T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DOKTORA TEZİ

UÇAK İNİŞ OPERASYONLARI İÇİN META SEZGİSEL YAKLAŞIMLAR KULLANARAK ONLINE ÇİZELGELEME UYGULAMASI ÇALIŞMASI

Gülnar ÇALIŞKAN

DANIŞMAN
Prof. Dr. Mehmet Mutlu YENİSEY

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

İSTANBUL-2020

Bu çalışma, 8.06.2020 tarihinde aşağıdaki jüri tarafından Endüstri Mühendisliği Anabilim Dalı, Endüstri Mühendisliği Programında Doktora tezi olarak kabul edilmiştir.

Tez Jürisi

Prof. Dr. Mehmet Mutlu YENİSEY(Danışman)
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Prof. Dr. Şakir ESNAF
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Prof. Dr. Selim ZAİM
İstanbul Şehir Üniversitesi
Mühendslik Fakültesi ve Doğa Bilimleri Fakültesi

Prof. Dr. Ş. Alp BARAY
İstanbul Üniversitesi-Cerrahpaşa
Mühendsilik Fakültesi

Prof. Dr. Tufan V. KOÇ
İstanbul Teknik Üniversitesi
İşletme Fakültesi

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi-Cerrahpaşa'nın abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Tez çalışmam boyunca, engin bilgi birikimi, tecrübesiyle yol gösterip, ufkumu aydınlatan danışmanım çok değerli hocam Prof. Dr. Mehmet Mutlu YENİSEY'e hem mesleki hem insani sonsuz desteği, güveni ve sabrı için tüm kalbimle çok teşekkür ederim. Yapıcı eleştilerileriyle daima yanımda olan, katkılarıyla sunumlarda kafamdaki soruları netleştirmemi sağlayan çok değerli hocam Prof. Dr. Şakir ESNAF'a ve katkılarından dolayı saygıdeğer hocam Prof. Dr. Selim ZAİM'e çok teşekkür ederim. Başlangıçta teze başlarken yanımda olan ve her nekadar tez konusunun kapsamı değişse de tez konusunu seçmemde engin bilgi birikimiyle yol gösterip, destek olan yardımcı danışman hocam rahmetli Prof. Dr. Rainer LEISTEN'e ve çok değerli desteği ve katkıları için çok kıymetli hocam Prof. Dr. Alp BARAY'a çok teşekkür ederim.

Bu süreçte her daim yanımda olan canım eşim Mesut Çalışkan'a sınırsız pozitifliğiyle verdiği koşulsuz gönül desteği için minnetarım. Varlıklarından güç aldığım, inanç ve cesaretle yoluma devam etmemi sağlayıp, daima destek olan canım annem, babam ve kardeşime, gönül desteklerini hep hatırlayacağım sevgili kuzenim Şenay Hatipoğlu ve eşi Ercan Hatipoğlu'na, bana olan inançları ve güvenleri için ismini yazamadığım geride kalan tüm aileme ve arkadaşlarıma çok teşekkür ederim.

Haziran 2020 Gülnar ÇALIŞKAN

İÇİNDEKİLER

Sayfa No

ÖNSÖZ ..... **iv**
İÇİNDEKİLER ..... **v**
ŞEKİL LİSTESİ ..... **viii**
TABLO LİSTESİ ..... **ix**
SİMGE VE KISALTMA LİSTESİ ..... **xii**
ÖZET ..... **xiv**
SUMMARY ..... **xvi**
1. GİRİŞ **1**
1.1. UÇAK ÇİZELGELEME PROBLEMİ TANIMI 5
1.2. UÇAK ÇIZELGELEME PROBLEMINDE KARAKTERSTIKLER VE TEMEL KAVRAMLAR 6
1.2.1. Statik ve Dinamik Durum 6
1.2.2. Pist Sayısı ve Pist Belirleme 7
1.2.3. Zaman Penceresi 7
1.2.4. İlk Gelene İlk Hizmet Edilir Kuralı (FCFS) 8
1.2.5. Ayırım Zamanı 8
1.3. ÇALIŞMANIN AMACI VE KAPSAMI 11
2. GENEL KISIMLAR **13**
2.1. OPTİMİZASYON 13
2.1.1. Optimizasyon Modellerinin Sınıflandırılması 14
2.1.2. Optimizasyon Yöntemleri 16
2.1.2.1. Klasik Optimizasyon Modelleri 17
2.1.2.2. Optimizasyon Algoritmaları 18
2.1.2.3. Matematiksel model 19
2.1.2.4. Matematiksel Modelin Bileşenleri 19
2.2. UÇAK ÇİZELGELEME PROBLEMLERİ İÇİN ÇÖZÜM YAKLAŞIMLARI 23
2.2.1. Kesin Çözüm Yöntemleri 23
2.2.2. Yaklaşık Çözüm Yöntemleri 23

2.2.2.1. Sezgisel Yöntemler *23*
2.2.2.2. Metasezgisel Yöntemler *24*
2.3. TEZDE KULLANILAN METASEZGİSEL ALGORİTMALAR 27
2.3.1. Genetik Algoritmalar 27
2.3.2. Tavlama Benzetimi (Simulated Annealing) 29
2.3.3. Tabu Arama 31
2.3.4. Karınca Kolonisi Algoritması 33
2.3.5. Kullanılan Metasezgisel Yöntemlerin Karşılaştırılması 34
2.4. LİTERATÜR TARAMASI 37
3. MALZEME VE YÖNTEM 67
3.1. MATEMATİKSEL MODEL 67
3.2. AKIŞ ŞEMALARI VE SÖZDE KOD 70
3.2.1. Akış Şemaları 70
3.2.2. Sözde Kod 76
3.3. KULLANILAN ALGORİTMALARIN VE L1, L2 OPTİMİZASYONLARI 78
3.3.1. Veri Seti 78
3.4. KULLANILAN ORTAK FONKSİYONLAR 84
3.4.1. Zaman Atama Fonksiyonu 84
3.4.2. Sıralama Doğrulama Fonksiyonu 87
3.4.3. L1 Optimizasyon Fonksiyonu 88
3.4.3.1. Yeni Üretilen 1. Sıralama *89*
3.4.3.2. Yeni Üretilen 2. Sıralama *89*
3.4.3.3. Yeni Üretilen 3. Sıralama *90*
3.4.4. L2 Optimizasyon Fonksiyonu 92
3.5. TABU ARAMA ALGORİTMASI 95
3.5.1. Tabu Arama Döngüsü 96
3.5.2. Tabu Arama Sonuç 99
3.6. TAVLAMA BENZETİMİ ALGORİTMASI 101
3.6.1. Tavlama Benzetimi Sıcaklık Değişim Döngüsü 101
3.6.2. Tavlama Benzetimi Sonuç 103
3.7. GENETİK ALGORİTMA 104
3.7.1. Genetik Algoritma Döngüsü 105
3.7.2. Turnuva Seçimi (Tournament Selection) 106
3.7.3. Çaprazlama (Crossover) 106

