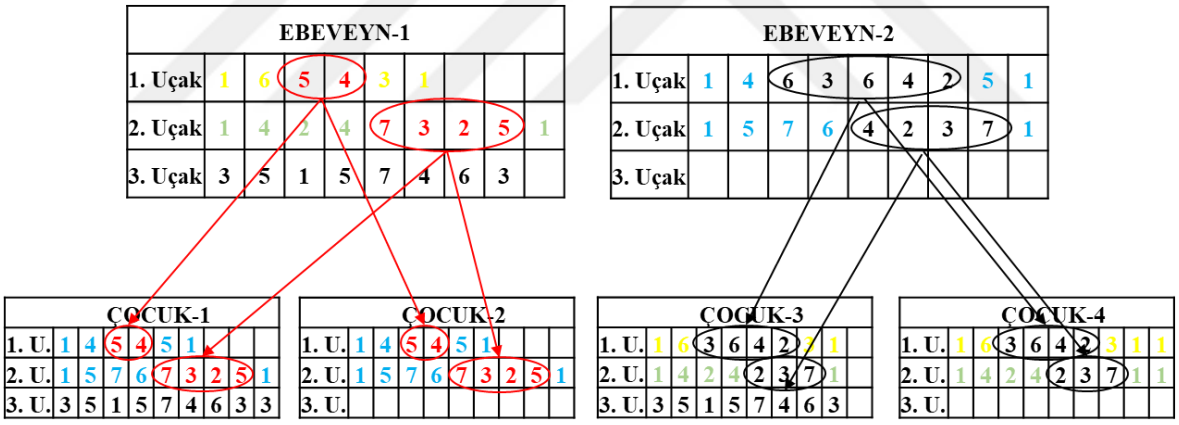
Şekil 6.8 Von Neumann Komşuluk Yapısı

6.3.5. Çaprazlama

Bu çalışmada, Kora ve Yadlapalli (2017) tarafından kullanılan sıralı çaprazlama (Order Crossover-OX) yöntemi kullanılmıştır. Sıralı çaprazlamada birinci ebeveynden bir rota parçası seçilir. Seçilen bu parça birinci çocuğa kopyalanır. Birinci çocuğun geri kalan kısımları, diğer ebeveynde bölünen noktadan itibaren bu ebeveyndeki sırasıyla kopyalanarak oluşturulur.

Çaprazlamalar iki farklı durum için ayrı ayrı yapılmıştır. İlk durumda çözümdeki uçaklara kendi içlerinde çaprazlama uygulanmıştır. Örneğin ebeveyn-1'deki 1'inci uçak ile ebeveyn-2'deki 1'inci uçak rotalarına kendi içinde çaprazlama yapılmıştır. İkinci durumda ise ebeveynlerdeki uçak rotaları rastgele seçilerek çaprazlama yapılmıştır. Çaprazlamaya ait bir örnek Şekil 6.9'da verilmiştir.

Ebeveyn-1'den birinci uçak rotasından rastgele bir parça seçilir. Ebeveyn-1'den seçilen parça ve ebeveyn-2'den seçim dışında kalan parçalar Çocuk-1'e kopyalanır. Eğer maksimum bacak sayısından fazla bacak kullanılmışsa en sağdan itibaren bacak sayısı maksimuma eşitlenene kadar, üsler teker teker rotadan çıkarılır. Eğer bir ebeveynde bir uçak hiç kullanılmamışsa bu uçağa ait rota bir çocuğa aynen kopyalanırken diğerinde hiç kullanılmaz. Bu işlemler ebeveyn-1 ve ebeveyn-2'nin sırası değiştirilerek tekrarlanır. Böylelikle toplam 4 çocuk oluşturulur. İkinci durumda yapılan işlemler aynıdır. Tek fark, bu işlemlerin aynı uçaklar için değil rastgele seçilen uçak rotaları için yapılmasıdır.



Şekil 6.9 Çaprazlama Örneği

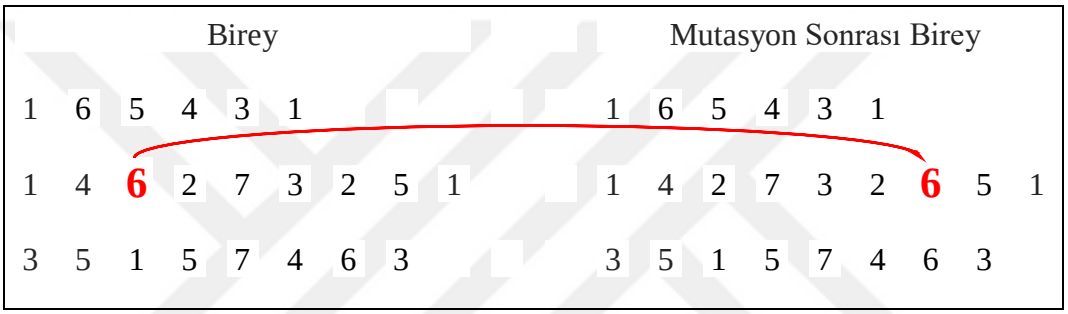
6.3.6. Mutasyon

Bu çalışmada birbirinden farklı üç mutasyon türü kullanılmıştır. Bu mutasyon türleri literatürdeki mevcut operatörlerin bu probleme uyarlanmış halidir. Her mutasyon türü, bireye ait bazı özellikleri değiştirerek farklı bireyler oluşturur. Oluşturulan yeni bireyler, daha iyi uyum değerlerine sahip ise türetildiği birey popülasyondan çıkarılarak, mutasyona uğramış olan birey popülasyonda tutulur. Mutasyonlardan hangisinin kullanılacağı rastgele

normal dağılıma göre seçilir. Bu problem için tasarlanan mutasyon tipleri aşağıda açıklanmıştır.

6.3.6.1. Araya sokma mutasyonu

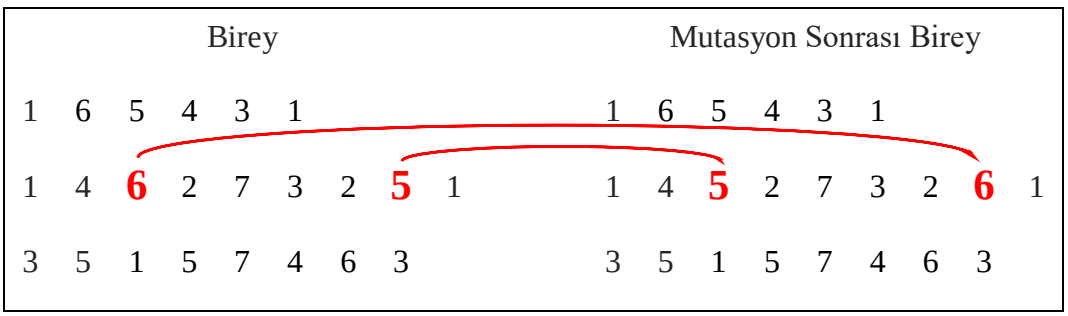
Bireyden rastgele bir uçak rotası seçilir. O rotada rastgele bir üs seçilir. Seçilen bu üs rotadan çıkarılır. Çıkarılan bu üs aynı rota içinde fakat bu kez farklı bir konumda olacak şekilde tekrar rotaya ilave edilir (Larrañaga vd. , 1999). Bu mutasyona ait basit bir örnek Şekil 6.10'da verilmiştir.



Şekil 6.10 Araya Sokma Mutasyonu Örneği

6.3.6.2. Değiş tokuş mutasyonu

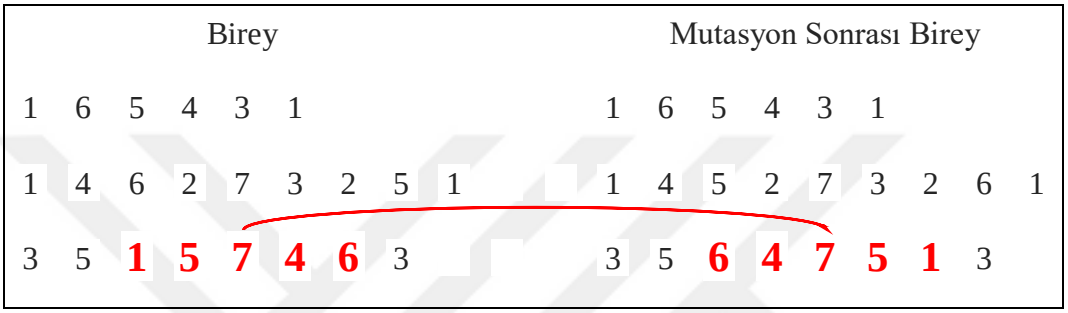
Bireyden rastgele bir uçak rotası seçilir. Seçilen rota üzerinde rastgele iki üs seçilir. Bu üslerin rota içerisindeki konumları birbiriyle yer değiştirilir (Larrañaga vd. , 1999). Bu mutasyona ait bir örnek Şekil 6.11'de verilmiştir.



Şekil 6.11 Değiş Tokuş Mutasyonu Örneği

6.3.6.3. Ters çevirme mutasyonu

Bireyden rastgele bir uçak rotası seçilir. Bu rota içerisinden rastgele bir uzunluğa sahip bir rota parçası belirlenir. Belirlenen rota parçasına ait üs sıralaması ters çevrilir. İlk uğranan üs en son, ikinci uğranan üs ise sondan bir önce uğranan üs olacak şekilde rota parçasındaki diğer bütün üslerin uğrama sırası benzer şekilde ters çevrilir (Abdoun vd. , 2012). Bu mutasyona ait bir örnek Şekil 6.12'de verilmiştir.



Şekil 6.12 Ters Çevirme Mutasyonu Örneği

6.3.7. Yer değiştirme

Yer değiştirmenin, algoritmanın arama kabiliyeti üzerindeki etkisi çok büyüktür. Optimum çözümü bulmak için gerekli olan jenerasyon sayısı dikkate alındığında, senkronize olmayan hGA senkronize olan hGA'ya göre daha çabuk optimum çözümü yakalamaktadır (Alba vd. , 2002). Bu çalışmada senkronize olmayan yer değiştirme yöntemi kullanılmıştır.

Çaprazlama ve mutasyon sonrası yeni üretilmiş olan bireylerin uyum değerleri hesaplanır. Bireyler hesaplanan bu uyum değerlerine göre sıralanır. En iyi uyum değerine sahip ve bütün talepleri taşıyabilen birey, merkez hücredeki bireyden daha iyi uyum değerine sahip ise yer değiştirme işlemine tabi tutulur. Bu şekilde soldan sağa satır satır popülasyondaki son bireye kadar devam edilerek daha iyi bireyler popülasyona eklenir. Bu sayede bir sonraki jenerasyonun kullanılmaya başlanması beklenilmeden daha iyi olan bireylerin bir sonraki jenerasyonun üretilmesinde hemen rol sahibi olması sağlanır. Elde edilen en iyi uyum değerine sahip birey tüm popülasyonlar boyunca daha iyisi bulunana kadar sabit tutulur.

Popülasyonun çok çabuk lokal bir optimuma yakınsamasını (Premature Convergence) önlemek için, belirli küçük bir olasılıkta merkezdeki bireyden daha kötü uyum değerine sahip bireyler yer değiştirme işlemine tabi tutulur. Bu yöntemle popülasyondaki çeşitlilik korunmaya çalışılır. Bu olasılığa kötü birey yer değiştirme olasılığı denir. En iyi uyum değerine sahip birey daha iyisi bulunana kadar her zaman popülasyonda tutulur.

6.4. hGA'ya Ait Parametre Analizi

Hücresel genetik algoritmanın işleyişi içerisinde birçok parametre kullanılmaktadır. Algoritmanın daha iyi sonuçlar üretebilmesi için bu parametrelere ait uygun değerlerin bulunması gerekir. Literatürde parametrelere ait en iyi değerlerin bulunması için kullanılan birçok yöntem ve bu yöntemlere ait birçok araştırma vardır. Bu çalışmada deneysel tasarımda yaygın olarak kullanılan "Tam Faktöriyel Tasarım" yöntemi, en iyi parametre değerlerinin elde edilmesinde kullanılmıştır (Mee, 2009). 7'inci bölüm olan deney setlerinin oluşturulması bölümünde verilen problemlerden 3 ve 14 numaralı problemler parametre optimizasyonunda kullanılmıştır. 3 ve 14 numaralı problemler yapıları bakımından birbirinden farklı problemler olduğu için parametre analizinde ele alınmış ve farklı problem tipleri için en iyi parametre değerlerinin elde edilmesi hedeflenmiştir. Algoritma üzerinde yapılan ön gözlemler sonucunda hGA'nın performansını en çok etkileyebilen dört parametre (faktör) belirlenmiştir. Bu parametreler jenerasyon sayısı, popülasyon büyüklüğü, çaprazlama oranı ve mutasyon oranıdır. hGA'nın diğer parametreleri sabit olarak alınmıştır ve değerleri yapılan ön gözlemlere göre belirlenmiştir. Parametre analizinde kullanılan faktörlerin düzeyleri, yapılan pilot çalışma ve Boyabatlı ve Sabuncuoglu (2004)'nun çalışmasında sunulan veriler göz önünde tutularak belirlenmiştir. Faktörler ve düzeyleri Çizelge 6.5'de verilmiştir.

Çizelge 6.5 Faktörler ve Düzeyleri

Faktör	**Alt Düzey**	**Üst Düzey**
Jenerasyon Sayısı	100	150
Popülasyon Büyüklüğü	49	100
Çaprazlama Oranı	0,7	0,9
Mutasyon Oranı	0,1	0,3

6.4.1. Tam faktöriyel deney tasarımı

Parametre optimizasyonunda, seçilen dört ayrı faktör ve bu faktörlere ait iki düzey olduğundan 2^4 tam faktöriyel tasarımı kullanılmıştır. 2^4 tam faktöriyel tasarımı 16 gözlem ile analiz edilebilir. Faktörlerin farklı problem tiplerinde algoritma performansına olan etkisinin analiz edilebilmesi için 2^4 tam faktöriyel tasarım iki farklı problem için tekrar edilmiştir. 3 ve 14 numaralı problemler oluşturulan deney seti içerisinden faktöriyel analiz için seçilmiştir. 3 numaralı problem Rastgele serpilmiş üslerden, yoğun ve büyük sapmalı taleplerden oluşmaktadır. 14 numaralı problem ise kümelenmiş üslerden, seyrek ve küçük sapmalı taleplerden oluşmaktadır. Birbirinin tam tam tersi özelliklere sahip iki problem deney setindeki tüm problem tiplerini temsilen seçilmişlerdir. 3 ve 14 numaralı problemlerin sahip olduğu özellikler deney seti kümesindeki tüm problem çeşitliliklerine ait özellikleri içermektedir. Duyarlılığın arttırılabilmesi için problemler 12 defa çözülmüştür. Parametre analizi için faktöriyel tasarımda $2^4 \times 12 \times 2 = 384$ adet gözlem kullanılmıştır. Faktöriyel tasarım ve gözlemlere ait sonuçlar Çizelge 6.6'de verilmiştir.

6.4.2. İstatistiksel modelin oluşturulması

Tam faktöriyel tasarıma ait gözlemler JMP yazılımı kullanılarak analiz edilmiştir. JMP yazılımının tam faktöriyel deney tasarımı ara yüzü kullanılarak bağımlı değişken ve faktörler tanımlanmıştır. Bağımlı değişken verisine uyum değeri, problemin hedefine minimizasyon, faktörler kısmına ise dört adet faktörün tipi, düzey sayısı ve değerleri girilmiştir. Daha sonra tekrar sayısı girilerek JMP yazılımının tam faktöriyel tasarımı oluşturması sağlanmıştır. Oluşturulan tabloya uyum değerlerine ait gözlemler her iki problem için de girilmiştir.

Verilerin girilmesinden sonra modele ait özelliklerin belirlendiği model ara yüzüne geçilmiştir. Her iki probleme ait uyum değerleri bağımlı değişken alanına atanmıştır. Faktörlerin tamamı seçilerek JMP yazılımının makroları arasından tam faktöriyel makrosu seçilerek, ana faktörlerin ve faktörler arası etkileşim terimlerinin tüm derecelerinin modele eklenmesi sağlanmıştır. Anlam düzeyi olarak 0,05 kullanılmıştır. Model çalıştırılarak modele ait istatistiksel çıktılar elde edilmiştir.

Çizelge 6.6 Deney Tasarımı ve Gözlem Sonuçları

Jenerasyon Sayısı	Hücre Sayısı	Çaprazlama Oranı	Mutasyon Oranı	G-1		G-2		G-3		G-4		G-5		G-6	
				P-3	P-14	P-3	P-14	P-3	P-14	P-3	P-14	P-3	P-14	P-3	P-14
100	49	0,7	0,1	3085	3425	3116	2939	3066	3278	2652	3410	2999	3228	2956	2644
100	49	0,7	0,3	3112	2313	2813	2329	2892	3028	2827	3449	3050	3131	3193	2264
100	49	0,9	0,1	2843	2633	2910	3267	2745	2947	3213	3159	3043	2798	2951	2669
100	49	0,9	0,3	3054	3574	3180	2713	2741	3327	2715	2580	2501	2481	2952	2254
100	100	0,7	0,1	2675	2109	2941	3012	3096	2965	2627	2259	2605	2820	3024	3075
100	100	0,7	0,3	2701	2328	2470	2436	2584	2752	2818	2911	2916	2800	2948	2208
100	100	0,9	0,1	2886	2423	3059	2313	2866	2600	2855	2467	2924	2896	2500	2168
100	100	0,9	0,3	2814	2387	2999	3048	2763	2468	2777	2488	2920	2419	2775	2170
150	49	0,7	0,1	2718	2280	2748	2378	3368	2778	2995	2468	2944	2324	2793	2750
150	49	0,7	0,3	2708	2272	2757	2505	2846	2505	2837	2637	2768	2104	2703	2082
150	49	0,9	0,1	2864	2049	2887	2299	2802	2741	2652	2990	2720	2580	3094	2296
150	49	0,9	0,3	2803	2250	2643	2248	2632	2424	2858	2526	3009	3024	2735	2447
150	100	0,7	0,1	3056	2672	2545	1741	2700	2527	2896	2512	3068	2479	2581	2312
150	100	0,7	0,3	2823	2302	2493	2938	2865	2653	2787	2391	3069	2427	2654	2097
150	100	0,9	0,1	2721	2108	2671	2021	2498	2053	2862	2685	2625	2424	2845	2435
150	100	0,9	0,3	2776	2182	2585	2580	2807	2161	2484	2177	2261	2073	2388	2344
Jenerasyon Sayısı	Hücre Sayısı	Çaprazlama Oranı	Mutasyon Oranı	G-7		G-8		G-9		G-10		G-11		G-12	
				P-3	P-14	P-3	P-14	P-3	P-14	P-3	P-14	P-3	P-14	P-3	P-14
100	49	0,7	0,1	3194	3497	2867	2950	2846	3159	3140	2984	2828	2864	3260	2831
100	49	0,7	0,3	2790	2221	2989	2300	2788	2791	3082	2902	2830	2290	2694	2617
100	49	0,9	0,1	2967	2438	3094	1678	3247	2599	3239	2992	2923	2666	2969	2768
100	49	0,9	0,3	2948	2796	3101	2872	3175	3112	2941	2930	2806	2190	2862	2528
100	100	0,7	0,1	2854	2992	2964	2831	2678	2843	2889	2799	2883	1613	2782	3126
100	100	0,7	0,3	2939	2942	2954	3172	2881	3166	2916	2466	2718	3027	2149	2869
100	100	0,9	0,1	2339	2621	2986	3194	2717	2495	2926	2379	3001	2500	2868	3079
100	100	0,9	0,3	2735	2779	2841	2268	2969	2146	2764	2751	2686	2236	2885	2554
150	49	0,7	0,1	2486	2528	2969	2793	2755	2236	2924	3107	2711	2940	2996	2437
150	49	0,7	0,3	2564	2728	2706	2610	2750	2133	2927	2643	2789	2447	2769	2488
150	49	0,9	0,1	2891	2131	2698	3220	2790	2152	2608	3166	2655	2415	2848	2991
150	49	0,9	0,3	2883	2405	2891	2066	2856	2137	2745	2499	2970	2050	2691	1630
150	100	0,7	0,1	2682	2381	2795	2159	2633	2948	2868	2444	2750	2449	2795	3035
150	100	0,7	0,3	2812	1677	2808	2329	2547	2116	2459	2299	2713	1592	2846	2251
150	100	0,9	0,1	2954	2197	2649	2185	2794	2501	2922	2489	2944	2446	2502	2223
150	100	0,9	0,3	2572	2249	2781	2290	2848	2072	2068	2149	2760	2226	2737	2213

6.4.3. İstatistiksel analiz

JMP istatistiksel analiz programı kullanılarak, 3 ve 14 numaralı problemlerin çözümlerinden elde edilen gözlemler istatistiksel olarak incelenmiştir. Elde edilen bulgular açıklanmıştır.