3.7.4. Mutasyon 111
3.7.5. Genetik Algoritma Sonuç 111
3.8. KARINCA KOLONİSİ OPTİMİZASYONU ALGORİTMASI 113
3.8.1. Karınca Kolonisi Algoritması Döngüsü 114
3.8.2. Sıralama Üretme Fonksiyonu 115
3.8.3. Feromon Ekleme Fonksiyonu 116
3.8.4. Feromon Buharlaştırma Fonksiyonu 116
3.8.5. Karınca Kolonisi Optimizasyonu Sonuç 117
4. BULGULAR 119
5. TARTIŞMA VE SONUÇ 130
KAYNAKLAR 133
EKLER 142
ÖZGEÇMİŞ 143

ŞEKİL LİSTESİ

Sayfa No

Şekil 1.1: 2010-2019 Yılları Arasında Uçak Gecikmelerinin Nedenleri 3

Şekil 1.2: Kuyruk türbülansı 9

Şekil 2.1: Optimizasyon problemlerinin sınıflandırılması(Büyüksaatçi, 2015).....15

Şekil 2.2: Optimizasyon metotlarının sınıflandırılması(Kıran, 2014) 17

Şekil 2.3: Klasik Optimizasyon Modelleri (Talbi, 2009).....18

Şekil 2.4: Optimizasyon algoritmalarının sınıflandırılması (Büyüksaatçi, 2015).....19

Şekil 2.5: Matematiksel Modelleme Süreci 20

Şekil 2.6: Problemin türüne göre matematiksel model çeşitleri (Eröz, 2020) 22

Şekil 2.7: Metasezgisel Yöntemler (Önal, 2020). 26

Şekil 2.8: Genetik algoritma sözde kodu (Yang, 2010). 29

Şekil 2.9: Tavlama benzetimi algoritması sözde kodu (Yang, 2010). 31

Şekil 2.10: Tabu arama algoritması sözde kodu (Alharkan ve diğerleri, 2019) 33

Şekil 2.11: Karınca Kolonisi Algoritması sözde kodu (Yang, 2010). 34

Şekil 3.1: L1 (Seviye1 Opt.)-L2 (Seviye2 Opt.) Algoritma genel akış şeması.....71

Şekil 3.2: L1 (Seviye1 Opt.), L2 (Seviye 2 Opt.) Tabu arama akış şeması. 72

Şekil 3.3: L1 (Seviye1 Opt.), L2 (Seviye 2 Opt.) Benzetim tavlaması akış şeması. 73

Şekil 3.4: L1 (Seviye1 Opt.), L2 (Seviye 2 Opt.) Genetik algoritma akış şeması. 74

Şekil 3.5: L1 (Seviye1 Opt.), L2 (Seviye 2Opt.) Karınca kolonisi akış şeması.....75

TABLO LİSTESİ

Sayfa No

Tablo 1.1: Uçaklar Arası Minimum Ayırım Süreleri. (saniye olarak) (Ghoniem ve diğerleri, 2014) 10

Tablo 2.1: Metasezgisellerin kullanım alanlarına ve çalışma prensiplerine göre sınıflandırılması (Fink, Voss, 1999) 35

Tablo 2.2: Metasezgisellerin kullanım alanlarına ve çalışma prensiplerine göre sınıflandırılması (Fink, Voss, 1999) (devam) 36

Tablo 2.3: Literatür Özet Tablosu 61

Tablo 2.4: Literatür Özet Tablosu (devam) 62

Tablo 2.5: Literatür Özet Tablosu (devam) 63

Tablo 2.6: Literatür Özet Tablosu (devam) 64

Tablo 2.7: Literatür Özet Tablosu (devam) 65

Tablo 2.8: Literatür Özet Tablosu (devam) 66

Tablo 3.1: Veri Setinin Ayrım Zamanları Bilgisi. 82

Tablo 3.2: Veri setindeki ilk üç uçak (Uçak Kimlik No: 0, 1, 2) için örnekleme 85

Tablo 3.3: TO ve AO'ya göre uçak sıralama. 87

Tablo 3.4: EO, LO, TO ve AO'ya göre uçak sıralama. 87

Tablo 3.5: L1 optimizasyonla TO ve AO'ya göre uçak sıralama. 89

Tablo 3.6: L1 optimizasyonla TO ve AO'ya göre uçak sıralama. 89

Tablo 3.7: L1 optimizasyonla TO ve AO'ya göre uçak sıralama. 90

Tablo 3.8: L1 optimizasyonla TO ve AO'ya göre uçak sıralama. 91

Tablo 3.9: L2 optimizasyonla TO ve AO'ya göre uçak sıralama. 93

Tablo 3.10: L2 optimizasyonla TO ve AO'ya göre uçak sıralama ve ceza maliyetleri 93

Tablo 3.11: L2 optimizasyonla TO ve AO'ya göre uçak sıralama ve ceza maliyeti. 94

Tablo 3.12: L2 optimizasyonla TO ve AO'ya göre uçak sıralama ve ceza maliyeti. 95

Tablo 3.13: Tabu arama algoritması TO ve AO'ya göre uçak sıralama.....96

Tablo 3.14: Tabu arama algoritması TO ve AO'ya göre uçak sıralama ve ceza maliyeti 99

Tablo 3.15: L1 optimizasyonla –Tabu arama algoritmasında TO ve AO'ya göre uçak sıralama ve ceza maliyeti.....100

Tablo 3.16: L2 optimizasyonla –Tabu arama algoritmasında TO ve AO'ya göre uçak sıralama ve ceza maliyeti.....100

Tablo 3.17: Tavlama Benzetiminde TO ve AO'ya göre uçak sıralama 101

Tablo 3.18: Tavlama benzetiminde TO ve AO'ya göre uçak sıralama ve ceza maliyeti 103

Tablo 3.19: L1 optimizasyonla-Tavlama benzetiminde TO ve AO'ya göre uçak sıralama ve ceza maliyeti.....103

Tablo 3.20: L2 optimizasyonla-Tavlama benzetiminde TO ve AO'ya göre uçak sıralama ve ceza maliyeti.....104

Tablo 3.21: Genetik algoritma, TO ve AO'ya göre uçak sıralama.....105

Tablo 3.22: Genetik algoritma çaprazlama işlemi. 107

Tablo 3.23: Genetik algoritma çaprazlama işlemi 107

Tablo 3.24: Genetik algoritma çaprazlama işlemi 108

Tablo 3.25: Genetik algoritma çaprazlama işlemi 108

Tablo 3.26: Genetik algoritma çaprazlama işlemi 108

Tablo 3.27: Genetik algoritma çaprazlama işlemi 109

Tablo 3.28: Genetik algoritma çaprazlama işlemi 109

Tablo 3.29: Genetik algoritma çaprazlama işlemi. 110

Tablo 3.30: Genetik algoritma çaprazlama işlemi. 110

Tablo 3.31: Genetik algoritma çaprazlama işlemi 111

Tablo 3.32: Genetik algoritmada TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti 111

Tablo 3.33: L1 optimizasyona- Genetik algoritmada TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti.....112

Tablo 3.34: L2 optimizasyonal-Genetik algoritmada TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti.....112

Tablo 3.35: Karınca kolonisi optimizasyonunda TO ve AO'ya göre uçak sıralama.....114

Tablo 3.36: Karınca kolonisi optimizasyonunda TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti 117

Tablo 3.37: L1 optimizasyona- Karınca kolonisi optimizasyonunda TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti 117

Tablo 3.38: L2 optimizasyonla-Karınca kolonisi optimizasyonunda TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti. 118

Tablo 4.1: Dinamik L1 Optimizasyonu Uygulanan OR Kütüphanesi Küçük Problemler Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Dinamik) 122

Tablo 4.2: Dinamik L1 Optimizasyonu Uygulanan OR Kütüphanesi Büyük Problemler Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Dinamik) 123

Tablo 4.3: Dinamik L2 Optimizasyonu Uygulanan OR Kütüphanesi Küçük Problemler Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Dinamik) 124

Tablo 4.4: Dinamik L2 – Optimizasyonu Uygulanan OR Kütüphanesi Büyük Problemleri Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Dinamik).....125

Tablo 4.5: Dinamik L1 – L2 Optimizasyonu Uygulanan OR Kütüphanesi Küçük-Büyük Problemler Karşılaştırma Tablosu.(Beasley-Statik, Geliştirilen Algoritma Dinamik).....126

Tablo 4.6: Statik L1 Optimizasyonu Uygulanan OR Kütüphanesi Küçük Problemler Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Statik).....127

Tablo 4.7: Statik L2 Optimizasyonu Uygulanan OR Kütüphanesi Küçük Problemler Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Statik).....128

Tablo 4.8: Statik L1 – L2 Optimizasyonu Uygulanan OR Kütüphanesi Küçük Problemler Karşılaştırma Tablosu (Beasley-Statik, Geliştirilen Algoritma Statik).....129

SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
a_i	: i uçağının gecikme süresi
b_i	: i uçağının havaalanına erken gelme süresi
c_i	: i uçağının geç kalkış süresi
d_i^+	: i uçağı için geç iniş maliyeti
d_i^-	: i uçağı için erken iniş maliyeti
E_{D_i}	: i uçağı için en erken kalkış zamanı
E_{L_i}	: i uçağı için en erken iniş zamanı
f_i^+	: i uçağı için geç kalkış maliyeti
g_{ij}	: i uçağının iniş zamanına göre alacağı değer (1, 0)
h_{ij}	: i uçağının kalkış zamanına göre alacağı değer (1, 0)
$f(x_i)$	: O andaki mevcut sıralamanın ceza maliyeti değeri
$f(x_j)$	: Yeni üretilmiş olan sıralamanın ceza maliyeti değeri
L_{L_i}	: i uçağı için en geç iniş zamanı
L_{D_i}	: i uçağı için en geç kalkış zamanı
P	: Uçak sayısı
P_T	: Sıralamanın kabul edilme olasılığı
ρ	: Rho katsayısı
S_{ij}	: arka arkaya inen i ve j uçakları arasında gerekli olan ayrım zamanı
T_{L_i}	: i uçağı için hedef iniş zamanı
T_{D_i}	: i uçağı için hedef kalkış zamanı
T	: O andaki sıcaklık değeri
τ_{ij}	: i ve j uçak ikilisi için atanan feromon değeri
w_i	: i uçağının büyüklüğüne ve operasyon tipine bağlı değişen katsayısı
x_i	: i uçağı için gerçekleşen iniş zamanı
y_i	: i uçağı için gerçekleşen kalkış zamanı

Kısaltmalar	Açıklama
AO	: Atanan Operasyon Zamanı (Assigned Operation Time)
AD	: Uçak Algılanma Zamanı (Aircraft Detected Time)
ALS	: Uçak İniş Problemi
ASP	: Uçak Sıralama Problemi
B&B	: Dal ve Sınır
BTS	: Amerika Ulaştırma İstatistikleri Bürosu
DP	**:** Dinamik Programlama
EO	: En Erken Operasyon Zamanı (Earliest Operation Time)
EP	: Erken Operasyon Cezası (Early Operation Penalty)
GA	**:** Genetik Algoritma
LO	: En Geç Operasyon Zamanı (Latest Operation Time)
LP	: Geç Operasyon Cezası (Late Operation Penalty)
L1	**:** 1. Seviye Optimizasyon
L2	**:** 2. Seviye Optimizasyon
MILP	: Karışık tamsayılı doğrusal programlama
MIP	: Karışık-tamsayı programlama
NAS	: Amerika Ulusal Havacılık Sistemindeki
SLT	: Uçak planlanan iniş zamanından
STT	: Kalkan uçaklar için planlanan kalkış zamanından
TCA	: Terminal Kontrol Alanı'ndan
TMA	: Terminal manevra alanını
TO	: Hedeflenen Operasyon Zamanı (Target Operation Time)
UÇP	: Uçak Çizelgeleme Problemi

ÖZET

UÇAK İNİŞ OPERASYONLARI İÇİN META SEZGİSEL YAKLAŞIMLAR KULLANARAK ONLINE ÇİZELGELEME UYGULAMASI ÇALIŞMASI

DOKTORA TEZİ

Gülnar ÇALIŞKAN

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman : Prof. Dr. Mehmet Mutlu YENİSEY

Havaalanlarının dinamik doğası nedeniyle, yeni ve dinamik trafik olaylarının yeniden planlanması için hesaplama açısından etkili ve uygun olan çizelgeleme algoritmaları geliştirmeye ihtiyaç vardır.