6.4.3.1. *3–Numaralı* probleme ait analiz

3 numaralı probleme ait varyans analizi Çizelge 6.7'da parametre tahminleri Çizelge 6.8'de verilmiştir. Varyans analizi çizelgesinde modelde önemli etkisi olan en az bir faktörün olduğunu görülmektedir. Parametre tablosundan ise Maksimum Jenerasyon, Popülasyon Büyüklüğü ve Mutasyon Olasılığı faktörlerinin etkilerinin anlamlı olduğu görülmektedir. Çaprazlama Olasılığı ana faktörü ve diğer etkileşimli faktörler istatistiksel olarak anlamlı bulunmamıştır.

Çizelge 6.7 Problem 3-Varyans Analizi

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	15	2004674.5	133645	4.2796
Error	176	5496169.8	31228	**Prob > F**
C. Total	191	7500844.3		<.0001*

Çizelge 6.8 Problem 3-Parametre Tahminleri

Term	Estimate	t Ratio	Prob>|t|
Jenerasyon (100,150)	-62.48277	-4.90	<.0001*
Popülasyon Büyüklüğü (7,10)	-61.38491	-4.81	<.0001*
Çaprazlama Olasılığı (0.7,0.9)	-8.327503	-0.65	0.5146
Mutasyon Olasılığı (0.1,0.3)	-34.51097	-2.71	0.0075*
Jenerasyon*Popülasyon Büyüklüğü	14.943339	1.17	0.2429
Jenerasyon*Çaprazlama Olasılığı	-17.65927	-1.38	0.1679
Jenerasyon*Mutasyon Olasılığı	-1.825054	-0.14	0.8864
Popülasyon Büyüklüğü*Çaprazlama Olasılığı	-5.679488	-0.45	0.6566
Popülasyon Büyüklüğü*Mutasyon Olasılığı	-1.252485	-0.10	0.9219
Çaprazlama Olasılığı*Mutasyon Olasılığı	4.5284695	0.36	0.7230
Jenerasyon*Popülasyon Büyüklüğü*Çaprazlama Olasılığı	-13.97354	-1.10	0.2747
Jenerasyon*Popülasyon Büyüklüğü*Mutasyon Olasılığı	-12.62289	-0.99	0.3236
Jenerasyon*Çaprazlama Olasılığı*Mutasyon Olasılığı	-3.864125	-0.30	0.7623
Popülasyon Büyüklüğü*Çaprazlama Olasılığı*Mutasyon Olasılığı	-8.745863	-0.69	0.4938
Jenerasyon*Popülasyon Büyüklüğü*Çaprazlama Olasılığı*Mutasyon O.	-21.67073	-1.70	0.0910

6.4.3.2. *14–Numaralı* probleme ait analiz

14–numaralı probleme ait varyans analizi Çizelge 6.9'de, parametre tahminleri Çizelge 6.10'da verilmiştir. Varyans analizi çizelgesinde modelde önemli etkisi olan en az bir faktörün olduğunu görülmektedir. Parametre tablosundan ise Maksimum Jenerasyon, Popülasyon Büyüklüğü, Çaprazlama Olasılığı, Mutasyon Olasılığı ana faktörlerinin ve Maksimum Jenerasyon*Popülasyon Büyüklüğü*Çaprazlama Olasılığı*Mutasyon Olasılığı etkileşimli faktörünün etkilerinin anlamlı olduğu görülmektedir. Diğer faktörler istatistiksel olarak anlamlı bulunmamıştır.

Çizelge 6.9 Problem 14-Varyans Analizi

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	15	9508391	633893	5.6003
Error	176	19921125	113188	**Prob > F**
C. Total	191	29429515		<.0001*

Çizelge 6.10 Problem 14-Parametre Tahminleri

Term	Estimate	t Ratio	Prob>|t|
Jenerasyon (100,150)	-161.7901	-6.66	<.0001*
Popülasyon Büyüklüğü (7,10)	-84.04346	-3.46	0.0007*
Çaprazlama Olasılığı (0.7,0.9)	-58.49026	-2.41	0.0170*
Mutasyon Olasılığı (0.1,0.3)	-75.24602	-3.10	0.0023*
Jenerasyon*Popülasyon Büyüklüğü	3.8682628	0.16	0.8736
Jenerasyon*Çaprazlama Olasılığı	20.10984	0.83	0.4087
Jenerasyon*Mutasyon Olasılığı	-16.72867	-0.69	0.4917
Popülasyon Büyüklüğü*Çaprazlama Olasılığı	-13.60629	-0.56	0.5759
Popülasyon Büyüklüğü*Mutasyon Olasılığı	29.114004	1.20	0.2321
Çaprazlama Olasılığı*Mutasyon Olasılığı	22.575188	0.93	0.3538
Jenerasyon*Popülasyon Büyüklüğü*Çaprazlama Olasılığı	5.1294537	0.21	0.8329
Jenerasyon*Popülasyon Büyüklüğü*Mutasyon Olasılığı	-12.94503	-0.53	0.5946
Jenerasyon*Çaprazlama Olasılığı*Mutasyon Olasılığı	-21.85768	-0.90	0.3692
Popülasyon Büyüklüğü*Çaprazlama Olasılığı*Mutasyon Olasılığı	-27.98126	-1.15	0.2507
Jenerasyon*Popülasyon Büyüklüğü*Çaprazlama Olasılığı*Mutasyon O.	59.196686	2.44	0.0158*

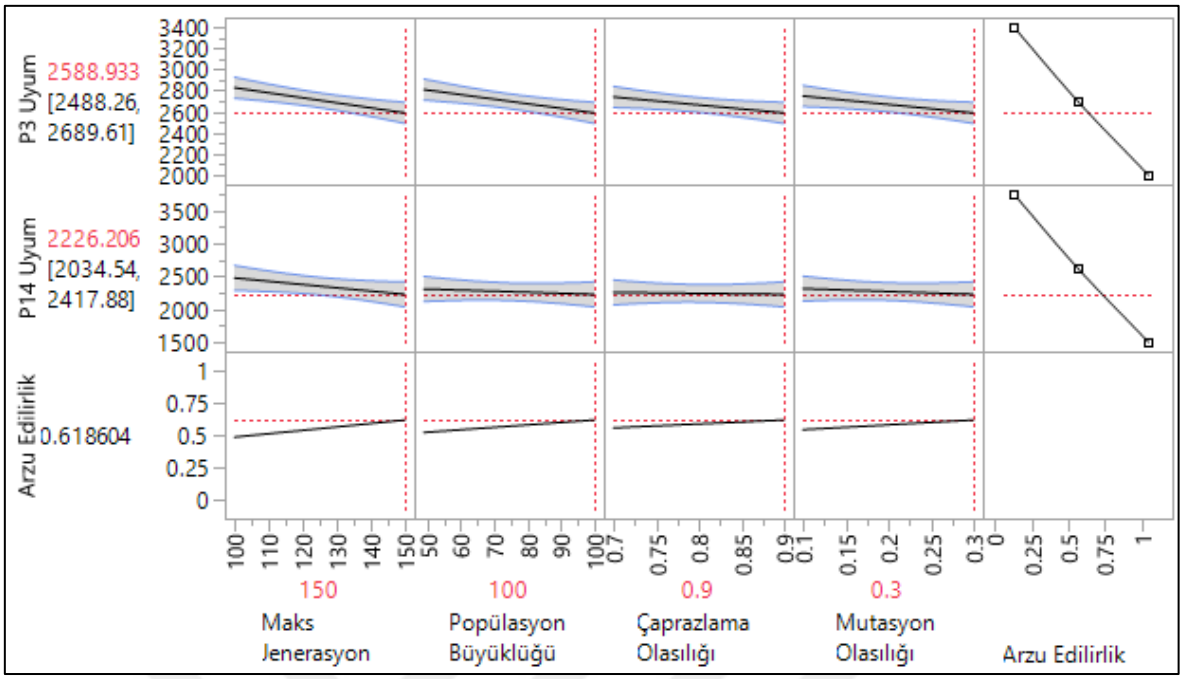
6.4.4. Bulgular

Genel olarak her iki problem için de elde edilen sonuçlar incelendiğinde ana faktörler olarak seçilen faktörler, hGA'nın performansına etki etmektedir. Faktörler arası etkileşimlerin genel olarak istatistiksel olarak önemsiz olduğu söylenebilir. Anlam düzeyi 0.1 seviyesine çıkarıldığında her iki problem için de problem Maksimum Jenerasyon*Popülasyon Büyüklüğü*Çaprazlama Olasılığı*Mutasyon Olasılığı etkileşimli faktörünün etkisinin anlamlı olduğu görülmektedir.

JMP yazılımında "Tahmin Profili (Prediction Profiler)" özelliği bulunmaktadır. Tahmin profili, faktörlerin değişmesi durumunda cevap yüzeyinin nasıl etkileneceğini görsel olarak ortaya koyar. Bu özellik ile modeldeki faktör değişimlerinin modelde ve diğer faktörler üzerinde nasıl bir etki yarattığı interaktif bir şekilde gözlemlenebilmektedir. Yanıt yüzeyine ait tahmin profilinin bir diğer özelliği ise bağımlı değişken tanımlanırken tanımlanan amaca göre bu amacı maksimize edecek uygun faktör değerlerini otomatik olarak bulan optimizasyon özelliğidir.

Optimizasyon özelliği mevcut faktörler ve bunların düzeylerini göz önünde tutarak, amacı minimizasyon olan problem tipleri için uyum değerini minimum yapan faktör değerlerini bulur. Uyum değeri aynı zamanda bir arzu edilirlik fonksiyonu ile gösterilir ve optimizasyon aşamasında maksimum arzu edilirlik hesaplanır. Faktörlere ait verilere kesikli veri tipinde dahi olsa tahmin profilinde bu veriler sürekli veriler olarak değerlendirilir. Faktörlerin düzeyleri arasındaki herhangi bir değer için arzu edilirlik veya uyum değeri bulunabilir.

Tahmin profili, her problemin uyum değeri için ayrı ayrı ya da her iki probleme ait uyum değeri için aynı anda analiz yapabilmektedir. Parametre analizinin amacı, her problem tipinde en iyi uyum değerlerini verecek faktör düzeylerini bulmaktır. Bu yüzden her iki probleme ait uyum değerleri aynı anda kullanılarak uyum değerini minimize eden faktör seviyeleri tahmin profilinden elde edilmiştir. Tahmin profilinden elde edilen faktör grafikleri ve optimizasyon özelliğinin kullanılması sonucu elde edilen faktör değerleri Şekil 6.13'de görülmektedir.



Şekil 6.13 Tahmin Profili

Elde edilen parametreler, deney setinde oluşturulan problemlerin çözümünde kullanılan hGA ve kGA'nın parametreleri olarak kullanılmıştır. Maksimum jenerasyon için 150, popülasyon büyüklüğü için 10, çaprazlama olasılığı için 0,9, mutasyon olasılığı için 0,3 değerleri en iyi uyum değerlerini veren faktör düzeyleri olarak bulunmuştur.

7. DENEY SETİNİN OLUŞTURULMASI

Bu çalışmaya konu olan askeri hava taşıma problemi daha önceki bölümlerde anlatıldığı gibi literatürdeki problemlerden farklılıklar göstermektedir. Probleme ait talep matrisi en önemli farklılıktır. Problemdeki üs sayısına bağlı olarak problemde ele alınması gereken talep sayısı $N^2–N$ sayısına kadar çıkabilmektedir. Bu durum problem boyutunun çok çabuk büyümesine neden olmaktadır. Klasik bir araç rotalama probleminde 10 şehirlik bir örnek için 10 adet talep bulunmaktayken, AHTP'de 10 şehirlik bir örnekte 10 ile 90 arasında değişen miktarda talep bulunabilmektedir.

Literatürdeki mevcut araç rotalama problemleri örnekleri incelendiğinde, talep matrislerinin yapısının incelenen probleme göre uygun olmadığı görülmüştür. Talep matrisleri eldeki probleme uygun bir yapıya getirilmek istendiğinde ise literatürdeki örnek problemin orijinal halinden çok uzaklaşıldığı, elde edilecek çözümlerin orijinal yapısında olamayacağı anlaşılmıştır. Eldeki problemin diğer özellikleri de dikkate alındığında probleme uygun yeni deney setinin oluşturulması gerektiği anlaşılmıştır. Bu bölümde örnek deney setinin nasıl oluşturulduğu anlatılmıştır. Uchoa vd. (2017)'nin yaptıkları çalışmada kullandıkları deney setini oluşturma adımları AHTP'ye uyarlanarak, benzer bir sınıflandırma yöntemi kullanılmıştır. Problem setlerinin oluşturulmasında literatürdeki birçok örnek problemde olduğu gibi iki-boyutlu Euclidean uzaklıklar kullanılmıştır. Üslerin konumları [0, 500] aralığında tam sayılı koordinatlar ile gösterilmiştir. Probleme ait özellikler; üs sayısı, üslerin konumları, talep dağılımı, uçak bilgileri ve hava savunma sistemlerinin konumları olarak ele alınmıştır. Bu özelliklerin değerleri ve bu değerlerin nasıl elde edildiği müteakip bölümlerde açıklanmıştır.

7.1. Deney Setinin Özellikleri

Üslere ait özellikler ve konumları, taleplerin yoğunluğu, taleplerdeki sapmalar ve uçak kapasitesi farklı problem tiplerinin oluşturulmasında kullanılmıştır. Bu verilere ait özellikler, algoritmaların farklı problem tiplerindeki performansının değerlendirilebilmesi için tasarlanmıştır. Deney setinin oluşturulmasında kullanılacak veriler ve bu verilere ait özellikler aşağıda açıklanmıştır.

7.1.1. Üslere ait özellikler ve konumları

AHTP'de her üsse ait; mevcut hava koşullarından oluşan bir hava koşulları risk değeri, pistin bombalanıp zarar görüp görmediğinin bilgisi ve o üsse alçak uçuş yapılmasının zorunlu olup olmadığı bilgileri vardır. Problem setlerinin oluşturulmasında her üs için hava koşulları risk değerleri (10, 20, 30, 40, 50, 60, 70, 80, 90, 100) kümesi içerisinden rastgele seçilmiştir. Bir üsse ait pistin hasar görüp görmediği %20'lik bir olasılıkla belirlenmiştir. Hangi üsse giderken veya gelirken alçak uçuşun zorunlu olup olmadığı %20'lik bir olasılıkla tespit edilmiştir. Üslerin konumlandırılmasında kümelenmiş ve rastgele olmak üzere iki alternatif yöntem kullanılmıştır. Bu iki durum Kartezyen uzayı içerisinde farklı dağılmış üslerin, algoritmaların çalışmasına olan etkisinin anlaşılması için tasarlanmıştır.

- **Rastgele (R):** Üsler koordinat düzleminde rastgele noktalara birbiri üzerine gelmeyecek şekilde yerleştirilmiştir.

- **Kümelenmiş (K):** Üsler belirli kümelerin elemanıdır ve bu kümeler düzlem üzerinde rastgele dağılmıştır. Kümelenmiş üslerin oluşturulmasında, ilk önce N problemdeki toplam üs sayısı olmak üzere $K = N/5$ tane küme merkezi sayısı belirlenmiştir. Küme merkezlerine ait konum koordinatları [0, 500] aralığından rastgele seçilmiştir. Daha sonra her bir küme merkezine N/K tane üs bağlanmıştır. Üslerin konumları bağlandığı küme merkezine [1, u] aralığında düzgün dağılacak uzaklıktaki koordinatlar olarak belirlenmiştir. Burada "u" küme merkezine maksimum uzaklık miktarını gösterir. "u" değeri büyüdükçe üsler birbirinden uzak ve dağınık, küçüldükçe üsler birbirine yakın ve sık kümeler oluşturur.

7.1.2. Talep yoğunluğu

Talep yoğunluğu, problemde kaç üssün birbirinden talepte bulunabileceğini belirler. Üslerin birbirlerinden olan taleplerinin sayısı ne kadar fazla ise talep yoğunluğu o kadar yüksektir. Yoğunlukların belirlenmesinde seyrek ve yoğun olmak üzere iki ayrı talep yoğunluk çeşidi tasarlanmıştır. Talep yoğunluğu arttıkça problemin boyutu da büyümektedir. Talep yoğunluğu değiştikçe uçakların kapasite kullanım karakteristiği de değişmektedir. Farklı talep yoğunlukları, farklı problem boyutları ve farklı uçak kapasite

kullanım karakteristikleri bakımından algoritmaların nasıl sonuçlar üreteceğinin test edilmesinde kullanılmıştır.

- **Seyrek talep (ST):** Bir üssün diğer üslerin kaçından talepte bulunabileceği [1, n/2] aralığında düzgün dağılımına göre belirlenir.

- **Yoğun talep (YT):** Bir üssün diğer üslerin kaçından talepte bulunabileceği [n/2, n] aralığında düzgün dağılımına göre belirlenir. Yoğun talebe sahip örneklerde problem boyutu büyümektedir. Ayrıca daha çok talebe cevap veren uçakların kapasite kullanım karakteristikleri de değişmektedir. Bu iki durum yoğun talebe sahip problem tiplerinin çözümünün daha zor olabileceğini göstermektedir.

7.1.3. Taleplerdeki sapmalar

Taleplere ait sapmalar üslere ait talep büyüklüklerinin nasıl değiştiğini belirler. Sapmalar arttıkça bir üssün talebi ile diğer üssün talebi arasındaki değişim miktarı artmaktadır. Taleplerdeki sapmaların artmasıyla bir problemde yüksek taleplere sahip üslerin sayısı da artmaktadır. Bu durum, bir uçağın kapasitesi dikkate alındığında iki yüksek talepli üsse hizmet verememesi olasılığını arttırmaktadır. Dolayısı ile yüksek sapmalar kapasite kullanımında değişikliklere neden olmaktadır. Farklı talep sapmalarının oluşturulması ile algoritmaların üreteceği sonuçların nasıl etkileneceğinin değerlendirilebilmesi amaçlanmıştır.

- **Büyük sapmalar (BS):** Her üs için talep miktarları [1, 50] aralığında düzgün dağılımından elde edilir.