Literatüre göre, uçak çizelgeleme problemleri (ASP) neredeyse herzaman statik durum için ele alınmıştır. Bu çalışma, özellikle dinamik durum için tek pistte uçak iniş problemini çözmek için 4 algoritma önermektedir. Uçak iniş problemini çözmek için 4 meta-sezgisel algoritma (Tabu Arama, Tavlama Benzetimi, Genetik Algoritma ve Karınca Kolonisi Optimizasyonu) özgün, seviye1 (L1) ve seviye 2 (L2) optimizasyonları kullanılarak hibrid algoritma haline getirilmiştir. Çalışmada uçak çizelgeleme probleminin esas amacı, iniş ve kalkış süresi boyunca hedef sürelerden toplam sapma maliyetinin en aza indirilmesidir. Önerilen çözüm yaklaşımının sayısal değerlendirmesi için OR kütüphanesinin test örnekleri kullanılmıştır. Problem, sıralama ve zaman ataması olarak iki kısma ayrılmıştır. Zaman atama fonksiyonu, zaman değerlerinin atanmadığı uçaklar için operasyon süresini atar. Zaman atama fonksiyonu ayrıca L1 ve L2 optimizasyon işlemleri kullanılarak yapılan optimizasyon sürecini de içerir. Algoritmaların çözümleri optimum karışık tamsayı çözümleri ve Beasley'in elde ettiği sonuçlar ile karşılaştırılmıştır. Performansları, uçak sıralamalarının ceza maliyetleri açısından değerlendirilmiştir. Problem, uçak çizelgeleme problemi için uygun çözümler üretebilen C

programlama diline uygulanmıştır. Sorunun çözümü, uçak çizelgeleme problemlerinin hem statik hem de dinamik durumu için kullanılabilir. Statik ve dinamik durum sonuçları Beasley'in makalelerinin uçak iniş çizelgeleme problemleri (ALS) için statik durum sonuçları ile karşılaştırılmıştır.

Çalışmanın genel sonuçları, küçük veri setleri için büyük çoğunlukla optimum sonuçlara ulaşıldığını göstermiştir. Büyük veri setleri için ise bilinen en iyi çözüm sonuçları (Z-best) ya da bilinen en iyi çözüme yakın sonuçlar elde edilmiştir. Parametre ayarına devam edilerek büyük veri setleri için perfomanslar daha da iyileştirilebilir. Küçük test problemlerinde, önerilen optimize edilmiş algoritmalar dinamik durumda çalıştırılsa dahi neredeyse herzaman Beasley'in statik durumda elde ettiği sonuçlardan daha iyi sonuçlar elde edilmiştir.

Haziran 2020, 161 sayfa.

Anahtar kelimeler: Uçak Çizelgeleme Problemi (UÇP), Uçak İniş Problemi (ALS), Tabu Arama, Genetik Algoritma, Tavlama Benzetimi, Karınca Kolonisi Optimizasyonu

SUMMARY

ONLINE SCHEDULING USING METAHEURISTICS APPROACHES WITH APPLICATIONS FOR AIRCRAFT LANDING OPERATIONS

Ph.D. THESIS

Gülnar ÇALIŞKAN

Istanbul University-Cerrahpasa

Institute of Graduate Studies

Department of Industrial Engineering

Supervisor : Prof. Dr. Mehmet MUTU YENİSEY

Due to the dynamic nature of the airports there is a need to develop scheduling algorithms that are computationally effective and proper for replanning new and dynamic traffic events occur.

According to the literature, airplane scheduling problems (ASP) are almost always considered for the static case. This study proposes 4 algorithms to solve aircraft landing problem on single runway for especially dynamic case. To solve the aircraft landing problem, 4 metaheuristic algorithms (Tabu Search, Simulated Annealing, Genetic Algorithm and Ant Colony Optimization) have been turned into a hybrid algorithm by using unique level1 (L1) and level 2 (L2) optimizations.

The main objective of the aircraft scheduling problem in this paper is the minimization of deviations' total cost from the target times during landing and take-off time. For the numerical evaluation of the proposed solution approach, the OR library's test instances are used. The problem is divided in two parts as sequencing and time assignment. The time assignment function assigns operation time for the aircraft in which the time values are not assigned. Time assignment function also includes optimization process which is done by using L1 and L2 optimization operations. The solutions of the algorithms are compared with the optimum mix

integer solutions and Beasley's results. Their performances are evaluated in terms of penalty cost of aircraft sequencing. The problem implemented in C programming language which can be used to calculate feasible solutions for the aircraft scheduling problems. The solution of the problem can be used for both static and dynamic cases of the aircraft scheduling problems. The static and dynamic case results are compared with Beasley's papers' static case results for the aircraft landing scheduling problem (ALS).

The overall results of the study indicate that optimum results have been mostly achieved for small data sets. The best known (Z-best) solution results or nearly the best known results are obtained for large data sets. Performances can be further improved for large data sets by continuing the parameter setting. Even if the proposed optimized algorithms are run in dynamic case, they almost always have better results than Beasley's results in static case for small data sets.

June 2020, 161 pages.

Keywords: Aircraft scheduling problem (ASP), Aircraft landing problem (ALP), Tabu Search, Genetic Algorithm, Simulated Annealing, Ant Colony Optimization

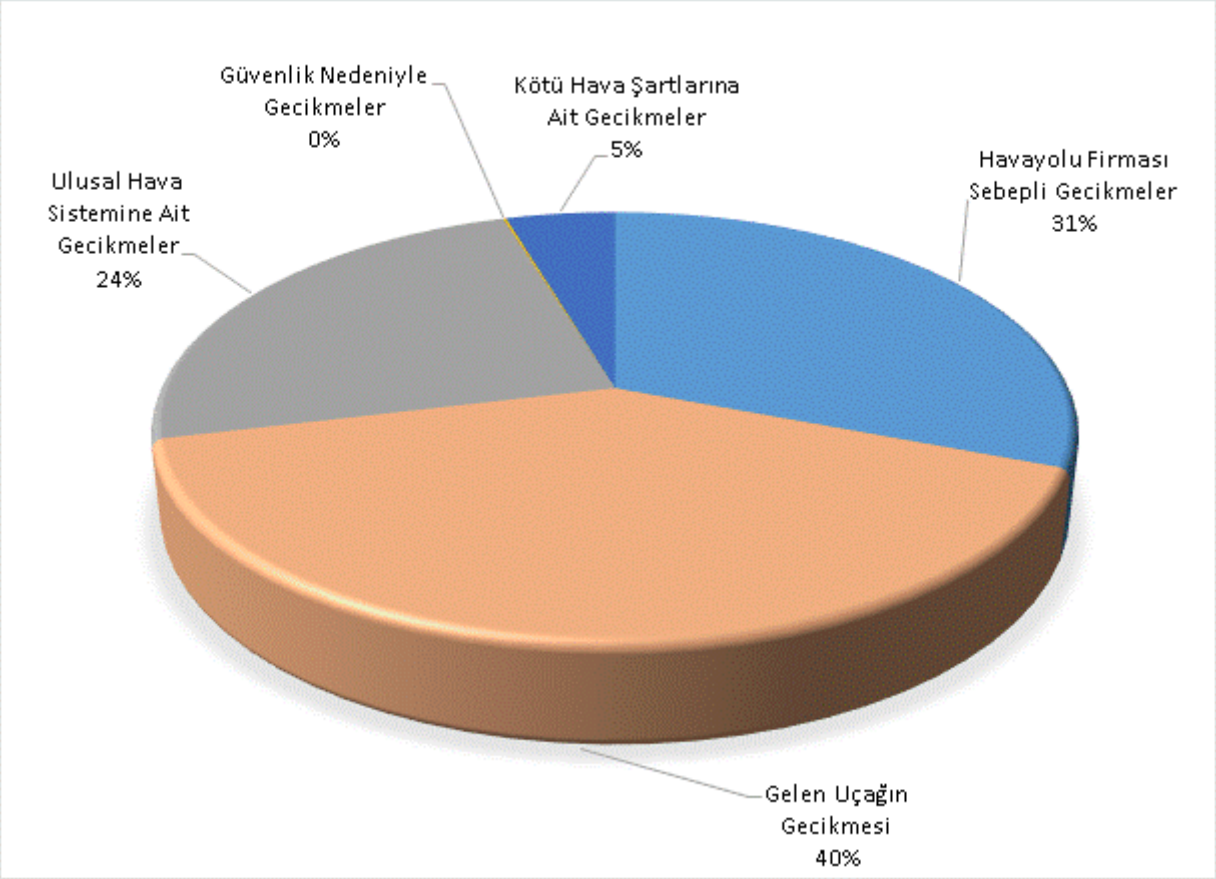
1. GİRİŞ

Özellikle son yıllarda hava trafiğindeki artış nedeniyle, havaalanlarının mevcut kapasitesi yetersiz kalmaktadır. Ekonomik büyüme, hava taşımacılığı talebini büyük ölçüde etkilemekte ve endüstrinin altyapı ve kaynak kapasitelerinin sınırlarına ulaşmasına zorlamaktadır. Bununla birlikte, altyapı açısından, en ihmal edilen alanlardan biridir (Wensveen, 2015). Hava trafik yoğunluğunun giderek artan bir ivmeyle daha da artacağı açıktır. Bu nedenle, havaalanlarına yaklaşan uçakların sayısı havaalanı pist kapasitesini aştığında uçakların bir kısmı inmeleri gereken hedef zamanda iniş yapamazlar ve gecikmeler kaçınılmaz hale gelir. Bazı uçaklar da yine kapasite kısıtı nedeniyle kalkmaları gereken zamanda kalkışa geçemeyebilirler. Ayrıca uçaklar, havaalanlarına zamanında varabilmek için ekonomik hızlarından daha hızlı uçtuklarından daha fazla yakıt harcarlar ve bu durum da yakıt maliyetlerinin yüksek olmasına sebep olmaktadır. Sonuç olarak, kaynak kapasitesini aştığı için hava taşımacılığı tesislerinde gecikmeler kaçınılmaz hale gelmektedir. Ortaya çıkan yüksek hacimli hava trafiği hem havalimanında tıkanıklıklara hem de gidiş ve gelişler için uzun kuyruklara yol açar ve bu durum da ek yakıt maliyetleri, yolcu memnuniyetsizliği ve çevre kirliliği gibi sorunlara yol açar.

Havaalanlarında mevcut kapasiteyi daha iyi kullanma ve kapasite kullanım oranlarını arttırma ihtiyacı ve önemi artmıştır. Havaalanlarındaki en büyük darboğazlardan birini birincil sınırlama faktörü (darboğaz) olarak pistler oluşturmaktadır ve pistler gibi darboğazların verimli bir şekilde kullanılmasını sağlamak büyük önem arz etmekdtedir. Çünkü bir havaalanının kapasitesi, pistlerin etkin kullanımına bağlıdır. Havaalanlarında, sistemlerin kapasite kullanım oranını arttırmanın en uygun yolu, havalanına gelen ve kalkan uçakların çizelgelemelerini iyileştirmektir ve bu tezin esas amacıdır. Böylece, ek pistlere ihtiyaç olmayabilir ve havaalanlarına yapılması gereken pahalı yatırımlar önlenebilir. Havaalanlarının toplam kapasitesini artırmak ve hava trafiğinin akışını düzeltmek için terminal manevra alanını (TMA) etkili bir şekilde kullanmak için gerekli düzenlemeler yapılmalıdır.

Terminal manevra alanı kullanımını geliştirmek ve hava trafiği akışını güvenli şekilde yürütebilmek için pist operasyonlarının etkin ve verimli bir şekilde çizelgelenmesi gerekmektedir. Pist kullanımını etkileyen ana faktörlerden biri, güvenlik hususlarından kaynaklanan uçaklar arasında asgari ayrımların uygulanmasıdır. Tüm uçaklar uçuşları sırasında arkalarında hava girdapları (wake vortices) oluştırduğundan; yapılan uçuş çizelgeleri, olumsuz etkileri önlemek için, ard arda uçan uçaklar arasındaki ayırım zamanları düzenlemelerini karşılamalıdır. Hava girdapları uçak büyüklüğüne (büyük, orta, küçük uçak), operasyon tipine (kalkış, iniş) ve ilgili uçak sınıfına (öncü ve ardıl) bağlıdır. Hava girdapları, eğer daha büyük bir uçak tarafından yaratıldıysa daha büyüktür. Ayrıca, takip eden uçak öncü uçaklardan daha küçük olduğunda daha büyük bir etkiye sahiptir.