- **Küçük sapmalar (KS):** Her üs için talep miktarları [1, 25] aralığında düzgün dağılımından elde edilir.

7.1.4. Uçak kapasitesi

Bir problemdeki uçak sayısı üslere ait taleplerin toplamına göre belirlenmiştir. Problemdeki üslere ait tüm taleplerin toplamının, uçaklara ait toplam kapasitenin altında

kalması sağlanmıştır. Öncelikle üs talepleri oluşturulmuştur. Oluşturulan bu taleplerin tümünün toplamı o probleme ait toplam talebi verir. Toplam uçak kapasitesinin %75'i alındığında elde edilen değer, toplam talepten küçük ise probleme yeni bir uçak eklenmiştir. Bu işleme toplam uçak kapasitesinin %75'i alındığında elde edilen değer, toplam talepten büyük oluncaya kadar devam edilir. Eklenen uçak tipi uçaklar kümesinden rastgele seçilmiştir. Örneğin probleme ait taleplerin toplamı 70 birim olsun. Uçakların kapasitelerinin birbirine eşit ve 50 birim olduğu varsayılsın. Bu problemde 1 adet uçak kullanıldığında, toplam uçak kapasitesi olan 50'nin %75'i alındığında 37,5 birim bulunur ve bu değer toplam talep olan 50'den küçüktür. Bu yüzden probleme bir adet uçak daha eklenir. Bu durumda toplam uçak kapasitesi 100 birim olur ve %75'i alındığında 75 birim olarak bulunur. Bu değer, 70 birim olan toplam talepten büyüktür ve problem iki uçaklı bir problem olarak üretilir.

7.2. Örnek Problemlerin Oluşturulması

Örnek problemler MATLAB programında kodlanarak oluşturulmuştur. Uçak tipleri ve özellikleri, hava savunma sistemlerinin konumu sabit olarak alınmıştır. Buna rağmen her bir özelliğe göre birden fazla problem üretilmesi durumunda çok büyük örnek sayıları ortaya çıkmaktadır. Problemlerin oluşturulmasında, problemin karakteristiğini değiştireceği değerlendirilen dört ana özellik; üs sayısı, üslerin konum yapısı, talep yoğunluğu ve talep sapmaları ele alınmıştır.

Bu özelliklerin iki düzeyi için problemler türetilmiştir. 2^4=16 adet farklı boyutta ve tipte problem ve her bir problem tipi ve boyutundan 5 adet olmak üzere toplam 16x5=80 adet problem üretilmiştir. 80 adet problem sayısı, yeterli çeşitlendirme sağlayacak kadar çok; çözülüp detaylı raporlanabilecek kadar da küçüktür. Türkiye için gerçek problem boyutlarına yakın olan 8 ve 12 adet üs problem boyutu olarak alınmıştır. Deney setlerine ait özellikler Çizelge 7.1'de gösterilmiştir. Çizelge 7.1'de "N" üs sayısını, "R" ve "K" rastgele ve kümeli olarak konumlandırılmış üsleri, "ST" seyrek talepleri, "YT" yoğun talepleri, "BS" büyük sapmaları, "KS" küçük sapmaları ifade eder.

Çizelge 7.1 Problem Tiplerine Ait Özellikler

N=8				N=12			
1	**R**	**ST**	**BS**	9	**R**	**ST**	**BS**
2			**KS**	10			**KS**
3		**YT**	**BS**	11		**YT**	**BS**
4			**KS**	12			**KS**
5	**K**	**ST**	**BS**	13	**K**	**ST**	**BS**
6			**KS**	14			**KS**
7		**YT**	**BS**	15		**YT**	**BS**
8			**KS**	16			**KS**

N= Üs Sayısı, R=Rastgele, K=Kümelenmiş, ST=Seyrek Talep, YT=Yoğun Talep, BS=Büyük Sapma, KS=Küçük Sapma, DK=Dar Kapasite, GK=Geniş Kapasite

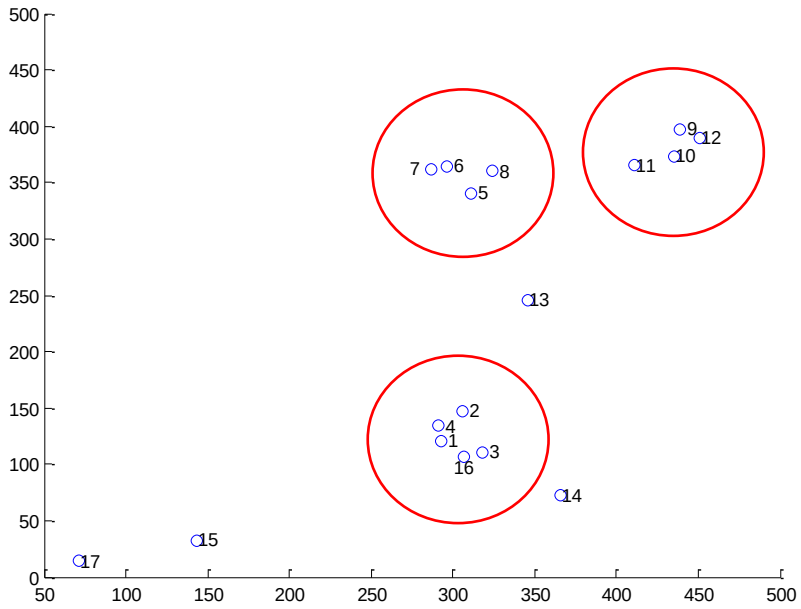
Oluşturulan deney seti http://dx.doi.org/10.17632/fr354t5jbb.1 internet adresinde verilmiştir (Aksoy ve Kapanoğlu, 2018b) ve Askeri Hava Taşıma Probleminin deney setleri olarak literatüre kazandırılmıştır. Deney setleri talep yapısı değiştirilerek farklı türdeki araç rotalama problemleri içinde kullanılabilir.

8. BULGULAR VE TARTIŞMA

Bu bölümde AHTP'nin çözümü için önerilen hGA'nın nasıl çalıştığı seçilen bir problem üzerinden anlatımı, oluşturulan test problemlerinin hGA ve kGA ile çözümünden elde edilen sonuçlar ve bu sonuçlara göre hGA'nın ve kGA'nın değerlendirilmesi yapılmıştır. hGA'nın ve kGA'nın karşılaştırması yapılırken her iki algoritmanın koşulları eşit tutulmuştur. Her iki algoritmaya ait popülasyon büyükleri, jenerasyon sayıları eşit alınmıştır.

8.1. hGA'nın Ürettiği Çözümlerin İncelenmesi

Deney setlerindeki örnek problemlerden 63 numaralı problem, hGA'nın nasıl çalıştığının ve ürettiği çözümlerin anlatılabilmesi için seçilmiştir. 63-numaralı problem 12 adet kümelenmiş üsse, yoğun ve büyük sapmalı taleplere sahip bir problem tipidir. Bu problem sahip olduğu özelliklerle çözülmesi daha zor bir problem olup hGA'nın çözüm üretme kapasitesinin gösterilmesi için ele alınmıştır. Üslerin ve hava savunma sistemlerinin (13, 14, 15, 16 ve 17 numaralı noktalar) koordinat düzlemi üzerindeki gösterimi Şekil 8.1'de verilmiş olup bu problemde 3 adet küme oluşturmuştur. Problemdeki uçaklara ait özellikler Çizelge 8.1'de verilmiştir.



Şekil 8.1 Üslerin ve Hava Savunma Sistemlerinin Konumları

Çizelge 8.1 Uçaklara Ait Özellikler

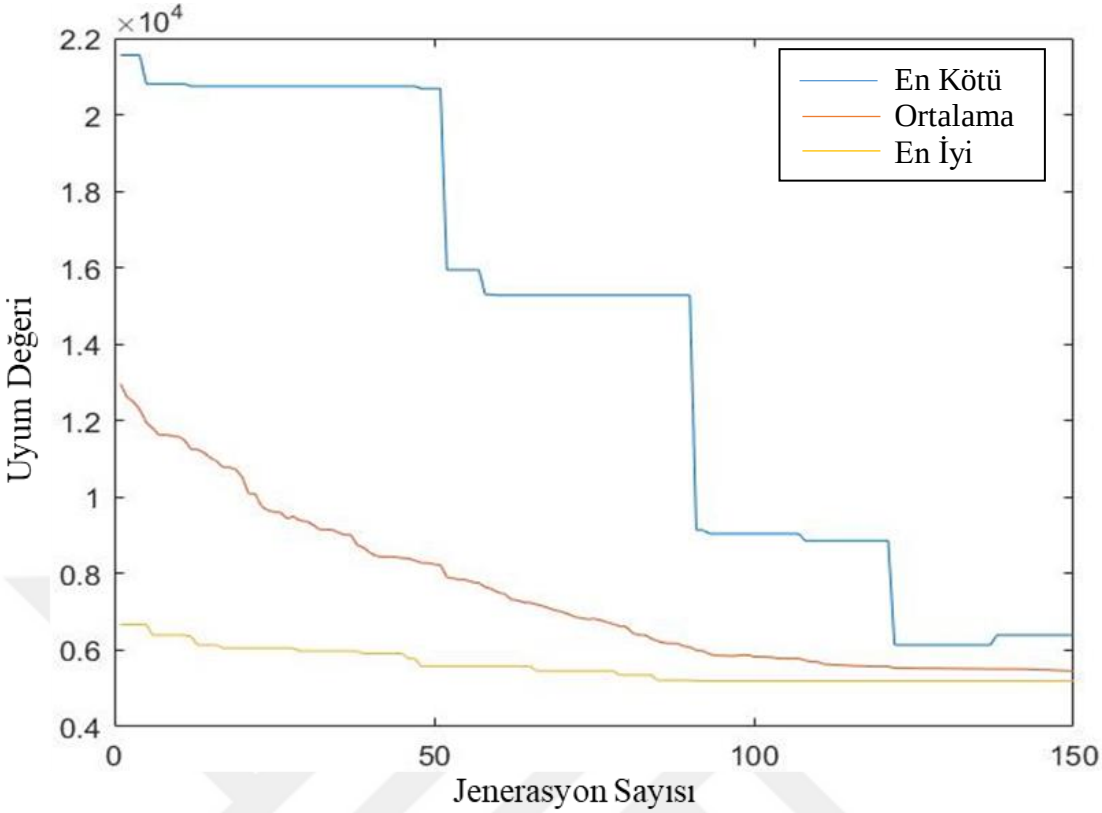
Uçak No	Hız (K)	Kapasite (Kg)	Birim Maliyet ($/saat)	Ana Üs	Menzil (NM)
1	275	1500	4000	1	1850
2	300	2500	6000	1	2160
3	275	1500	4000	1	1850
4	300	2500	6000	1	2160

Üslere ait talep matrisi Çizelge 8.2'de verilmiştir. Problemde 12 üs olmasına rağmen taşınması gereken 100 adet talep vardır. Problem hGA ve kGA ile çözülmüştür.

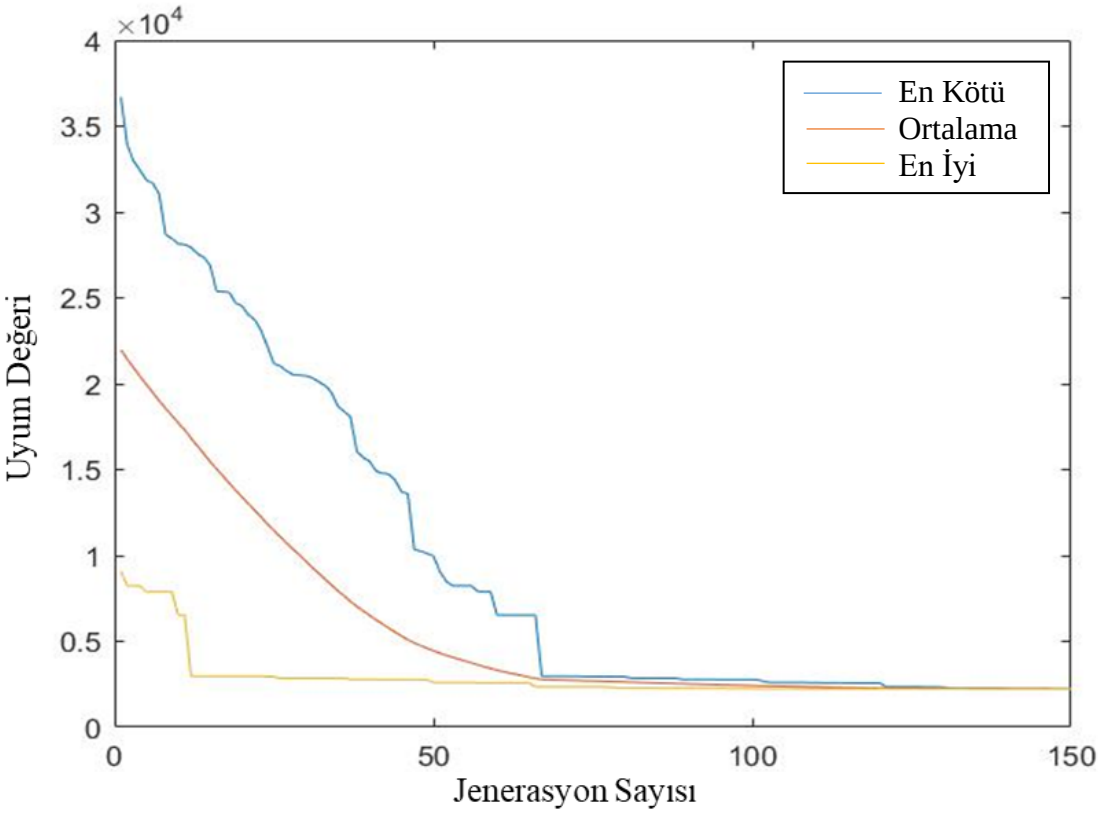
Çizelge 8.2 *63*-Numaralı Probleme Ait Talep Matrisi

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	-	8	41	-	36	-	46	47	33	7	48	47
2	45	-	49	-	14	11	-	-	19	35	36	31
3	45	-	-	-	-	46	15	23	25	-	45	28
4	26	-	28	-	1	13	41	42	39	21	-	4
5	38	-	32	-	-	40	-	14	18	-	43	20
6	38	2	21	-	17	-	10	50	31	35	35	27
7	-	47	45	1	9	10	-	39	13	40	32	38
8	1	-	38	-	30	-	23	-	9	23	-	31
9	-	-	-	31	11	46	-	48	-	5	37	-
10	47	-	27	49	47	26	50	13	41	-	30	19
11	36	24	17	-	17	-	-	35	-	17	-	10
12	-	18	35	20	6	36	20	24	22	48	14	-

Jenerasyonlar boyunca bulunan en iyi ve en kötü çözüm, popülasyondaki bireylerin uyum değerlerinin ortalamasının grafiği hGA için Şekil 8.2'de, kGA için Şekil 8.3'de verilmiştir. hGA'da evrimin daha yavaş olduğu ve kGA'da ise 60'ıncı jenerasyondan sonra tüm popülasyonun bulunan en iyi değere yakınsadığı görülmüştür. Bu beklenen bir sonuçtur. hGA komşuluk yapısıyla çözüm uzayı içerisinde daha yoğun bir arama yaparken kGA kısa zamanda bulduğu lokal en iyiye yakınsamaktadır.

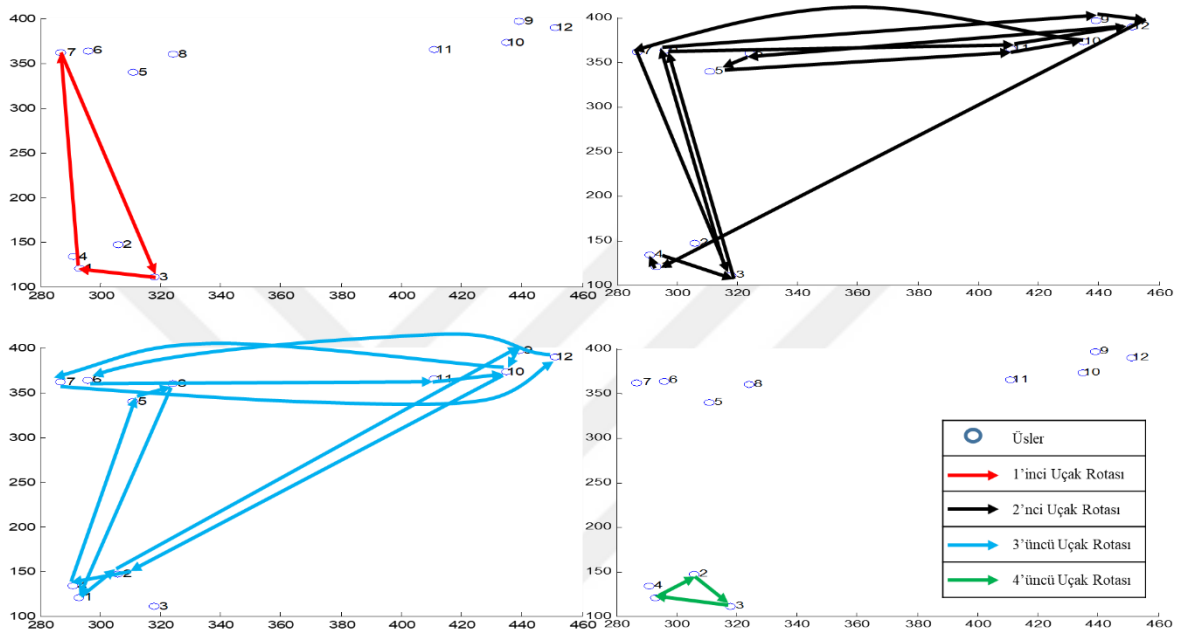


Şekil 8.2 hGA ’da Popülasyonun Evrimi Grafiği

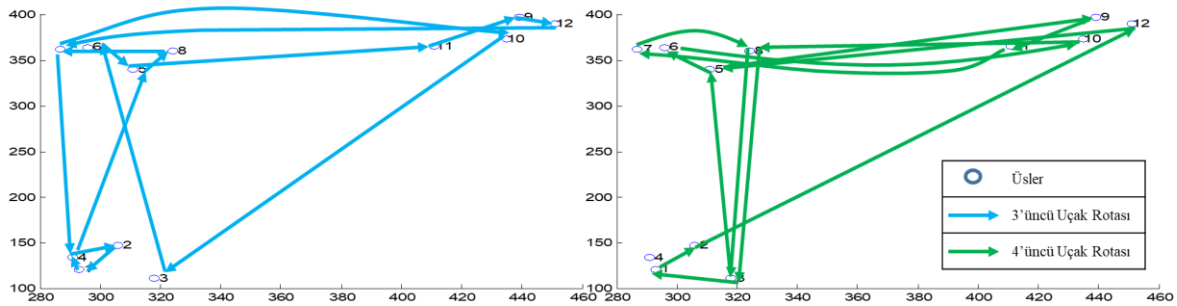


Şekil 8.3 kGA ’da Popülasyonun Evrimi Grafiği

hGA tarafından üretilen çözümlerde ara jenerasyonda elde edilen en iyi çözüm Şekil 8.4'de, son jenerasyonda elde edilen en iyi çözüm Şekil 8.5'de gösterilmiştir. Bu şekillerde üsler ve konumları Kartezyen düzlemi içerisinde 1'den 12'ye kadar numaralandırılarak, her uçak ve bu uçaklara ait rotalar ise farklı renkli oklarla gösterilmiştir. hGA'da 4 uçaktan oluşan ara en iyi çözüm, algoritma içinde 2 uçaklı daha iyi bir çözüme evirilmiştir.



Şekil 8.4 Ara Jenerasyonda Erişilen En İyi Çözüm (Kullanılan uçak sayısı:4)

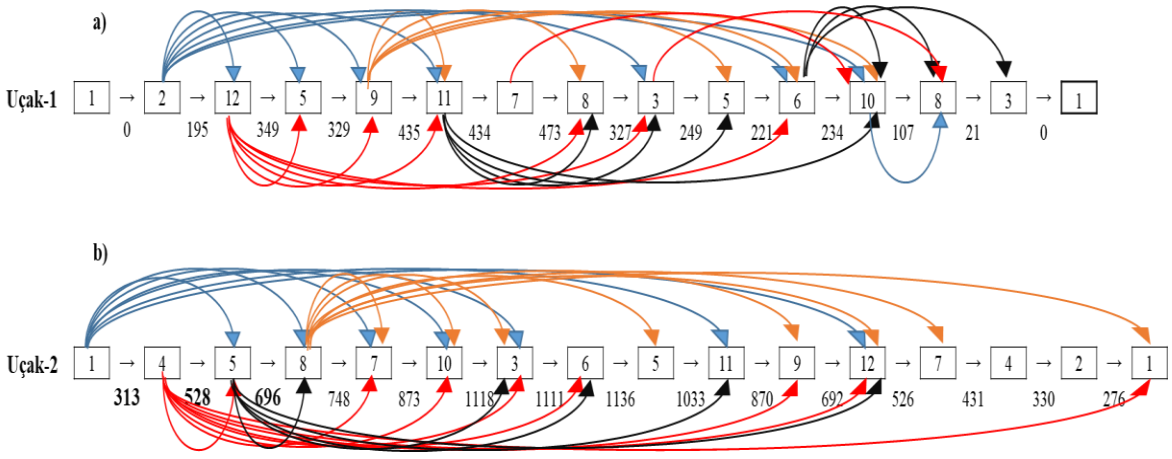


Şekil 8.5 Son Jenerasyonda Erişilen En İyi Çözüm (Kullanılan uçak sayısı:2)

AHTP'ye ait bir hGA çözümü; uçakların üsleri hangi sıra ile dolaştığı, hangi uçağın hangi talebin ne kadarını hangi sıra ile taşıdığı, uçaklara yüklemelerin hangi sıra ile yapıldığı bilgilerini içerir.

Üsler arasındaki taleplerin uçak rotası içinde nasıl taşındığının bir örneği Şekil 8.6'da ilk rotada taşınan taleplerin miktarları ve yüklenme sıraları Çizelge 8.3'de verilmiştir. 1'inci uçağa ait rota Şekil 8.6 a)'da, ikinci uçağa ait rota Şekil 8.6 b)'de gösterilmiştir. İlk uçağa ait rota incelendiğinde, uçak ana üssünden boş olarak kalkmıştır çünkü ana üsten taşınacak bütün talepler ikinci rotada taşınmıştır. Bu durumla gündelik hayatta da karşılaşılabilmektedir. Daha sonra uçak *2–numaralı* üsse inmiş ve buradan taşınan 49+14+11+19+35+36+31=195 kg malzemeyi yükleyerek *12–numaralı* üsse gitmiştir. *12–numaralı* üste, *2–numaralı* üsten gelen 31 kg malzeme indirilip *12–numaralı* üsten gitmesi gereken 35+6+36+24+22+78+14=185 kg malzeme yüklenmiştir. Uçağın *5–numaralı* üsse giderken taşıdığı yük miktarı denklem (6.3)'den "195–31=GiÜ–185", GiÜ=195+185–31=349 kg olarak hesaplanabilmektedir.

Uçakların rotası üzerinde her bacakta taşıdığı yük miktarları bu şekilde hesaplanmaktadır. Eğer bir üsten gitmesi gereken taleplerin toplamı uçağın mevcut kapasitesini geçiyorsa, kapasite koşulu sağlanana kadar ya taşınan talepler uçağa yüklenmemekte ya da taşınan en büyük talebin taşınabilen kısmı uçağa yüklenmekte yani talep bölme işlemi yapılmaktadır.



Şekil 8.6 Taleplerin Nasıl Taşındığına Dair Bir Örnek

Çizelge 8.3 Uçak-1 Rotasında Taşınan Miktarlar ve Sıraları

1.	2→3: 49	11.	12→8: 24	21.	11→5: 17
2.	2→5: 14	12.	12→9: 22	22.	11→8: 35
3.	2→6: 11	13.	12→10: 48	23.	11→10: 17
4.	2→9: 19	14.	12→11: 14	24.	7→8: 39
5.	2→10: 35	15.	9→5: 11	25.	3→8: 23
6.	2→11: 36	16.	9→6: 46	26.	6→3: 21
7.	2→12: 31	17.	9→8: 48	27.	6→8: 50
8.	12→3: 35	18.	9→10: 5	28.	6→10: 35
9.	12→5: 6	19.	9→11: 37	29.	10→8: 13
10.	12→6: 36	20.	11→3: 17		

Şekil 8.6 (a)'daki ilk rotadaki yani 1'inci uçağa ait taşımaların tamamı olan 29 adet talep gösterilebilmişken, (b)'deki 2'nci uçağa ait rotada toplam 79 adet olan taleplerin bir kısmı gösterilebilmiştir. Görüldüğü üzere kimi talepler alındıktan sonra hemen talebi yapan üsse götürülebilmişken kimi talepler neredeyse tüm rota boyunca uçak üzerinde taşınmıştır. Örneğin *2–numaralı* üsten *10–numaralı* üsse gitmesi gereken 35 kg malzeme, rotanın ilk bacağından sonra uçağa yüklenmiş ve 11'inci bacağında *10–numaralı* üsse inene kadar uçak üzerinde taşınmıştır. Bunun tam tersi olarak *2–numaralı* üsten *12–numaralı* üsse gitmesi gereken 31 kg malzeme, rotanın ilk bacağından sonra uçağa yüklenmiş ve 2'inci bacağında *12–numaralı* üsse inilerek bu üsse bırakılmıştır. İlk örnekte ara duraklı taşıma yapılmışken ikinci örnekte direk taşıma gerçekleştirilmiştir.

8.2. Test Problemlerinin Çözümleri

Deney setinde toplam 80 adet problem vardır. Bu 80 adet problem 16 farklı tipten 5'er adet problem üretilmesi ile elde edilmiştir. 80 adet problem hem hücresel genetik algoritma hem de klasik genetik algoritma ile beşer defa çözülmüştür. Bu çözümler sonucunda elde edilen en iyi değerler hGA ve kGA için Çizelge 8.4'de verilmiştir.

Her iki algoritmanın en iyi değerleri karşılaştırıldığında 80 problemin 33'ünde hGA, 25'inde kGA daha iyi sonuçlar üretmiş ve 22'sinde her iki algoritmada aynı sonuca ulaşmıştır.

Çizelge 8.4 hGA ve kGA İle Elde Edilen En İyi Değerler

	hGA	kGA	% Fark
1	6395,14	6339,83	0,01
2	5230,66	5371,87	-0,03
3	2058,67	2246,88	-0,09
4	2173,79	2371,04	-0,09
5	724,26	726,45	0,00
6	956,39	1617,11	-0,51
7	968,17	1586,22	-0,48
8	824,34	824,34	0,00
9	6788,73	3716,2	0,58
10	3917	3956,99	-0,01
11	3603,25	8353,48	-0,79
12	8438,34	8956,74	-0,06
13	3296,68	2341,25	0,34
14	3335,86	2253,15	0,39
15	4373,02	3641,51	0,18
16	1725,78	3015,43	-0,54
17	1429,14	1429,14	0,00
18	1754,71	1754,71	0,00
19	2350,29	2372,75	-0,01
20	2098,5	2282,71	-0,08
21	5619,67	5619,67	0,00
22	655,12	655,12	0,00
23	2009,94	2957,66	-0,38
24	1586,72	910,52	0,54
25	8976,62	9282,06	-0,03
26	9281,42	9157,17	0,01
27	8995,4	7238	0,22
28	8041,11	5166,46	0,44
29	1854,91	2367,27	-0,24
30	1025,25	1058,01	-0,03
31	3790,22	2879,52	0,27
32	1691,72	1445,44	0,16
33	6311,05	6311,05	0,00
34	1741,54	1741,54	0,00
35	2361,9	2504,03	-0,06
36	1904,71	1904,71	0,00
37	523,39	594,62	-0,13
38	701	641,84	0,09
39	995,98	995,98	0,00
40	6069,27	6069,27	0,00

	hGA	kGA	% Fark
41	5216,1	5206,43	0,00
42	8033,59	7868,62	0,02
43	4089,41	4547,55	-0,11
44	7794,8	8097,04	-0,04
45	1799,52	1480,04	0,19
46	6334,86	6898,08	-0,09
47	4775,52	2928,47	0,48
48	2347,06	1993,83	0,16
49	1302,5	1302,5	0,00
50	6764,04	6764,04	0,00
51	7365,14	7365,61	0,00
52	1707,01	1707,01	0,00
53	522,92	522,92	0,00
54	854,31	854,31	0,00
55	5493,6	5493,6	0,00
56	855,75	855,75	0,00
57	7327,72	7557,66	-0,03
58	7708,69	7503,91	0,03
59	3494,71	8309,05	-0,82
60	7765,6	7810,83	-0,01
61	8978,99	9178,72	-0,02
62	3305,78	3642,84	-0,10
63	3051,04	2304,32	0,28
64	7059,71	7050,69	0,00
65	2647,82	2628,12	0,01
66	1966,22	1966,22	0,00
67	10351,54	10454,6	-0,01
68	1811,62	1811,62	0,00
69	517,12	517,12	0,00
70	6104,6	6104,6	0,00
71	599,51	616,62	-0,03
72	5342,7	5342,7	0,00
73	4105,9	3707,63	0,10
74	6105,8	6084,36	0,00
75	3895,12	4408,13	-0,12
76	3538,73	7892,28	-0,76
77	2623,02	2676,72	-0,02
78	7471,26	7368,99	0,01
79	2451,23	1865,82	0,27
80	2299,4	4457,76	-0,64

Elde edilen sonuçlar arasındaki % fark denklem (8.1) ile hesaplanmıştır. Denklem (8.1)'deki negatif bir değer hGA'nın, pozitif bir değer kGA'nın daha iyi sonuçlar ürettiğini gösterir. İki algoritmanın en iyi sonuçları arasındaki yüzde farkın toplamı hGA lehine %1,59 olarak bulunmuştur. Sadece hGA'nın iyi olduğu problemlerde yüzde farkın toplamı %6,36 olmuştur. Sadece kGA'nın iyi olduğu problemlerde yüzde farkın toplamı %4,79 bulunmuştur. Bu sonuç, hGA'nın iyi olduğu değerlerdeki farkın kGA'nın iyi olduğu değerlerdeki farktan daha fazla olduğunu yani optimuma daha yakın sonuçlar ürettiğini göstermektedir.

$$\%FARK = \frac{hGA - kGA}{\frac{hGA + kGA}{2}} \tag{8.1}$$

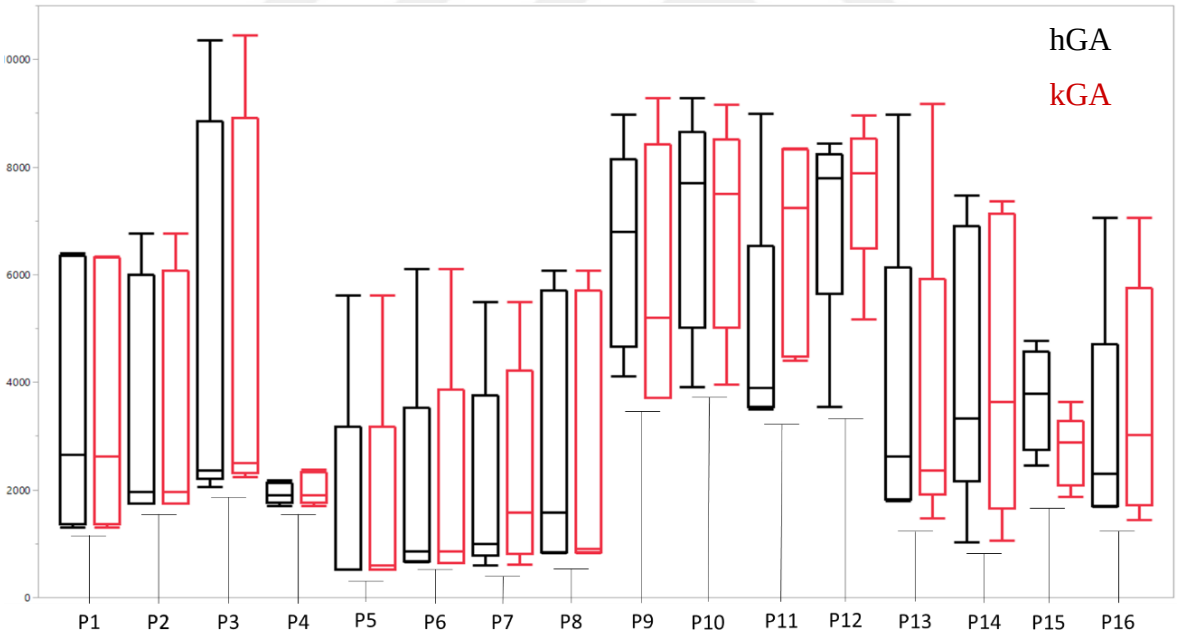
Problem tiplerine göre hangi algoritmanın kaç defa daha iyi çözümler ürettiği Çizelge 8.5'da verilmiştir. 8 üslü problemlerde hGA 14 kez kGA 4 kez daha iyi çözümler üretmiştir. 12 üslü problemlerde hGA 19 kez kGA 21 kez daha iyi çözümler üretmiştir. Buradan hGA'nın problem boyutunun görece küçük olduğu problemlerde çözüm uzayını daha iyi aradığı, görece büyük boyutlu problemlerde ise kGA ile benzer performans gösterdiği söylenebilir. Rastgele konumlandırılmış üslü problemlerde hGA 19 kez, kGA 11 kez, kümelenmiş üslü problemlerde hGA 14 kez, kGA 14 kez, seyrek talepli problemlerde hGA 13 kez, kGA 14 kez, yoğun talepli problemlerde hGA 20 kez, kGA 11 kez büyük sapmalı problemlerde hGA 19 kez, kGA 13 kez, küçük sapmalı problemlerde hGA 14 kez, kGA 12 kez daha iyi çözümler üretmiştir. Bu sonuçlardan hareketle, lokal minimumların daha çok olduğu yoğun talepli ve büyük sapmalı problem tiplerinde hGA'nın komşuluk yapısının bir getirisi olan yoğunlaşmış (intensification) arama kabiliyeti sayesinde daha iyi performans gösterdiği söylenebilir. Diğer problem tiplerinde birbirlerine yakın performans gösterdikleri gözlemlenmektedir.

Deney setinde 16 problem tipinde 5'er adet birbirinden farklı toplam 80 adet problem oluşturulmuştur. 16 problem tipinin hGA ve kGA ile çözülmesi sonucu elde edilen çözümlerin kutu grafikleri yan yana olmak üzere Şekil 8.7'de verilmiştir. Her iki algoritma birbirine benzer tablolar oluşturmakla beraber hGA'ya ait çözümlerin dağılımının kGA

çözümlerine göre daha az olduğu gözlenmektedir. 11, 12 numaralı problem tiplerinde hGA; 9, 15 numaralı problem tiplerinde de kGA görece daha iyi sonuçlar elde etmiştir.

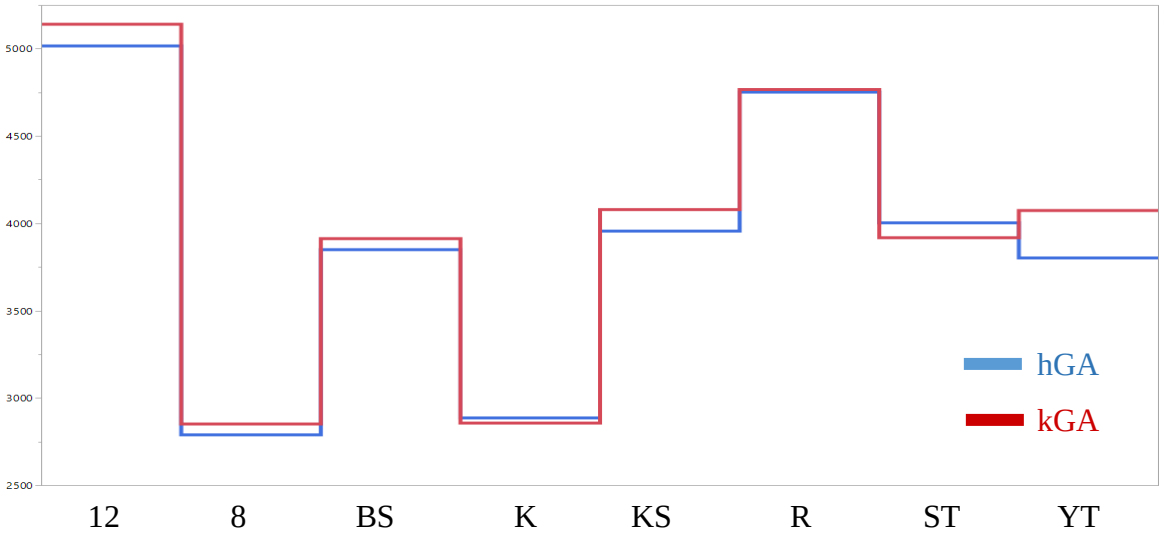
Çizelge 8.5 Problem Tipine Göre Karşılaştırma

	hGA	kGA	EŞİT
8 Üs	14	4	22
12 Üs	19	21	0
Rastgele	19	11	10
Kümeli	14	14	12
Seyrek Talep	13	14	13
Yoğun Talep	20	11	9
Büyük Sapma	19	13	8
Küçük Sapma	14	12	14



Şekil 8.7 Problem Tipi Bazında hGA ve kGA Karşılaştırması

Deney setinde 8, 12, Büyük Sapma (BS), Kümelenmiş (K), Küçük Sapma (KS), Rastgele (R), Seyrek Talep (ST), Yoğun Talep (YT) olmak üzere 8 ayrı özelliğe sahip problem tipi bulunmaktadır. Her problem tipinde algoritmalardan elde edilen sonuçların ortalaması alınmış ve bu ortalamaların algoritma bazındaki grafiği Şekil 8.8'de verilmiştir.



Şekil 8.8 Problem Ozelliği Bazında Karşılaştırma

Görüldüğü üzere hGA seyrek talep özelliği dışında diğer bütün problem özelliklerinde daha iyidir. Özellikle yoğun talep olan problemlerde hGA açık ara daha iyi sonuçlar elde etmiştir. Burada elde edilen sonuçlar Çizelge 8.5'dan elde edilen sonuçlarla tutarlılık göstermektedir.