Amerika Ulaştırma İstatistikleri Bürosu (BTS), hava trafiği gecikmelerini 5 grupta toplayıp açıklamaktadır. Bunlar; Havayolu firması sebepli gecikmeler, gelen uçağın gecikmesi, ulusal hava sistemine ait gecikmeler, güvenlik nedeniyle gecikmeler ve kötü hava şartlarına ait gecikmelerdir. Aşağıdaki şekil, 2010-2019 yılları arasında Amerika Ulusal Havacılık Sistemindeki (NAS) uçuş gecikmelerinin nedenlerini göstermektedir. Buna göre gelen uçağın gecikmesine yani aynı uçak geç geldiği için geç kalkmasına ait nedenler %40'la ilk sırada yer almaktadır. Daha sonra ise %31 ile hava yolu firmasına ait sebeplerden kaynaklanan gecikmeler bulunmaktadır.



Şekil 1.1: 2010-2019 Yılları Arasında Uçak Gecikmelerinin Nedenleri (https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations)

Çizelgeleme, kaynakların verilen sürelerde görevlere tahsis edilmesine odaklanan bir karar verme sürecidir. ASP (Uçak Çizelgeleme Problemi) şöyle tanımlanabilir: Bir uçak grubu kapılarda (gate) ya da pistlerde ve kalkışa hazırlanırken, başka bir uçak grubu havaalanına hava yoluyla yaklaşmakta ve inmeye hazırlanmaktadır. Her uçak, büyüklüğüne ve ağırlığına bağlı olarak bir uçak sınıfına aittir ve bir zaman dilimi içinde iniş veya kalkış için bir hedef süreye sahiptir. Eldeki karar sorunu, her uçağa bir pist ve bir kalkış veya iniş zamanı atamaktır. Ortaya çıkan pist programının, operasyon tipine, yani kalkış veya inişe, önceki ve sonraki operasyonların ilgili uçak sınıfına bağlı oldukları için, sıraya bağlı olan ayırma gerekliliklerini karşılaması gerekir. Bir operasyon geciktiğinde, ilgili operasyon sınıfına, yani operasyon tipine ve uçak sınıfına ve gecikmenin uzunluğuna bağlı olarak bir gecikme cezası uygulanır. (Lieder, Stolletz, 2016)

Problemlerin çözünürlük zamanları problem büyüklüğü arttıkça uzayabilirken, küçük boyutlu problemler ise kesin çözüm yöntemleri ile çözülebilir. Uçak çizelgeleme problemi karmaşık ve NP-zor bir problem çeşididir. Bu nedenle, uçak çizelgeleme problemini çözmek için bu tezde 4 metasezgisel algoritma (Tabu Arama, Genetik Algoritma, Tavlama Benzetimi, Karınca Kolonisi Optimizasyonu) ile birlikte, 2 seviyeli özgün optimizasyon yöntemleri kullanılmıştır. Besley'in geliştirmiş olduğu OR kütüphanesindeki gelen uçaklar için test verileri kullanılarak, Beasley'in(Beasley ve diğerleri 2001) makalesinde bulmuş olduğu sonuçlarla kıyaslanmıştır.
Optimal çözümler bulmak için karışık tamsayılı doğrusal programlama (MILP) modeli geliştirilmiştir. Ancak, tüm ağırlıkları eşit olan tek bir tezgahta bile toplam ağırlıklı gecikmeyi en aza indirgemek NP-zor bir problem (Lawler, 1982) olduğundan, ASP de NP-zordur, bu da büyük ölçekli örnekleri makul bir sürede çözmenin hesaplamalı olarak zor olduğu anlamına gelir. Bu nedenle, makul hesaplama zamanlarında iyi kalitede çözümlere ulaşmak için uygun yöntemler geliştirmek gereklidir.
Havayolu şirketleri için yolcu memnuniyeti, uçuş gecikmelerini en aza indirilerek en üst düzeye çıkarılabilecek önemli hususlardan biridir. Günlük operasyonlar boyunca, şiddetli hava koşulları ve beklenmeyen uçak veya personel arızaları gibi beklenmeyen çeşitli olayların bir sonucu olarak, planlanan uçuş programında sapma potansiyeli her havayolu için söz konusudur. Havaalanı operasyonları genel anlamda kara ve hava tarafı operasyonları olarak ikiye ayrılır. Kara tarafı operasyonları, direk olarak uçağa yolcu erişimi konusunda operasyonlardan oluşmaktadır.
Hava tarafı operasyonları daha karışıktır ve bütünsel olarak ele alınmasının zorluğu açısından çeşitli birimlere ayrılır. Birimleri esas hedefi ise operasyonların güvenliği ile sistemin performansını arttırmaya yöneliktir. Hava trafik kontrolörleri, hava alanları büyük olduğunda genel olarak dört kontrolör grubuna ayrılır: (1) kapı kontrolörü (uçağa kapı atar ve geri itme izni verir), (2) rampa kontrolörü (rampa için boşluk sağlar ve rampadaki uçakları sıralar), (3) yer kontrolörü (taksi boşluklarını düzenler ve kalkış / varış taksi kuyruğunu düzenler) ve (4) yerel (kule) denetleyici (gelen / giden uçakların iniş / kalkışları için pist ve başlangıç zamanlarını atar).
Havaalanlarında kontrolörler pistlerdeki gecikmeleri ve tıkanıklığı en aza indirmeye çalışırken, hava trafiğinin güvenli ve düzgün akışını sağlamaktan sorumludurlar. Özellikle yoğun saatlerde pist operasyonlarının hacmi, bazı aksaklıklarla birlikte belirsizlikler nedeniyle çok karmaşık ve

dinamik bir çalışma ortamına neden olur. Ancak, yerel denetleyiciler bu kritik zamanlama işleminde genellikle düşük düzeyde otomasyon kullanır. İdeal bir durumda, yerel kontrolörler daha geniş bir bilgi sistemine ve veritabanlarına bağlı bir karar destek sistemi aracılığıyla bu çizelgeleme işini etkileşimli olarak yürütmelidir. Bu karar destek sistemlerinde pist operasyonları planlaması için kullanılacak etkin algoritmaların geliştirilmesi önemli ve aktif bir araştırma alanıdır.