9. SONUÇ VE ÖNERİLER

Bu çalışmada araç rotalama probleminin, yeni ve aynı zamanda daha zor bir varyantı olan askeri hava taşıma problemi tanımlanmıştır. AHTP askeri üsler arasındaki taşımaların nasıl yapılacağını birçok ek kısıt altında, uçuş risklerini göz önünde tutarak planlar. Askeri üsler hem bir talep hem de bir depo noktası olabilmektedir. Bir üs talep noktası olarak diğer üslerden ihtiyaç duyduğu malzemeleri talep eder, depo noktası olduğunda ise diğer üslerin ihtiyaçlarını karşılar. AHTP'nin öne çıkan özellikleri şu şekildedir:

i. Çok depolu,
ii. Farklı kapasitelere, menzile, maliyet ve hıza sahip heterojen uçaklar,
iii. Eş zamanlı toplama ve dağıtma yapılması,
iv. Taleplerin bölünebilir olması,
v. Ara duraklara uğranabilir olması,
vi. Maksimum görev süresi,
vii. Maksimum uçuş bacağı sayısı,
viii. Bir üsse **aynı uçağın** birden fazla kez uğrayabilmesi,
ix. Bir üssün birden fazla **farklı uçaklarla** ziyaret edilebilmesi,
x. Hasarlı pistlerin kullanılmaması,
xi. Alçak irtifa uçuş için limitlerin olması,
xii. Rota farklılaştırma yapılması,
xiii. Hava savunma sistemlerine olan yakınlık,
xiv. Hava koşullarının etkilerinin göz önünde bulundurulması.

AHTP'nin x, xi, xii, xiii ve xiv numaralı özellikleri problemin askeri boyutundan kaynaklanan özelliklerdir. Bu özellikler savaş veya gerginlik zamanlarında hava taşımalarının emniyet ile yapılabilmesini hedeflemektedir. Bu özellikler birer risk faktörü olarak tanımlanmıştır. Her risk faktörü için maksimum kabul edilebilir risk değerleri belirlenmiştir. Bu değerler planlayıcı tarafından, rotaların oluşturulmasında göze alınacak risk miktarının belirlenmesine olanak sağlamaktadır.

Risk faktörlerinden hava koşullarına ilişkin risk değeri bir bulanık kontrolcü ile modellenmiştir. Hava koşullarının iniş kalkışa etkileri konunun uzmanları olan pilotlar

tarafından değerlendirilmiş ve bu uzman görüşleri modele yansıtılmıştır. Bir üsse meteorolojik veriler bulanık kontrolcüye girdi olarak verilmiş ve çıktı olarak o üsse ait hava koşulları riski elde edilmiştir. Meteorolojik verilerin bir bulanık kontrolcü ile hava koşulları riskine dönüştürülmesi literatürde ilk defa bu çalışmada ele alınmıştır.

AHTP'nin daha iyi anlaşılabilmesi ve optimum çözümlerin elde edilebilmesi için bir karma tam sayılı matematiksel model önerilmiştir. Önerilen model AHTP'nin tüm özelliklerinin yanı sıra birden fazla ara durağa inebilme özelliğini de ele alabilmektedir. Birden fazla ara durağa inebilme özelliği literatürde çok az sayıdaki çalışmada bulunmakla birlikte matematiksel olarak ilk defa burada modellenmiştir. Matematiksel model mevcut bilgisayarların işlem gücü ile görece küçük boyutlu problemleri çözebilmektedir.

AHTP'nin çözümünde geliştirilen matematiksel model mevcut şartlarda yetersiz kalmıştır. Genetik algoritma AHTP gibi gerçek hayatta karşımıza çıkan, karmaşık problemlerin çözümünde oldukça başarılı bir algoritmadır. Hücresel ve klasik olmak üzere genetik algoritmaya ait iki varyantı AHTP'yi çözmek üzere geliştirilmiştir. Bu sayede her bir algoritmanın AHTP'yi çözmede ayrı ayrı performansı değerlendirilmiş hem de genetik algoritma çatısı altındaki iki farklı algoritmanın karşılaştırması yapılabilmiştir. Ayrıca genetik algoritmaların performansına etki eden parametreler tam faktöriyel tasarımla analiz edilmiş ve en iyi parametre tahminleri bulunmuştur. Her iki algoritmaya ait kodlar MATLAB ortamında geliştirilmiş ve koşturulmuştur.

Literatürde, yeni bir problem olan AHTP'ye ait deney setleri bulunmamaktadır. AHTP için 16 ayrı tipe ve 80 probleme sahip deney seti oluşturulmuştur. Önerilen deney seti sahip olduğu farklı özellikler ile kullanıcılara geliştirdikleri algoritmaları birçok açıdan test etmelerine imkân sağlamaktadır. AHTP için oluşturulan deney seti hem literatürde bu alandaki boşluğu doldurmuş hem de geliştirilen algoritmaların performans değerlendirmesinde kullanılmıştır.

Oluşturulan deney setindeki 80 problem hGA ve kGA ile beşer defa çözülmüştür. Elde edilen çözümler analiz edilerek hGA ile kGA arasında bir karşılaştırma yapılmıştır. Genel çözüm kalitesi değerlendirildiğinde hGA'nın daha iyi çözümler elde ettiği görülmüştür. Bunun yanında, lokal minimumların daha çok olduğu problem tiplerinde

hGA'nın komşuluk yapısının bir getirisi olan yoğunlaşmış (intensification) arama kabiliyetinin hGA'nın performansına olumlu katkıda bulunduğu gözlemlenmiştir.

Sonuç olarak, NP-Zor bir problem olan Askeri Hava Taşıma Problemi; askeri lojistik alanında kayda değer tasarruflar sağlayacak, lojistik planlayıcılar için risk alma veya riskten kaçınma yönünde seçim şansı verecek ve birçok kısıtı aynı anda göz önünde bulundurmalarını sağlayacak, bu konuda çalışma yapan araştırmacılara yeni ufuklar açacak bir araçtır.

Bu çalışmada uçağa malzeme yükleme ve uçaktan malzeme boşaltma işlemleri için sabit zamanlar ele alınmıştır. Fakat gerçek hayat için yükleme ve boşaltma için sabit zamanların kullanılmasının geçerli olmayacağı durumlar olabilir. Mevcut yaklaşım kolaylıkla, işlem görecek malzeme miktarı ile doğrusal olarak değişebilecek yükleme boşaltma zamanlarını kullanabilir hale getirilebilir. Bununla birlikte yükleme boşaltma zamanlarının işlem görecek malzeme miktarına göre stokastik varyans ile değişmesi durumu için önerilen yaklaşım düzenlenebilir.

Uçaklara yüklenen malzemeler için bu çalışmada kapasite kısıtı olarak uçakların taşıyabileceği maksimum kargo ağırlık kullanılmıştır. Kapasite kısıtı içerisine malzemelerin hacimleri, yoğunlukları vb. özellikler ele alınmamıştır. Patlayıcı malzemelerin taşınması, bazı malzemelerin sadece belirli uçak tipleri ile taşınabilmesi, personel ile malzemelerin aynı uçak içerisinde taşınmasına müsaade edilmemesi gibi durumlar ile gerçek hayatta karşılaşılabilmektedir. Buna benzer yükleme konfigürasyonları için farklı kısıtlar bu çalışmanın bir uzantısı olarak tasarlanabilir.

Zaman pencereleri birçok araç rotalama probleminde bulunan bir özelliktir. Bu çalışmada zaman penceresi olarak bir günlük süreç ele alınmıştır. Tüm taleplerin gün sonunda karşılanmış olması hedeflenmiştir. Fakat gerçek hayatta bazı malzemeler zamana duyarlı olabilmektedir. Bu tarz zamana duyarlı malzemelerin taşınmasında zaman pencereleri kullanılarak daha dar zaman pencereleri olan bir yaklaşım geliştirilebilir.

Ülkelerin barış ve güvenliği açısından askeri harcamalar ve yatırımlar kaçınılmaz ve tartışılmaz bir öneme sahiptir. Bu önem sebebiyledir ki 2018 yılında küresel askeri

harcamalar 1,7 trilyon dolar düzeyindedir. Gene 2015 yılına ait raporda Amerika Birleşik Devletleri savunma harcamalarına yaklaşık 600 milyar dolar harcarken eğitime 70 milyar dolar ayırmaktadır (Anonim, 2015) . Ülkelerin güvenlik kaygılarından taviz vermeyen ve savunma beklentilerini karşılayan, bilimsel yöntemlerle desteklenen daha ekonomik ve verimli askeri planlama, sevk ve idare çok ciddi tasarruflar sağlama potansiyeline sahiptir. Bu ve benzeri çalışmaların yaratabileceği tasarruflar sivil yaşamın iyileştirilmesine, eğitime ve sağlığa sunulacak yeni ekonomik kaynaklar ortaya koyabilir.

KAYNAKLAR DİZİNİ

Abdoun, O., Abouchabaka, J., Tajani, C. (2012), Analyzing the performance of mutation operators to solve the travelling salesman problem, International Journal of Emerging Sciences, 2, 1, 61-77

Akgun, V., Erkut, E., Batta, R. (2000), On finding dissimilar paths, European Journal of Operational Research, 121, 232-246

Aksoy, Ö., Kapanoglu, M. (2012), Multi-commodity, multi-depot, heterogenous vehicle pickup and delivery problem for air transportation in the turkish air force, Journal of Aeronautics and Space Technologies, 5, 4, 53-57

Aksoy, Ö., Kapanoğlu, M. (2014), Fuzzy risk constrained airlift routing with split, multi-way p&d policy in turkish air force, CIE 2014 - 44th International Conference on Computers and Industrial Engineering Proceedings, 1519-1529

Aksoy, Ö., Kapanoğlu, M. (2018a), A cellular and classic genetic algorithm matlab code for risk-oriented scheduling and routing for military air transportation problem, Mendeley Data, http://dx.doi.org/10.17632/ddx6gjy7fy.1

Aksoy, Ö., Kapanoğlu, M. (2018b), Military airlift on demand with flight risks benchmark instances, Mendeley Data, http://dx.doi.org/10.17632/fr354t5jbb.1

Alba, E., Dorronsoro, B. (2006), Computing nine new best-so-far solutions for capacitated vehicle routing problem with a cellular genetic algorithm, Information Processing Letters, 98, 6, 225-230

Alba, E., Dorronsoro, B. (2008), Cellular genetic algorithms, Springer, p. 9,13-20.

Alba, E., Dorronsoro, B., Giacobini, M., Tomassini, M. (2005), Decentralized cellular evolutionary algorithms, 7, 103-120

Alba, E., Giacobini, M., Tomassini, M., Romero, S. (2002), Comparing synchronous and asynchronous cellular genetic algorithms, 2439, 601-610

Anonim, 2015, Federal spending: Where does the money go, https://www.nationalpriorities.org/budget-basics/federal-budget-101/spending/, 01.10.2018.

KAYNAKLAR DİZİNİ (devam)

Baker, B. M., Ayechew, M. A. (2003), A genetic algorithm for the vehicle routing problem, Computers & Operations Research, 30, 5, 787-800

Berbeglia, G., Cordeau, J. F., Gribkovskaia, I., Laporte, G. (2007), Static pickup and delivery problems: A classification scheme and survey, Top, 15, 1, 1-31

Best, M. J. (2014), Relooking army aviation maintenance readiness reporting, U.S. Army War College,

Błażej, P., Wnętrzak, M., Mackiewicz, P. (2016), The role of crossover operator in evolutionary-based approach to the problem of genetic code optimization, Biosystems, 150, 61-72

Bole, A., Wall, A., Norris, A. (2014), Basic radar principles, p. 1-28.

Boyabatlı, O., Sabuncuoglu, I. (2004), Parameter selection in genetic algorithms, Journal of Systemics, Cybernetics and Informatics, 4, 2, 78-83

Braekers, K., Ramaekers, K., Van Nieuwenhuyse, I. (2016), The vehicle routing problem: State of the art classification and review, Computers & Industrial Engineering, 99, Supplement C, 300-313

Campbell, I. M. D. (2013), Construction heuristics for the airline taxi problem, Doktora tezi, University of the Witwatersrand, 211 p.

Chambers, L. (2001), The practical handbook of genetic algorithms, Chapman & Hall/CRC, p. 31-61, 236-251.

Chaturvedi, D. K. (2008), Soft computing: Techniques and its applications in electrical engineering, Springer, p. 223-293.

Cordeau, J.-F., Laporte, G., Potvin, J.-Y., Savelsbergh, M. W. P. (2007), Transportation on demand, Elsevier, p. 429-466.

KAYNAKLAR DİZİNİ (devam)

Diana, M., Dessouky, M. M. (2004), A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows, Transportation Research Part B: Methodological, 38, 6, 539-557

Dorronsoro, B. e., Alba, E. (2006), A simple cellular genetic algorithm for continuous optimization, 2838-2844.

Eksioglu, B., Vural, A. V., Reisman, A. (2009), The vehicle routing problem: A taxonomic review, Computers & Industrial Engineering, 57, 4, 1472-1483

Espinoza, D., Garcia, R., Goycoolea, M., Nemhauser, G. L., Savelsbergh, M. W. P. (2008), Per-seat, on-demand air transportation part 1: Problem description and an integer multicommodity flow model, Transportation Science, 42, 3, 263-278

FAA (2010), Weather-related aviation accident study 2003–2007, Federal Aviation Administration,

Fagerholt, K., Foss, B. A., Horgen, O. J. (2009), A decision support model for establishing an air taxi service: A case study, Journal of the Operational Research Society, 60, 9, 1173-1182

Foyle, D. C., Kaiser, M. K., Johnson, W. W. (1992), Visual cues in low-level flight: Implications for pilotage, training, simulation, and enhanced/synthetic vision systems, American Helicopter Society 48th Annual Forum, 1, 253-260.

Gleason, D. L. (2012), Airfield damage repair operations, U.S. Air Force, AFPAM 10-219, 117.

Harris, J. (2006), Fuzzy logic applications in engineering science, Springer, p. 1-11.

Hellendoorn, H., Thomas, C. (1993), Defuzzification in fuzzy controllers, Journal of Intelligent and Fuzzy Systems, 1, 109-123

Holland, J. H. (1975), Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence, University of Michigan Press, p. 1-20.

KAYNAKLAR DİZİNİ (devam)

J. J. van der Bruggen, L., K. Lenstra, J., Schuur, P. (1993), Variable-depth search for the single-vehicle pickup and delivery problem with time windows, Transportation Science, 27, 3, 298-311

Jebari, K., Madiafi, M. (2013), Selection methods for genetic algorithms, International Journal of Emerging Sciences, 3, 4, 333-344

Karakatič, S., Podgorelec, V. (2015), A survey of genetic algorithms for solving multi depot vehicle routing problem, Applied Soft Computing, 27, 519-532

Keysan, G. (2009), Tactical and operational planning for per-seat, on-demand air transportation, Doktora Tezi, Georgia Institute of Technology, 115 p.

Klir, G. J., Yuan, B. (1995), Fuzzy sets & fuzzy logic theory & applications, Prentice Hall PTR, p. 1-33.

Koç, Ç., Bektaş, T., Jabali, O., Laporte, G. (2016), Thirty years of heterogeneous vehicle routing, European Journal of Operational Research, 249, 1, 1-21

Koç, Ç., Laporte, G. (2018), Vehicle routing with backhauls: Review and research perspectives, Computers & Operations Research, 91, 79-91

Kora, P., Yadlapalli, P. (2017), Crossover operators in genetic algorithms: A review, International Journal of Computer Applications, 162, 10, 34-36

Lahyani, R., Khemakhem, M., Semet, F. (2015), Rich vehicle routing problems: From a taxonomy to a definition, European Journal of Operational Research, 241, 1, 1-14

Laporte, G. (2009), Fifty years of vehicle routing, Transportation Science, 43, 4, 408-416

Larrañaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., Dizdarevic, S. (1999), Genetic algorithms for the travelling salesman problem: A review of representations and operators, Artificial Intelligence Review, 13, 2, 129-170

KAYNAKLAR DİZİNİ (devam)

Lee, D. W., Bass, E. J., Patek, S. D., Boyd, J. A. (2008), A traffic engineering model for air taxi services, Transportation Research Part E: Logistics and Transportation Review, 44, 6, 1139-1161

Mee, R. (2009), A comprehensive guide to factorial two-level experimentation, Springer, p. 27-74.

Melanie, M. (1999), An introduction to genetic algorithms, MIT Press, p. 2-21.

Montoya Torres, J. R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., Herazo-Padilla, N. (2015), A literature review on the vehicle routing problem with multiple depots, Computers & Industrial Engineering, 79, 115-129

Parragh, S., Doerner, K., Hartl, R. (2008a), A survey on pickup and delivery problems: Part i: Transportation between customers and depot, Journal für Betriebswirtschaft, 58, 2, 81-117

Parragh, S. N., Doerner, K. F., Hartl, R. F. (2008b), A survey on pickup and delivery problems: Part ii: Transportation between pickup and delivery locations, Journal für Betriebswirtschaft, 58, 2, 81-117

Passino, K. M., Yurkovich, S. (1998), Fuzzy control, Addison Wesley, p. 10.

Pedrycz, W. (1993), Fuzzy control and fuzzy systems, John Wiley \ Sons, Inc., p.

Sarma, J., De Jong, K. (1996), An analysis of the effects of neighborhood size and shape on local selection algorithms, International Conference on Parallel Problem Solving from Nature, 236-244

Savelsbergh, M. W. P. (1995), The general pickup and delivery problem, Transportation Science, 29, 1, 17-29

Schönfisch, B., de Roos, A. (1999), Synchronous and asynchronous updating in cellular automata, Biosystems, 51, 3, 123-143

KAYNAKLAR DİZİNİ (devam)

Simoncini, D., Verel, S., Collard, P., Clergue, M. (2006), Anisotropic selection in cellular genetic algorithms, Genetic And Evolutionary Computation Conference, 559 - 566.

Soy, H. (2006), Gömülü sistem üzerinden bulanık kontrol ile ev otomasyonunun gerçekleştirilmesi, Yüksek Lisans tezi, Selçuk Üniversitesi, 96 p.

Toth, P., Vigo, D. (2001), The vehicle routing problem, Society for Industrial and Applied Mathematics, p. 109-126.

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., vd. (2017), New benchmark instances for the capacitated vehicle routing problem, European Journal of Operational Research, 257, 3, 845-858

Wang, L.-X. (1994), Adaptive fuzzy systems and control design and stability analysis, PTR Prentice Hall, p. 15-17.

Xiang, Z. H., Chu, C. B., Chen, H. X. (2006), A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints, European Journal of Operational Research, 174, 2, 1117-1139

Xu, H., Chen, Z.-l., Rajagopal, S., Arunapuram, S., Inc, M. (2001), Solving a practical pickup and delivery problem, Transportation Science, 37, 3, 255-364

Zabarankin, M., Uryasev, S., Murphey, R. (2006), Aircraft routing under the risk of detection, Naval Research Logistics (NRL), 53, 8, 728-747

EK AÇIKLAMALAR

Ek Açıklama-A: Pilotlardan Elde Edilen Verilerin Toplanmış Hali 94
Ek Açıklama-B: Üyelik Fonksiyonları ve Matlab Bulanık Kontrolcü Ara Yüzü..... 98
Ek Açıklama-C: IBM ILOG CPLEX Kodu..... 101
Ek Açıklama-D: hGA Matlab Kodu 111

Ek Açıklama-A: Pilotlardan Elde Edilen Verilerin Toplanmış Hali

GÖRÜŞ İÇİN TOPLAMLAR					
GÖRÜŞ(m)	**ÇOK DÜŞÜK**	**DÜŞÜK**	**NORMAL**	**İYİ**	**ÇOK İYİ**
100	37	0	0	0	0
200	36	1	0	0	0
300	32	4	1	0	0
400	24	12	1	0	0
500	18	17	2	0	0
600	7	21	9	0	0
700	4	24	9	0	0
800	3	22	11	0	1
900	1	20	13	2	1
1000	1	14	19	2	1
1100	0	5	26	5	1
1200	0	5	26	5	1
1300	0	4	26	6	1
1400	0	3	26	7	1
1500	0	3	23	10	1
1600	0	1	11	22	3
1700	0	1	10	23	3
1800	0	1	8	25	3
1900	0	1	8	24	4
2000	0	1	8	24	4
2200	0	0	1	21	15
2400	0	0	1	20	16
2600	0	0	0	20	17
2800	0	0	0	19	18
3000	0	0	0	18	19
3200	0	0	0	4	33
3400	0	0	0	3	34
3600	0	0	0	2	35
3800	0	0	0	2	35
4000	0	0	0	2	35
>4000	0	0	0	0	37

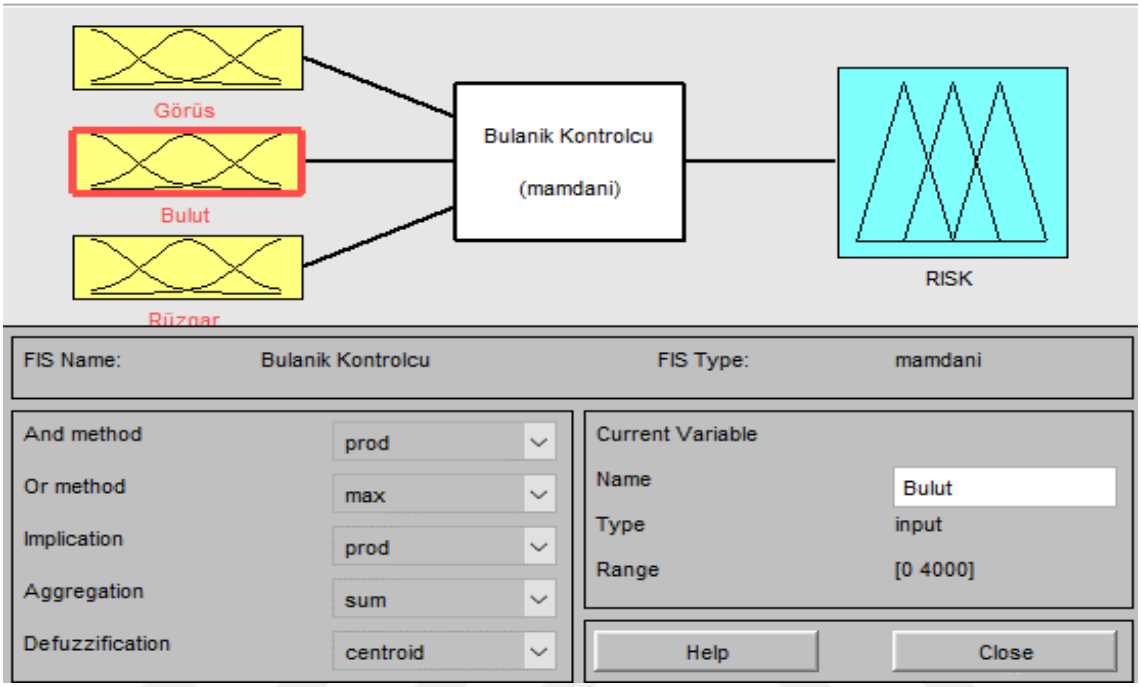
BULUT ALT TAVANI İÇİN TOPLAMLAR					
BULUT ALT TAVAN(ft)	**ÇOK DÜŞÜK**	**DÜŞÜK**	**NORMAL**	**İYİ**	**ÇOK İYİ**
100	37	0	0	0	0
200	33	4	0	0	0
300	22	15	0	0	0
400	18	16	3	0	0
500	13	19	5	0	0
600	4	24	9	0	0
700	3	21	12	1	0
800	3	19	14	1	0
900	2	16	15	3	1
1000	1	15	17	3	1
1100	1	7	18	9	2
1200	1	5	19	10	2
1300	1	3	20	11	2
1400	1	3	17	14	2
1500	1	2	16	15	3
1600	0	1	11	19	6
1700	0	1	11	19	6
1800	0	1	10	20	6
1900	0	1	10	19	7
2000	0	1	10	18	8
2200	0	0	4	17	16
2400	0	0	4	17	16
2600	0	0	3	16	18
2800	0	0	1	15	21
3000	0	0	1	13	23
3200	0	0	0	5	32
3400	0	0	0	4	33
3600	0	0	0	2	35
3800	0	0	0	1	36
4000	0	0	0	1	36
>4000	0	0	0	0	37

RÜZGÂR HIZI İÇİN TOPLAMLAR					
RÜZGÂR(K)	**ÇOK SAKİN**	**SAKİN**	**ORTA**	**ŞİDDETLİ**	**ÇOK ŞİDDETLİ**
1	37	0	0	0	0
2	37	0	0	0	0
3	37	0	0	0	0
4	37	0	0	0	0
5	36	1	0	0	0
6	25	12	0	0	0
7	14	23	0	0	0
8	12	25	0	0	0
9	8	27	2	0	0
10	8	26	3	0	0
11	1	23	13	0	0
12	1	22	14	0	0
13	1	18	16	2	0
14	1	15	18	3	0
15	1	14	19	3	0
16	1	1	22	13	0
17	0	0	23	14	0
18	0	0	22	14	1
19	0	0	18	18	1
20	0	0	18	18	1
21	0	0	6	23	8
22	0	0	5	23	9
23	0	0	3	25	9
24	0	0	3	23	11
25	0	0	3	21	13
26	0	0	0	15	22
27	0	0	0	14	23
28	0	0	0	14	23
29	0	0	0	12	25
30	0	0	0	12	25
31	0	0	0	3	34
32	0	0	0	3	34
33	0	0	0	3	34
34	0	0	0	2	35
35	0	0	0	2	35
36	0	0	0	0	37
37	0	0	0	0	37
38	0	0	0	0	37
39	0	0	0	0	37
40	0	0	0	0	37

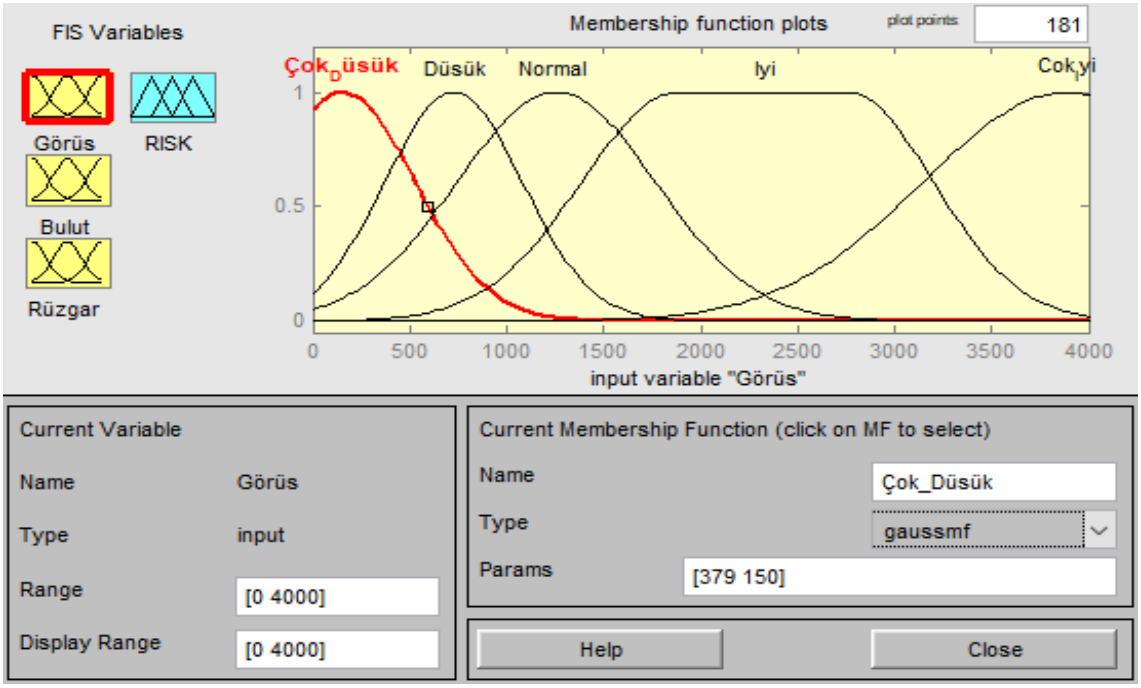
RİSK İÇİN TOPLAMLAR					
RİSK DEĞERİ	**RİSKSİZ**	**DÜŞÜK RİSKLİ**	**NORMAL RİSKLİ**	**RİSKLİ**	**YÜKSEK RİSKLİ**
2	36	0	0	0	0
4	35	1	0	0	0
6	30	6	0	0	0
8	28	8	0	0	0
10	23	13	0	0	0
12	8	22	6	0	0
14	8	22	6	0	0
16	8	21	6	1	0
18	7	20	6	3	0
20	7	19	6	4	0
22	4	14	13	4	1
24	4	13	14	4	1
26	3	12	14	5	2
28	2	11	16	5	2
30	2	9	17	6	2
32	1	6	18	9	2
34	0	6	18	9	3
36	0	5	17	11	3
38	0	5	16	12	3
40	0	4	15	14	3
42	0	4	14	14	4
44	0	3	14	15	4
46	0	2	14	16	4
48	0	2	12	18	4
50	0	1	12	18	5
52	0	0	6	17	13
54	0	0	6	17	13
56	0	0	6	17	13
58	0	0	6	16	14
60	0	0	6	15	15
62	0	0	3	14	19
64	0	0	3	14	19
66	0	0	3	14	19
68	0	0	1	13	22
70	0	0	1	12	23
72	0	0	0	11	25
74	0	0	0	10	26
76	0	0	0	9	27
78	0	0	0	8	28
80	0	0	0	8	28
82	0	0	0	3	33
84	0	0	0	3	33
86	0	0	0	0	36
88	0	0	0	0	36
90	0	0	0	0	36
92	0	0	0	0	36
94	0	0	0	0	36
96	0	0	0	0	36
98	0	0	0	0	36
100	0	0	0	0	36

Ek Açıklama-B: Üyelik Fonksiyonları ve Matlab Bulanık Kontrolcü Ara Yüzü

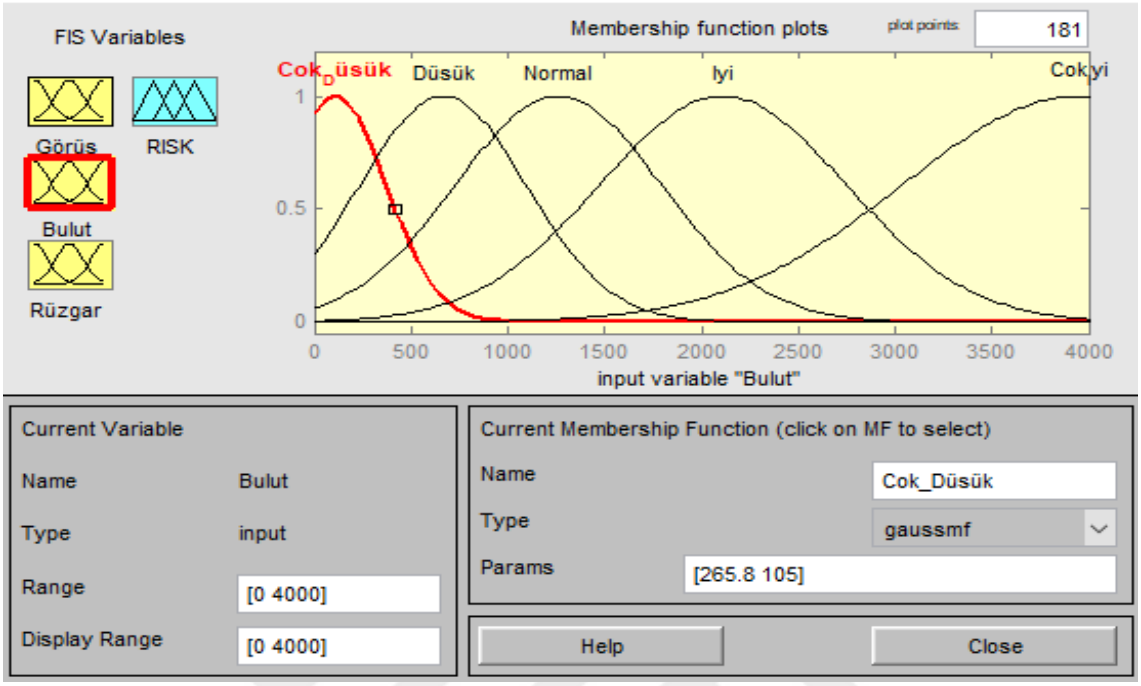
B.1. Matlab Bulanık Kontrolcü Ara Yüzü



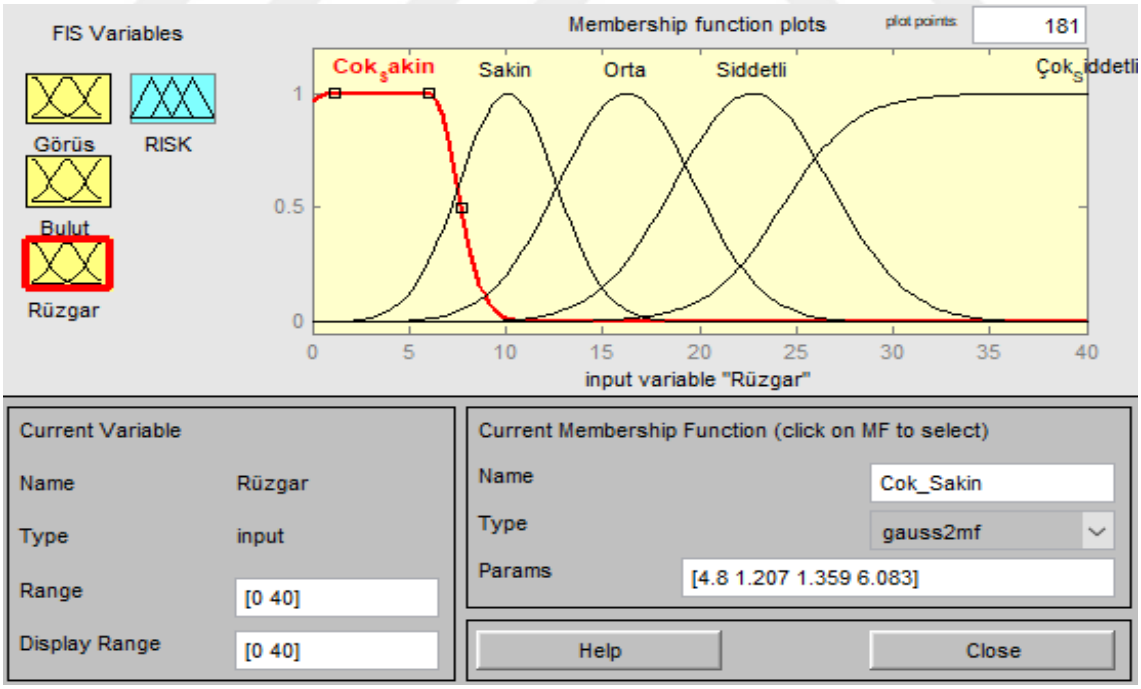
B.2. Görüş Bulanık Sözel Değişkeni Üyelik Fonksiyonları



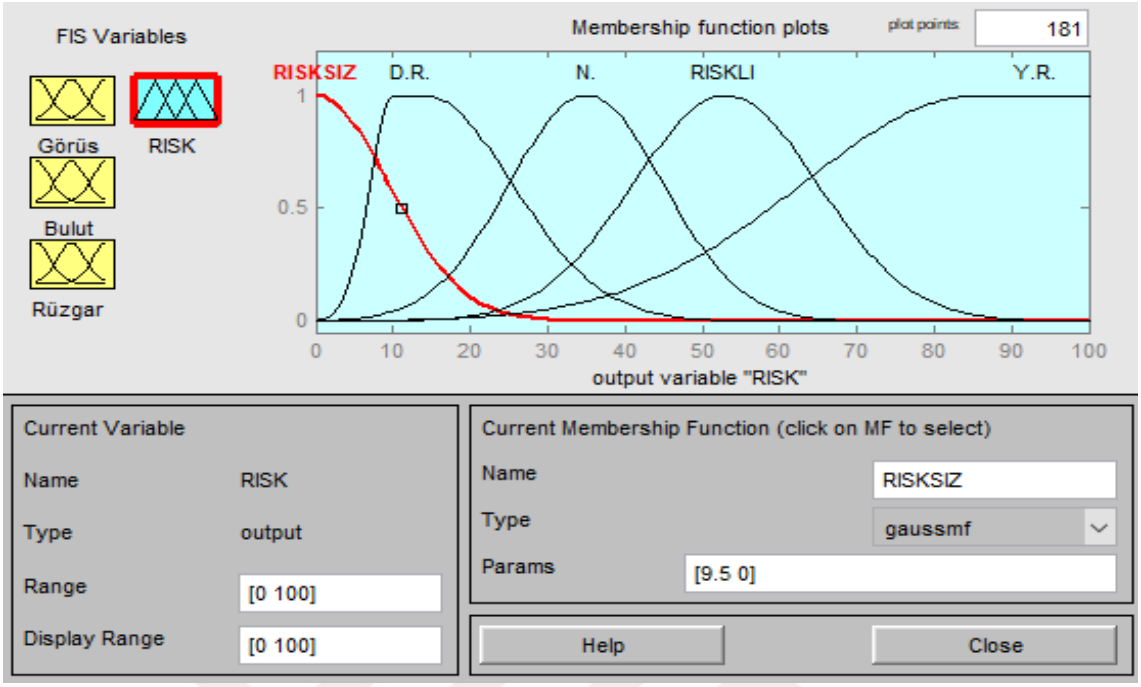
B.3. Bulut Alt Tavanı Bulanık Sözel Değişkeni Üyelik Fonksiyonları