Ayrıca, şu anda uygulamada, yerel kontrolörler, başlangıçta planlanan uçak programlarını planlanmamış olay meydana gelmesi ile güncelleyerek belirsizliklerle reaktif bir yaklaşımla ilgilenmektedir. Bununla birlikte, TMA (Terminal Manevra Alanı) son derece yoğun bir ortamdır ve yeniden planlanan uçak yerel denetleyicilerin iş yükünü büyük ölçüde arttırır ve ayrıca ilk planlara yapılan her müdahale, tüm hava trafik akışını bir şekilde etkiler. Tüm uçak ayrımlarına tampon (gevşeme) süreleri eklenmesinin, belirsizliklere dayanıklı çözümler üretebileceği düşünülebilir; bununla birlikte, bu tür çizelgeler yetersiz çözümlere ve pistlerin yetersiz kullanılmasına ve havaalanı kapasitesinin kullanımında düşmeye yol açacaktır. Bu nedenle, sağlam çizelgeler elde etmek ve pistlerin daha iyi kullanılmasını sağlamak için belirsizlik modellere açıkça dahil edilmelidir. (Soykan, 2016)

Mevcut hava trafiği akış operasyonlarında; hava taşıtlarını sıralama, iniş / kalkış zamanlarını planlama ve her bir hava aracına gümrükleme talimatlarını verme, uçaklar arasında minimum ayırımları korumak için gerektiğinde programları ayarlarlama işlemleri büyük ölçüde yerel kontrolörler tarafından yürütülür. (Soykan, 2016)

1.1. UÇAK ÇİZELGELEME PROBLEMİ TANIMI

Literatüre göre uçak çizelgeleme probleminde bir dizi uçak ya havalanında kapılardadır yadahava sahasındadır veya kalkışa hazırlanmaktadır. Başka bir uçak seti de hava limanına hava yoluyla yaklaşmakta ve inmeye hazırlanmaktadır. Her uçak, boyutuna ve ağırlığına göre bir uçak sınıfına aittir ve bir zaman aralığında kalkış veya iniş için hedef süreye sahiptir. Burdaki karar problemi; her bir uçağa iniş, kalkış zamanı veya pist atamaktır (Lieder ve diğerleri, 2016) Kısaca, Uçak Çizelgeleme Problemi (UÇP), gelen ve kalkan uçakları pistlere atamak ve aynı zamanda her pistte aynı anda sıralamak olan ve bunları belli bir çizelge ile sıralayan, polinom olmayan süreli zor (NP-zor) bir yöneylem araştırma problemi olarak tanımlanabilir. Ardışık ve ardışık olmayan işlemler arası ayırım süreleri ve işleme alınacak

operasyonlar arası zaman pencereleri, uçakların çizelgelenmesi konusundaki en temel gereksinimlerindendir. Kuyruk türbülansı etkisi olasılığını azaltmak için, ardışık ve ardışık olmayan belirli işlemler arasında minimum ayırım süreleri gereklidir.

Uçak Çizelgeleme Problemleri büyük ölçeklidir ve 3 adımdan oluşmaktadır denilebilir. İlk adımda uçakları havalanında mevcut bulunan pistlere atanır. İkinci adımda pistlere atanan uçakların sıralanması gerçekleştirilir. Üçüncü adımda ise, bu adımlar sırasındaki her işlemin başlama zamanı belirlenir.

Uçakların belirlenen zamandan önce veya sonra yere inişi ek maliyet oluşturur.

En erken uçuş zamanı, uçağın en yüksek hızda uçtuğunda havada bulunduğu zamandır. En geç uçuş zamanı ise, izin verilen maksimum uçuş zamanını havada geçiren ve uçağın sahip olduğu yakıta bağlı olarak havada kalabileceği en fazla süreyi kapsayan zamandır. Bu süre bitince uçak piste iniş yapmalıdır. (Aydın, 2009)

Her uçak, en ekonomik yakıt tüketiminin sağladığı bir hıza sahiptir. Bu uçuş hızı, yolculuk boyunca uçağın seyrettiği ekonomik hızdır. Uçak, maksimum hızla uçtuğundaki uçuş zamanı, en erken uçuş zamanı olarak tarif edilir. Uçağın tercih edilen veya hedeflenen uçuş zamanı, ekonomik hızla uçmasını gerektirerek yere inmesini sağlar. Eğer hava trafik kontrolörü, uçağın hızını kesmesini, aynı hızla uçuşa devam etmesini ya da hızını artırmasını isterse, bu durum beraberinde ek maliyetleri de meydana getirir. (Aydın, 2009)

1.2. UÇAK ÇIZELGELEME PROBLEMINDE KARAKTERSTIKLER VE TEMEL KAVRAMLAR

1.2.1. Statik ve Dinamik Durum

Statik durumda hesaplama, bir veya birden fazla piste inmeyi bekleyen uçaklar için en uygun sıralamayı planlamak için kullanılır. Statik durumda çizelgeleme işlemi genel olarak, farklı uçaklar için pist kullanımını maksimuma çıkarmak veya hedeflenen iniş zamanları ile gerçekleşen iniş zamanları arasındaki farkı en aza indirmek için ne zaman inmeleri gerektiğinin planlanmasında yapılır. Çizelgeleme işlemi, hesaplamadaki herhangi bir uçak havaalanına yakın olmadan yapıldığı için hesaplama süresi üzerindeki kısıtlar azdır. Ayrıca uçak sayısı sabit olduğu için en uygun çözümü bulmak mümkündür. Uçak çizelgeleme problemini tanımlayan makalelerin çoğu; araştırılan durum, hazır süreler, hedef süreler ve bitiş süreleri dahil tüm verilerin önceden bilindiği ve süreçte dikkate alınabileceği varsayıldığı için statik durumla

ilgilidir. Genel olarak bir hava kontrolörü tarafından planlanan bir dizi iniş analiz edilir ve daha sonra yazarlar bazı buluşsal yöntemlere dayalı olarak daha iyi bir çözüm ararlar. Makalelerin bazılarında, rastgele üretilen uçaklara ve üzerine inebilecekleri sabit sayıda piste dayanan kesin bir teorik çözüm de vardır.