B.4. Rüzgâr Bulanık Sözel Değişkeni Üyelik Fonksiyonları



B.5. Risk Bulanık Sözel Değişkeni Üyelik Fonksiyonları



Ek Açıklama-C: IBM ILOG CPLEX Kodu

C.1. Model Dosyası

```
/*********************************************
 * OPL 12.8 Model
 * Author: OZGUR

int sehirsayisi=...;
int aracsayisi=...;
int anaussayisi=...;
int risksayisi=...;
int savunmasayisi=...;
int fac=prod(i in 1..(sehirsayisi-2)) i;
range sehirler=(1..sehirsayisi);
range araclar=1..aracsayisi;
range ser=(1..sehirsayisi-2);
range ser1=(1..fac);
range savsay=1..savunmasayisi;
{int} anausler=...;
int B=...;// maksimum bacak sayısı
int G=...;// maksimum uçuş süresi
tuple aractipiozellikleri {
  float hiz;
  int kapasite;
  int maliyet;
  int anaus;
  float menzil;};
aractipiozellikleri arac[araclar]=...;
range riskfaktoru=1..risksayisi;
float faktor[riskfaktoru]=...;
tuple usriskleri{ // uslere aıt rısklerın tanımlanması
  int risk1;
  int risk2;
  int risk5;};
usriskleri usriski[sehirler]=...;
tuple edge    {int i; int j;};
setof(edge) Edges={<i,j> |  i,j in sehirler : i!=j};
setof(edge) asd[araclar][Edges][ser][ser1]=...;
float mesafeler[sehirler][sehirler];
float talepler[sehirler][sehirler]=...;
float ayritlar[sehirler][sehirler][savsay];
float ayritort[sehirler][sehirler];
dvar int x[Edges][araclar] in 0..1;
dvar int+ y[Edges][araclar];//i den j ye giderken k aracı üzerindeki yük
dvar int+ u[Edges][araclar];//i şehrinden j şehrine k aracı ile gönderilen ürün miktarı
dvar float+ a[sehirler][araclar];
```

```
tuple koor{
  float xx;
  float yy;};
koor uskoor[sehirler]=...;
koor savkoor[savsay]=...;

execute MESAFE{
  for(var i in sehirler){
    for (var j in sehirler){mesafeler[i][j]=Math.sqrt((uskoor[i].xx-
uskoor[j].xx)*(uskoor[i].xx-uskoor[j].xx)+ (uskoor[i].yy-uskoor[j].yy)*(uskoor[i].yy-
uskoor[j].yy));}}};

execute AYRIT{
  for(var i in sehirler){
    for(var j in sehirler){
      for(var k in savsay){

          ayritlar[i][j][k]=Math.sqrt((((uskoor[i].xx+uskoor[j].xx)/2)-
savkoor[k].xx)*(((uskoor[i].xx+uskoor[j].xx)/2)-savkoor[k].xx)+
(((uskoor[i].yy+uskoor[j].yy)/2)-savkoor[k].yy)*(((uskoor[i].yy+uskoor[j].yy)/2)-
savkoor[k].yy));ayritlar[i][i][k]=0;}}}};

execute AYRITOR{
  for(var i in sehirler){
    for(var j in sehirler){
      for(var k in savsay){
    ayritort[i][j]= ayritort[i][j]+(ayritlar[i][j][k])/savunmasayisi;}}}};

//        AMAÇ FONKSIYONU
minimize sum(i,j in sehirler, k in araclar : i!=j)(x[<i,j>][k]*(mesafeler [i][j]));

subject to {

// ARAÇLAR DEPODAN ÇIKSIN
forall (k in araclar)
  ct1:
    sum(i in sehirler:i!=arac[k].anaus)x[<arac[k].anaus,i>][k]<=1;

// ARAÇLAR DEPOYA DÖNSÜN
forall (k in araclar)
  ct2:
    sum(i in sehirler: i!=arac[k].anaus)(x[<i,arac[k].anaus>][k])<=1;

// BIR ŞEHRE GELEN ARAÇ BAŞKA ŞEHRE BU ŞEHIRDEN GITSIN
forall(i in sehirler, k in araclar)
  ct3:
    sum(j in sehirler: j!=i) (x[<i,j>][k]-x[<j,i>][k])==0;

// TALEPLER KARŞILANSIN
```

```
forall(i,j in sehirler:i!=j)
  ct4:
   sum(k in araclar) u[<i,j>][k]==talepler[i][j];

//  KAPASITE AŞILMASIN
forall(i,j in sehirler, k in araclar : i!=j)
  ct5:
    y[<i,j>][k]<=arac[k].kapasite;

//          FLOW CONSTRAINTS
forall ( <i,j> in Edges, k in araclar)
  ct6:
(u[<i,j>][k]!=0)=>    (x[<i,j>][k]==1)+   sum(r   in   ser,   t   in   ser1)(sum(<l,m>   in
asd[k][<i,j>][r][t])x[<l,m>][k]==r+1)==1;

//  BAĞLANTI KISITI
forall(i,j in sehirler, k in araclar : i!=j)
  ct7:
    (y[<i,j>][k]>=1)-(x[<i,j>][k]==1)==0;

 //SUBTOUR ELIMINATION
forall (i,j in sehirler, k in araclar : i!=j && i!=arac[k].anaus && j!=arac[k].anaus)
  ct8:
    a[i][k]-a[j][k] +sehirsayisi*x[<i,j>][k]<=sehirsayisi-1;

//  AKIŞ KISITI

forall(i in sehirler, k in araclar: i!=arac[k].anaus)
  ct9:
          sum(g      in      sehirler:g!=i)(y[<g,i>][k]-(u[<g,i>][k]))==sum(g      in      sehirler:
g!=i)(y[<i,g>][k]-(u[<i,g>][k]));

forall(j in sehirler, k in araclar:j==arac[k].anaus)
  ct10:
          sum(i in sehirler:i!=j)(y[<j,i>][k]-u[<j,i>][k])==0;

forall(j in sehirler, k in araclar:j==arac[k].anaus)
  ct11:
          sum(i in sehirler:i!=j)(y[<i,j>][k]-u[<i,j>][k])==0;

// TOPLAM BACAK SAYISI SINIRI

forall(k in araclar)
  ct12:
    sum(<i,j> in Edges) x[<i,j>][k]<=B;


//          TOPLAM UÇUŞ SÜRESI SINIRI
```

```
forall(k in araclar)
  ct13:
    sum(i,j in sehirler: i!=j) x[<i,j>][k]*(mesafeler [i][j]/arac[k].hiz)+0.25<=G;

//   MENZİL KISITI

forall(i,j in sehirler: i!=j, k in araclar)
  ct14:
    x[<i,j>][k]*mesafeler [i][j]<=arac[k].menzil;

  //RISK KISITLARI

// HAVA RISKLERI
          forall(k in araclar)
          ct15:
    sum(i,j in sehirler: i!=j)x[<i,j>][k]*usriski[i].risk1 <= sum(i,j in sehirler:
i!=j)x[<i,j>][k]*faktor[1];


// ANA ÜSTE PIST KÖTÜ ISE BU ÜSTEN UÇAK KALKMASIN
          ct16:
                    sum(i,j in sehirler, k in araclar: usriski[i].risk2>0 && i==arac[k].anaus)
x[<i,j>][k]==0;

// DIĐER ÜSLERDE PIST KÖTÜ ISE 1 SAATTEN ÖNCE BU ÜSSE INIŞ OLMASIN
  forall(i in sehirler, k in araclar: usriski[i].risk2!=0 && i!=arac[k].anaus)
          CT17:
                    if                          ((mesafeler[arac[k].anaus][i]/arac[k].hiz)<=faktor[2])
x[<arac[k].anaus,i>][k]==0;

  forall(i,j in sehirler, k in araclar: i!=j && usriski[i].risk2!=0 && i!=arac[k].anaus &&
j!=arac[k].anaus )
          CT18:
                    if        ((mesafeler[arac[k].anaus][j]/arac[k].hiz        +        0.25        +
mesafeler[j][i]/arac[k].hiz)<=faktor[2])
                     x[<arac[k].anaus,j>][k]+ x[<j,i>][k]==0;

// BIR AYRITTAN UÇAKLARIN KAÇ DEFA GEÇEBILECEĐINI KISITLAR
  forall(i,j in sehirler:i!=j)
          ct19:
                    sum(k in araclar) (x[<i,j>][k]+x[<j,i>][k])<=faktor[3];

  // HAVA SAVUNMA SISTEMLERINDEN UZAKLIĐI KISITLAR
  forall(k in araclar)
    ct20:
          sum(<i,j>                in                Edges)(x[<i,j>][k]*((sum(h                in
savsay)(ayritlar[i][j][h]))/savunmasayisi))<=faktor[4]*sum(i,j in sehirler:i!=j)x[<i,j>][k];

// ALÇAK UÇUŞ SÜRESI KISITI
```

```
forall(k in araclar)
   ct21:
        sum( i,j in sehirler: usriski[i].risk5!=0 && i!=j)((x[<i,j>][k]*(mesafeler[i][j])/arac[k].hiz)+

(x[<j,i>][k]*(mesafeler[j][i])/arac[k].hiz))+
                                    sum( i,j in sehirler: usriski[i].risk5!=0 &&usriski[j].risk5!=0 && i!=j )(x[<i,j>][k]*((mesafeler[i][j])/arac[k].hiz))<=faktor[5];}
```

C.2. Data Dosyası

```
/*********************************************
 * OPL 12.8 Data
 * Author: OZGUR

SheetConnection sheet("1.xls");
SheetConnection sheetData("1.xls");
aracsayisi  from SheetRead(sheet,"ARACLAR!H1");
arac from SheetRead(sheetData,"aracbilgileri");
sehirsayisi  from SheetRead(sheet,"KONUM!H1");
B=10;// maksimum bacak sayısı
G=12;// maksimum uçu süresi

risksayisi=5;
anaussayisi=1;
anausler from SheetRead(sheet,"anaus");
savunmasayisi=5;      // hava savunma ünitesı mıktarı

uskoor from SheetRead(sheet,"KONUM!A1:B5");
savkoor from SheetRead(sheet,"SAVUNMA!A1:B5");
talepler from SheetRead(sheet,"TALEP!A1:E5");

usriski from SheetRead(sheet,"riskler");

//Rısk Faktör Sayıları
//      Hava          Erteleme      Max Geçiş     Hava          Alçak
//      Durumu        Zamanları     Sayısı        Savunma       Uçuş
//      Rısk1         Rısk2         Rısk3         Rısk4         Rısk5
faktor=[90            0.5           3             2800          10];

asd=[[[[{< 1 3 > < 3 2 >}
{< 1 4 > < 4 2 >}
{< 1 5 > < 5 2 >}
][{< 1 3 > < 3 4 > < 4 2 >}
{< 1 3 > < 3 5 > < 5 2 >}
{< 1 4 > < 4 3 > < 3 2 >}
{< 1 4 > < 4 5 > < 5 2 >}
{< 1 5 > < 5 3 > < 3 2 >}
{< 1 5 > < 5 4 > < 4 2 >}
][{< 1 3 > < 3 4 > < 4 5 > < 5 2 >}
{< 1 3 > < 3 5 > < 5 4 > < 4 2 >}
{< 1 4 > < 4 3 > < 3 5 > < 5 2 >}
{< 1 4 > < 4 5 > < 5 3 > < 3 2 >}
{< 1 5 > < 5 3 > < 3 4 > < 4 2 >}
{< 1 5 > < 5 4 > < 4 3 > < 3 2 >}]]

[[{< 1 2 > < 2 3 >}
{< 1 4 > < 4 3 >}
{< 1 5 > < 5 3 >}
```

][{< 1 2 > < 2 4 > < 4 3 >}
{< 1 2 > < 2 5 > < 5 3 >}
{< 1 4 > < 4 2 > < 2 3 >}
{< 1 4 > < 4 5 > < 5 3 >}
{< 1 5 > < 5 2 > < 2 3 >}
{< 1 5 > < 5 4 > < 4 3 >}
][{< 1 2 > < 2 4 > < 4 5 > < 5 3 >}
{< 1 2 > < 2 5 > < 5 4 > < 4 3 >}
{< 1 4 > < 4 2 > < 2 5 > < 5 3 >}
{< 1 4 > < 4 5 > < 5 2 > < 2 3 >}
{< 1 5 > < 5 2 > < 2 4 > < 4 3 >}
{< 1 5 > < 5 4 > < 4 2 > < 2 3 >}]]

[[{< 1 2 > < 2 4 >}
{< 1 3 > < 3 4 >}
{< 1 5 > < 5 4 >}
][{< 1 2 > < 2 3 > < 3 4 >}
{< 1 2 > < 2 5 > < 5 4 >}
{< 1 3 > < 3 2 > < 2 4 >}
{< 1 3 > < 3 5 > < 5 4 >}
{< 1 5 > < 5 2 > < 2 4 >}
{< 1 5 > < 5 3 > < 3 4 >}
][{< 1 2 > < 2 3 > < 3 5 > < 5 4 >}
{< 1 2 > < 2 5 > < 5 3 > < 3 4 >}
{< 1 3 > < 3 2 > < 2 5 > < 5 4 >}
{< 1 3 > < 3 5 > < 5 2 > < 2 4 >}
{< 1 5 > < 5 2 > < 2 3 > < 3 4 >}
{< 1 5 > < 5 3 > < 3 2 > < 2 4 >}]]

[[{< 1 2 > < 2 5 >}
{< 1 3 > < 3 5 >}
{< 1 4 > < 4 5 >}
][{< 1 2 > < 2 3 > < 3 5 >}
{< 1 2 > < 2 4 > < 4 5 >}
{< 1 3 > < 3 2 > < 2 5 >}
{< 1 3 > < 3 4 > < 4 5 >}
{< 1 4 > < 4 2 > < 2 5 >}
{< 1 4 > < 4 3 > < 3 5 >}
][{< 1 2 > < 2 3 > < 3 4 > < 4 5 >}
{< 1 2 > < 2 4 > < 4 3 > < 3 5 >}
{< 1 3 > < 3 2 > < 2 4 > < 4 5 >}
{< 1 3 > < 3 4 > < 4 2 > < 2 5 >}
{< 1 4 > < 4 2 > < 2 3 > < 3 5 >}
{< 1 4 > < 4 3 > < 3 2 > < 2 5 >}]]

[[{< 2 3 > < 3 1 >}
{< 2 4 > < 4 1 >}
{< 2 5 > < 5 1 >}
][{< 2 3 > < 3 4 > < 4 1 >}

{< 2 3 > < 3 5 > < 5 1 >}
{< 2 4 > < 4 3 > < 3 1 >}
{< 2 4 > < 4 5 > < 5 1 >}
{< 2 5 > < 5 3 > < 3 1 >}
{< 2 5 > < 5 4 > < 4 1 >}
][{< 2 3 > < 3 4 > < 4 5 > < 5 1 >}
{< 2 3 > < 3 5 > < 5 4 > < 4 1 >}
{< 2 4 > < 4 3 > < 3 5 > < 5 1 >}
{< 2 4 > < 4 5 > < 5 3 > < 3 1 >}
{< 2 5 > < 5 3 > < 3 4 > < 4 1 >}
{< 2 5 > < 5 4 > < 4 3 > < 3 1 >}]]

[[{< 2 4 > < 4 3 >}
{< 2 5 > < 5 3 >}
][{< 2 4 > < 4 5 > < 5 3 >}
{< 2 5 > < 5 4 > < 4 3 >}]]

[[{< 2 3 > < 3 4 >}
{< 2 5 > < 5 4 >}
][{< 2 3 > < 3 5 > < 5 4 >}
{< 2 5 > < 5 3 > < 3 4 >}]]

[[{< 2 3 > < 3 5 >}
{< 2 4 > < 4 5 >}
][{< 2 3 > < 3 4 > < 4 5 >}
{< 2 4 > < 4 3 > < 3 5 >}]]

[[{< 3 2 > < 2 1 >}
{< 3 4 > < 4 1 >}
{< 3 5 > < 5 1 >}
][{< 3 2 > < 2 4 > < 4 1 >}
{< 3 2 > < 2 5 > < 5 1 >}
{< 3 4 > < 4 2 > < 2 1 >}
{< 3 4 > < 4 5 > < 5 1 >}
{< 3 5 > < 5 2 > < 2 1 >}
{< 3 5 > < 5 4 > < 4 1 >}
][{< 3 2 > < 2 4 > < 4 5 > < 5 1 >}
{< 3 2 > < 2 5 > < 5 4 > < 4 1 >}
{< 3 4 > < 4 2 > < 2 5 > < 5 1 >}
{< 3 4 > < 4 5 > < 5 2 > < 2 1 >}
{< 3 5 > < 5 2 > < 2 4 > < 4 1 >}
{< 3 5 > < 5 4 > < 4 2 > < 2 1 >}]]

[[{< 3 4 > < 4 2 >}
{< 3 5 > < 5 2 >}
][{< 3 4 > < 4 5 > < 5 2 >}
{< 3 5 > < 5 4 > < 4 2 >}]]

[[{< 3 2 > < 2 4 >}

{< 3 5 > < 5 4 >}
][{< 3 2 > < 2 5 > < 5 4 >}
{< 3 5 > < 5 2 > < 2 4 >}]]

[[{< 3 2 > < 2 5 >}
{< 3 4 > < 4 5 >}
][{< 3 2 > < 2 4 > < 4 5 >}
{< 3 4 > < 4 2 > < 2 5 >}]]

[[{< 4 2 > < 2 1 >}
{< 4 3 > < 3 1 >}
{< 4 5 > < 5 1 >}
][{< 4 2 > < 2 3 > < 3 1 >}
{< 4 2 > < 2 5 > < 5 1 >}
{< 4 3 > < 3 2 > < 2 1 >}
{< 4 3 > < 3 5 > < 5 1 >}
{< 4 5 > < 5 2 > < 2 1 >}
{< 4 5 > < 5 3 > < 3 1 >}
][{< 4 2 > < 2 3 > < 3 5 > < 5 1 >}
{< 4 2 > < 2 5 > < 5 3 > < 3 1 >}
{< 4 3 > < 3 2 > < 2 5 > < 5 1 >}
{< 4 3 > < 3 5 > < 5 2 > < 2 1 >}
{< 4 5 > < 5 2 > < 2 3 > < 3 1 >}
{< 4 5 > < 5 3 > < 3 2 > < 2 1 >}]]

[[{< 4 3 > < 3 2 >}
{< 4 5 > < 5 2 >}
][{< 4 3 > < 3 5 > < 5 2 >}
{< 4 5 > < 5 3 > < 3 2 >}]]

[[{< 4 2 > < 2 3 >}
{< 4 5 > < 5 3 >}
][{< 4 2 > < 2 5 > < 5 3 >}
{< 4 5 > < 5 2 > < 2 3 >}]]

[[{< 4 2 > < 2 5 >}
{< 4 3 > < 3 5 >}
][{< 4 2 > < 2 3 > < 3 5 >}
{< 4 3 > < 3 2 > < 2 5 >}]]

[[{< 5 2 > < 2 1 >}
{< 5 3 > < 3 1 >}
{< 5 4 > < 4 1 >}
][{< 5 2 > < 2 3 > < 3 1 >}
{< 5 2 > < 2 4 > < 4 1 >}
{< 5 3 > < 3 2 > < 2 1 >}
{< 5 3 > < 3 4 > < 4 1 >}
{< 5 4 > < 4 2 > < 2 1 >}
{< 5 4 > < 4 3 > < 3 1 >}

][{< 5 2 > < 2 3 > < 3 4 > < 4 1 >}
{< 5 2 > < 2 4 > < 4 3 > < 3 1 >}
{< 5 3 > < 3 2 > < 2 4 > < 4 1 >}
{< 5 3 > < 3 4 > < 4 2 > < 2 1 >}
{< 5 4 > < 4 2 > < 2 3 > < 3 1 >}
{< 5 4 > < 4 3 > < 3 2 > < 2 1 >}]]

[[{< 5 3 > < 3 2 >}
{< 5 4 > < 4 2 >}
][{< 5 3 > < 3 4 > < 4 2 >}
{< 5 4 > < 4 3 > < 3 2 >}]]

[[{< 5 2 > < 2 3 >}
{< 5 4 > < 4 3 >}
][{< 5 2 > < 2 4 > < 4 3 >}
{< 5 4 > < 4 2 > < 2 3 >}]]

[[{< 5 2 > < 2 4 >}
{< 5 3 > < 3 4 >}
][{< 5 2 > < 2 3 > < 3 4 >}
{< 5 3 > < 3 2 > < 2 4 >}]]]];

Ek Açıklama-D: hGA Matlab Kodu

D.1. Ana Kod

```
%% GENREATION OF INITIAL POPULATION
[ intpop, intpop_tasimalar, intpop_demandler]=initial_population_construction2(); % ilk popülasyonun yaratılması
%% FITNESS CALCULATION FOR INITIAL POPULATION
intpop_fitness=cell(celldimension,celldimension);
intpop_fitnesstotal=zeros(celldimension);
for i=1:celldimension
   for j=1:celldimension
[intpop_fitness{i,j}(1),intpop_fitness{i,j}(2),intpop_fitness{i,j}(3),intpop_fitness{i,j}(4),intpop_fitness{i,j}(5),intpop_fitness{i,j}(6),intpop_fitness{i,j}(7),intpop_fitness{i,j}(8),intpop_fitness{i,j}(9),intpop_fitness{i,j}(10),intpop_fitness{i,j}(11)]=fitnessfunction(intpop{i,j},intpop_tasimalar{i,j});
      intpop_fitnesstotal(i,j)=intpop_fitness{i,j}(1);
   end
end
best_found=cell(max_generetion,4);
deviation=zeros(max_generetion,2);
%% GENERATIONS
midgen=cell(1,8);
midgen_tasima=cell(1,8);
midgen_demand=cell(3,8); % 1.satır taşınan demandlerin sırasını, 2.satır tüm demandler taşınmışsa=1 diğer durumda=0, 3.satır demandlerin araçlara atanma sırası
midgen_fitnesstotal=zeros(1,8);
midgen_fitness=zeros(1,8);
.
.
.
nextgen=cell(celldimension,celldimension);
nextgen_tasima=cell(celldimension,celldimension);
nextgen_demand=cell(celldimension,celldimension); % 1.satır taşınan demandlerin sırasını, 2.satır tüm demandler taşınmışsa=1 diğer durumda=0, 3.satır demandlerin araçlara atanma sırası
nextgen_fitness=cell(celldimension,celldimension);% her generationda oluşan fitness function sonuçlarını saklar
plot(1:max_generetion,deviation(1:max_generetion,1),'-
r',1:max_generetion,deviation(1:max_generetion,2),'--
b',1:max_generetion,cell2mat(best_found(1:max_generetion,4)),':g');
legend('max','mean','all best')
disp(best_found(max_generetion,4));
toc
```

D.2. İlk Popülasyonun Yaratılması

```
function [ intpop, intpop_tasimalar, intpop_demandler]=initial_population_construction2()

while i<=(celldimension*celldimension)
   ucaksay=randi(ar);
   ucakno=sort(datasample(arac_kume,ucaksay,'Replace',false)); %seçilen uçaklar?n farkl? uçak olmasını sağlar
   route=cell(1,maxleg);
   for k=1:ucaksay
      route{ucakno(k),1}=araclar(ucakno(k),4);  %rotanın ilk noktasın? uçağın? ana üssü yapar


.
.
.
 [intpopulation1{1,i},intpopulation2{1,i},intpopulation2{2,i},intpopulation2{3,i}] =ucak_yukle_demand_split(intpopulation{1,i}); % ucak_yukle fonksiyonu ile populasyona ait demand atamalarının yapılması
   intpopulation3{1,i}=intpopulation2(:,i);
   if intpopulation2{2,i}==1 %bütün demandler taşınıyorsa sayacı bir arttırır

[intpopulation{1,i},intpopulation1{1,i}]=gereksiz_bacak_silme(intpopulation{1,i},intpopulation1{1,i});

      for j=1:size(intpopulation{1,i},1)

         aradeger=cell2mat(intpopulation{1,i}(j,:));
         aradeger=num2cell(aradeger);
         intpopulation{1,i}(j,1:size(aradeger,2))={aradeger{1:size(aradeger,2)}};
         for mq=(size(aradeger,2)+1):size(intpopulation{1,i},2)
            intpopulation{1,i}{j,mq}=[];
         end
         aradeger=[];
      end
      i=i+1;
   end
   demand=demand1;
end
intpop=reshape(intpopulation,[celldimension,celldimension]);
intpop_tasimalar=reshape(intpopulation1(1,:),[celldimension,celldimension]);
intpop_demandler=reshape(intpopulation3,[celldimension,celldimension]);
end
```

D.3. Uçak Yükleme Fonksiyonu

```
function [y,z,c,x]= ucak_yukle_demand_split(yolum) %
%y= i'den j'ye giden uçağın taşıdığı yük miktarı;  z=taşınan yüklerin %taşınma sırası; c=demandlerin hepsi taşınmışsa c=1 otherwise c=0;%x=uçaklara yükleme sırası
if size(yolum,2)<=1
   y=0;
   z=0;
   c=0;
   x=0;
else
   ucaklar=(1:size(yolum,1));
   demand= evalin('base', 'demand');
   araclar= evalin('base', 'araclar');
   say=1;
   z=cell(1,size(yolum,1));
   y=cell(size(yolum,1),1);
   x=zeros(1,size(yolum,1));
   % rotadaki her uçak için döngü
   while ~isempty(ucaklar)
      ucakno=ucaklar(randi(numel(ucaklar))); % rast gele bir uçak numarası seçer
      ucaklar(ucaklar==ucakno)=[]; % seçilen uçağı uçaklar kümesinden çıkarır
      ucakyolu=cell2mat(yolum(ucakno,:)); % seçilen uçağın rotasının arraye çevrilmesi
      if isempty(ucakyolu) % uçağa ait bir yol yoksa bir sonraki uçağa geçirir.
         tasinan_yuk(ucakno,:)=0;
         ucak_yuku_listesi{ucakno}=[];
         z{ucakno}=ucak_yuku_listesi{ucakno}; % her uçağın taşıdığı yükler ve bu yüklerin taşınma sırası
         x(say)=ucakno; % uçaklara yükleme sırası
         say=say+1;
         continue
      end
      x(say)=ucakno; % uçaklara yükleme sırası
      say=say+1;
   end


.
.
.


      c=0;
   end
   if rotasize<size(yolum,2)
      tasinan_yuk(ucakno,rotasize+1:size(yolum,2))=0;
   end
   y=tasinan_yuk; % i'den j'ye giden uçağın taşıdığı yük miktarı. eğer kapasiteden fazla ise çözüm infeasable
end
end
```

D.4. Gereksiz Bacak Silme Fonksiyonu

```
function [a,b] = gereksiz_bacak_silme(birey,birey_tasimalar)
ar=size(birey,1); % bireydeki rota sayısı
ac=size(birey,2); % bireydeki bacak sayısı
z=0;

for j=1:ar
   for i=(find(diff(birey_tasimalar(j,:))==0)+1) %bireyin taşımalarında art arda iki aynı? Değer olan noktalar üzerinden döngü yaptırır.
      if isempty(birey{j,i-z}) %bireyin bos olan kısımlarına gelince döngüden çıkarır
         break
      else
         i=i-z;
         birey(j,:) = [birey(j, setdiff(1:ac,i)) {[]}]; % art arda iki değerin aynı olan yoldaki şehri bireyden siler
         birey_tasimalar(j,:) = [birey_tasimalar(j, setdiff(1:ac,i)) (0)]; % art arda iki değerin aynı olan taşımayı kaldırır
      end
      z=z+1;
   end
   z=0;
   if isempty(birey{j,2}) || isempty(birey{j,3})
      for k=1:3
         birey{j,k}=[];
         birey_tasimalar(j,k)=0;
      end
   end
   %rotanın son noktasının ana üs olmasını sağlar.
   if size(cell2mat(birey(j,:)),2)>0
      if birey{j,1}~=birey{j,(size(cell2mat(birey(j,:)),2))}  % silmeler sonunda uçağın son durağı ana üs değil ise ana üsse gitmesi için bir bacak ekler
         birey{j,(size(cell2mat(birey(j,:)),2))+1}=birey{j,1};
      end
   end

end
a=birey;
b=birey_tasimalar;
end
```

D.5. Uyum Değeri Fonksiyonu

```
function[fitness1,a2,b3,c4,d5,e6,f7,g8,h9,ii10,jj11,kk11]=
fitnessfunction(intpop,intpop_tasimalar)
if size(intpop,2)<=2
   fitness1=10000000000;% Toplam Fitness
   a2=0;          % Toplam rota uzunluğu
   b3=0;          % Leg count Ceza for maxleg
   c4=0;          % Workhour Ceza
   d5=0;          % Kapasite Ceza
   e6=0;          % Range Ceza
   f7=0;          % Hava Riski Ceza
   g8=0;          % Bombalanmış Pist Ceza
   h9=0;          % Alçak Uçuş Ceza
   ii10=0;        % Air Defence Ceza
   jj11=0;        % Edge usage cezası
   kk11=0;        % Bir rota içerisinde bir üsse Maks_ugrama_sayisi cesazı
else
   distmat= evalin('base', 'dist'); %distance matrisinin alınması
   maxleg= evalin('base', 'maxleg'); %maksimum bacak sayısının alınması
   pilotworkhour= evalin('base', 'pilotworkhour'); % pilot work hour alınması
   araclar=evalin('base', 'araclar'); %araç bilgilerinin alınması 1. kolon hız, 2. kolon kapasite,
3. kolon maliyet, 4. kolon ana üs, 5. kolon menzil
   koordinat=evalin('base', 'koordinat'); %araç bilgilerinin alınması  1. kolon koordinat x, 2.
kolon koordinat 4, 3. kolon hava riski, 4. kolon zarar görmüş pist, 5. kolon alçak uçu
   accaptable_weather_risk=evalin('base', 'accaptable_weather_risk');
   lowfligt_time=evalin('base', 'lowflight_time');
   airdefencedist=evalin('base', 'airdefencedist');
   distance_to_airdefence=evalin('base', 'distance_to_airdefence');
   edgeusage=evalin('base', 'edgeusage');
   maks_ugrama_sayisi=evalin('base', 'maks_ugrama_sayisi');

.
.
.

      if size(cell2mat(intpop(i,:)),2)>3
      us_say_kont=tabulate([intpop{i,:}]);
      if all(us_say_kont(:,2)<=maks_ugrama_sayisi)
         ceza_maks_ugrama_sayisi(i)=0;
      else
         ceza_maks_ugrama_sayisi(i)=ceza_maks_ugrama_sayisi_deger;
      end
      else
         ceza_maks_ugrama_sayisi(i)=0;
      end
   end
end
end
```

D.6. Komşu Bulma Fonksiyonu

```
function [parent1, parent2]=find_neighbour(x,y)
celldimension= evalin('base', 'celldimension'); %celldimension alınması
generation= evalin('base', 'intpop'); %generation alınması
komsular=cell(1,5); %komsular= komsular (1)=merkezin solu; komsular(2)=merkezin sağı; komsular(3)=merkezin altı; komsular(4)=merkezin üstü; komsular(5)=kendisi
if y-1==0
   komsular{1}=[];
else
   komsular{1}=generation{x,y-1};
end
if y+1>celldimension
   komsular{2}=[];
else
   komsular{2}=generation{x,y+1};
end
if x+1>celldimension
   komsular{3}=[];
else
   komsular{3}=generation{x+1,y};
end
if x-1==0
   komsular{4}=[];
else
   komsular{4}=generation{x-1,y};
end
komsular{5}=generation{x,y};
ran1=randi([1,5],1);
ran2=randi([1,5],1);
kontrol=true;
while kontrol
   if ~isempty(komsular{ran1})
      if ~isempty(komsular{ran2})
         if ran1~=ran2
            kontrol=false;
         else
            ran2=randi([1,5],1);
         end
      else
         ran2=randi([1,5],1);
      end
   else
      ran1=randi([1,5],1);
   end
end
parent1=komsular(ran1);
parent2=komsular(ran2);
end
```

D.7. Çaprazlama Fonksiyonu-1

```
function [a,b,c,d] = crossover_in_same_aircraft( parent1,parent2 )
ar= evalin('base', 'ar'); % uçak sayısı
child=cell(1,4);
for i=1:ar
   if i>size(parent1,1)
      ucakyolu1=[];
   else
      ucakyolu1=cell2mat(parent1(i,:));
   end
   if  i>size(parent2,1)
      ucakyolu2=[];
   else
      ucakyolu2=cell2mat(parent2(i,:));
   end
   if isempty(ucakyolu1) && isempty(ucakyolu2) && ar<2 %her iki uçak yolu da boşsa
      a=parent1;
      b=parent2;
      c=parent1;
      d=parent2;
      return
   else
      if isempty(ucakyolu1)
         child{1}(i,1:size(ucakyolu2,2))=ucakyolu2;
         child{3}(i,1:size(ucakyolu2,2))=ucakyolu2;
         child{2}(i,1:size(ucakyolu2,2))=ucakyolu2;
         child{4}(i,1:size(ucakyolu2,2))=ucakyolu2;
         continue
      end
.
.
.
child{1}=num2cell(child{1}); %double'ı cell'e dönüştürür.
child{2}=num2cell(child{2});
child{3}=num2cell(child{3});
child{4}=num2cell(child{4});
child{1}(cellfun(@not,child{1}))={[]}; % sıfırları cellden atar
child{2}(cellfun(@not,child{2}))={[]};
child{3}(cellfun(@not,child{3}))={[]};
child{4}(cellfun(@not,child{4}))={[]};
a=child{1};
b=child{2};
c=child{3};
d=child{4};
end
```

D.8. Çaprazlama Fonksiyonu-2

```
function [a,b,c,d] = crossover_different_aircraft(parent1,parent2)

araclar=evalin('base', 'araclar'); % % 1. kolon hız, 2. kolon kapasite, 3. kolon maliyet, 4. kolon ana üs, 5. kolon menzil
child=cell(1,4);
k=0;
ucaklar1=(1:size(parent1,1));
ucaklar2=(1:size(parent2,1));
%% Parent1'e Göre Crossover

for i=1:size(parent1,1)
   ucakno1=ucaklar1(randi(numel(ucaklar1)));
   ucakno2=ucaklar2(randi(numel(ucaklar2)));
   ucakyolu1=cell2mat(parent1(ucakno1,:));
   ucakyolu2=cell2mat(parent2(ucakno2,:));
   while isempty(ucakyolu1) && isempty(ucakyolu2) % Her iki rotanında boş olmamasını sağlar
      ucakno1=ucaklar1(randi(numel(ucaklar1)));
      ucakyolu1=cell2mat(parent1(ucakno1,:));
      k=k+1;
      if k>(size(ucakyolu1,2)+100)
         break
      end

   end

   if k>=100
      a=parent1;
      c=parent2;
   else


.
.
.

a=child{1};
b=child{2};
c=child{3};
d=child{4};


end
```

D.9. Mutasyon Fonksiyonu-1

```
function [a] = mutation_araya_sokma(birey) % rotada 4 şehir olursa sonsuz döngüye girer
ar=size(birey,1); % uçak sayısı
say=1;
if size(birey,2)<=4 || all(cellfun('isempty',birey(:,1)))% birey boşsa (hiçbir uçağa ait yol yoksa) mutasyon yapma
a=birey;
else
point3=randi([1,ar],1);
point4=size(cell2mat(birey(point3,:)),2); % seçilen uçak rotasının uzunluğu
kontrol1=true;
while kontrol1 % rota uzunluğu 4'den küçükse, rota uzunluğu 43'den büyük olan aracı seçer
   if point4<=4
      kontrol1=true;
      point3=randi([1,ar],1);
      point4=size(cell2mat(birey(point3,:)),2);
   else
      kontrol1=false;
   end
   if say>=ar*5
      kontrol1=false;
      a=birey;
      return;
   end
   say=say+1;
end

point1=randi([2,point4-2],1);

kontrol=true;
while kontrol
   point2=randi([point1,point4-1],1);
   if point1==point2
      kontrol=true;
   else
      kontrol=false;
   end
end

yol=birey;
yol(point3,1:point1-1)=birey(point3,1:point1-1);
yol(point3,point1:point2-1)=birey(point3,point1+1:point2);
yol(point3,point2)=birey(point3,point1);
yol(point3,point2+1:point4)=birey(point3,point2+1:point4);
a=yol;
end
end
```

D.10. Mutasyon Fonksiyonu-2

```
function [a] = mutation_revers(birey)
ar=size(birey,1); % uçak sayısı
if size(birey,2)<=3 || all(cellfun('isempty',birey(:,1)))
   a=birey;
else
point3=randi([1,ar],1);
yol=birey(point3,:);
say=1;
while isempty(yol)
point3=randi([1,ar],1);
yol=birey(point3,:);
end
point4=size(cell2mat(yol),2);  % seçilen uçak rotasının uzunluğu
kontrol1=true;
while kontrol1 % rota uzunluğu 3'den küçükse, rota uzunluğu 3'den büyük olan aracı seçer
   if point4<=3
      kontrol1=true;
      point3=randi([1,ar],1);
      yol=birey(point3,:);
      point4=size(cell2mat(yol),2);
   else
      kontrol1=false;
   end
   if say>=ar*5
      kontrol1=false;
      a=birey;
      return;
   end
   say=say+1;
end
point1=randi([2,point4-2],1);
kontrol=true;
while kontrol
   point2=randi([point1,point4-1],1);
   if point1==point2
      kontrol=true;
   else
      kontrol=false;
   end
end

birey(point3,point1:point2)=fliplr(birey(point3,point1:point2));
a=birey;
end
end
```

D.11. Mutasyon Fonksiyonu-3

```
function [a] = mutation_degis_tokus(birey)
ar=size(birey,1); % uçak sayısı
say=1;
if size(birey,2)<=3 || all(cellfun('isempty',birey(:,1)))
   a=birey;
 else
point3=randi([1,ar],1);
point4=size(cell2mat(birey(point3,:)),2);  % seçilen uçak rotasının uzunluğu
while point4==0;
point3=randi([1,ar],1);
point4=size(cell2mat(birey(point3,:)),2);  % seçilen uçak rotasının uzunluğu
end
 kontrol1=true;
while kontrol1 % rota uzunluğu 3'den küçükse, rota uzunluğu 3'den büyük olan aracı seçer
   if point4<=3
      kontrol1=true;
      point3=randi([1,ar],1);
      point4=size(cell2mat(birey(point3,:)),2);
    else
      kontrol1=false;
   end
   if say>=ar*5
      kontrol1=false;
      a=birey;
      return;
   end
   say=say+1;
end

point1=randi([2,point4-2],1);

kontrol=true;
while kontrol
   point2=randi([point1,point4-1],1);
   if point1==point2
      kontrol=true;
   else
      kontrol=false;
   end
end
ara1=birey(point3,point1);
ara2=birey(point3,point2);
birey(point3,point1)=ara2;
birey(point3,point2)=ara1;

a=birey;
end
end
```

ÖZGEÇMİŞ

20.10.1981 tarihinde Eskişehir'de dünyaya gelen Özgür AKSOY, ilk ve orta öğretimini Eskişehir'de tamamlamıştır. Lise eğitimini Bursa Işıklar Askeri Lisesi'nde tamamlamış ve öğrenci seçme uçuşlarını başarı ile tamamlayarak Hava Harp Okulu'na girmeye hak kazanmıştır. 2003 yılında Hava Harp Okulu'ndan mezun olmuş ve İzmir 2'nci Ana Jet Üs K. lığında pilotaj eğitimine katılmıştır. 2005 yılında pilotaj eğitimini başarı ile tamamlayarak Hava Kuvvetlerinde Pilot Subay olarak göreve başlamıştır. Takiben sırasıyla Ankara, Kabil/Afganistan, Malatya, Konya ve Eskişehir'de farklı birliklerde çalışmıştır. Konya Selçuk Üniversitesi Endüstri Mühendisliği Bölümü yüksek lisans eğitimini 2007 yılında başarı ile tamamlayarak 2007 yılında Eskişehir Osmangazi Üniversitesi'nde Doktora eğitimine başlamıştır. Halen Hava Kuvvetlerinde Hv.Plt.Bnb. olarak çalışan Özgür AKSOY evli ve bir kız çocuk babasıdır.