Statik durumda, bilinen veya tahmin edilen bilgilerle gelen uçak operasyonları gerçekleşmeden önce problem çözülürken dinamik durumdaki problemlerde ise uçak gerçek zamanlı olarak radar sahasında geldikçe, gerçek zamanlı olarak zaman çizelgeleri oluşturularak çözülür. Dinamik durumda hızlı hesaplama süreleri hem gerekli hem çok öenmlidir. Ayrıca dinamik durum için kullanılan modelleme yaklaşımları, statik durum için kullanılan modellemeyaklaşımlarından oldukça farklılık gösterir. Bu durum ise, hangi uçağın yeniden planlanacağı ve uçağın sırasının ne zaman yeniden planlanacağı hakkında bazı bilgilerin göz önüne alınmasını gerekli hale getirir.

1.2.2. Pist Sayısı ve Pist Belirleme

Pist kapasiteleri uçakların çizelgelenmesi konusunda bir havalanındaki en önemli kısıtlardan biridir. Pist kapasiteleri; uçak tipi, hava koşulları, pist sayısı (tekli-çoklu pist), etkileşimli-bağımsız pistler ya da pist operasyon türüne göre tekli-karışık mod olmasına göre uçak çizelgeleme problemlerine etki eder. Bağımsız pistlerde uçaklar arası ayırım süreleri önemli olmadığı halde, etkileşimli pistlerde pistin birindeki uçaklar arası ayırım süreleri diğer pistteki operasyondan etkilenir. Uçak operasyonlarında ise tekli modun geçerli olduğu durumda sadece gelen ya da sadece kalkan uçaklar için çezelgeleme yapılırken, karışık mod için geçerli olan durumda hem gelen hem de kalkan uçaklar aynı pisti kullandığı için ona göre çizelgeleme yapılır. Tezde de kullanılan şekliyle uçak operasyon türünün karışık mod olması durumunda havalanı kapasite kullanım oranı daha fazladır. Pist sayısının arttırılması bu durumu daha da olmlu yönde etkiler.

1.2.3. Zaman Penceresi

Belirlenen durumlar çerçevesinde her uçağın iniş ya da kalkşı için en erken ve en geç olmak üzere olası bir zaman penceresi mevcuttur ve bu zaman çerçevesine göre iniş ya da kalkış işlemine başlama süreleri belirlidir. Zaman pencereleri belirlenirken; yakıt maliyeti, uçağın

maksimum uaçabileceği hız, pistlerin uygunluğu, gelen uçaklar için bağlantılı uçuşların olması, maksimum gecikme ihtimali olan süre ve hava koşulları gibi durumlar göz önünde bulundurulur. En geç iniş zamanında uçağın bulundurduğu yakıt miktarı önemli iken, uçak ekonomik uçuş hızında uçarken yakıt harcaması ve dolayısıyla maliyeti minimum olmktadır. Ticari uçaklarda da hedef iniş zamanı belirlenirken bu durum göz önünde bulundurulur.

Zaman pencereleri her ne kadar gelen uçaklar için önemli olduğu düşünülde de havaalanlarından kalkan uçaklar için de son derece önemlidir. Kalkan uçaklar için planlanan kalkış zamanından (STT) 5 dakika önce ve 10 dakika sonra kalkış için bir zaman aralığı düşünülür. Kalkış yapan uçağın, planlanan kalkış zamanından 10 dakikadan daha uzun ir süre sonra pistten ayrılması durumunda Terminal Kontrol Alanı'ndan (TCA) çıkmada gecikmiş olduğu düşünülür. Eğer gelen uçak planlanan iniş zamanından (SLT) sonra iniş yaparsa gecikmiş olur (Samà ve diğerleri, 2017).

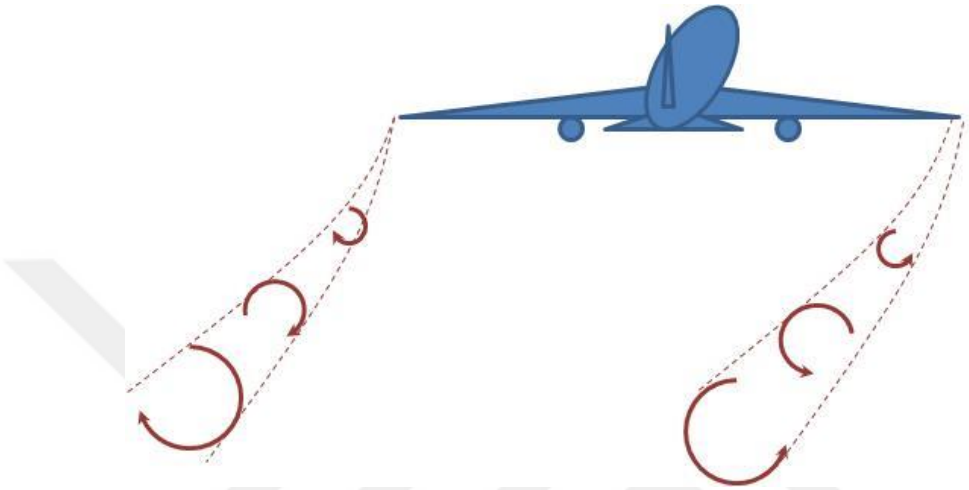
1.2.4. İlk Gelene İlk Hizmet Edilir Kuralı (FCFS)

Uçak çizelgeleme problemi için basit ve pratik olması nedeniyle en yaygın kullanılan yöntemlerden biri ilk gelen uçağa ilk hizmet edilir kuralı (FCFS)'dır. FCFS, uçak iniş sırasını piste tahmini varış zamanı (ETA) sırasına göre çözer. ETA, gelen bir uçağın hava sahası sınır çizgisini geçtiğinde kontrol merkezi aracılığıyla hesaplanır. Birbirini takip eden iki uçağın ETA'sı arasındaki sapma en düşük ayırma süresi kısıtlamalarına aykırı ise, takip eden uçağın Planlı Varış Zamanı (STA) buna göre hazırlanır (Mahmud, 2017). FCFS yöntemi, adaletli olduğu düşünüldüğü ve tahmini iniş zamanları ve gelen uçaklardaki gecikmelerin hesaplanmasında olduğu gibi hava trafik kontrolörlerinin iş yükünü hafiflettiği için en uygun yöntem gibi görülse de pist kullanım oranı dahil pek çok açıdan en uygun yöntem değildir.

1.2.5. Ayırım Zamanı

Federal Havayolları Yönetimi (FAA), inen uçaklar arasında uçak kuyruk türbülansı tehlikesi arasında minimum boşluk gereksinimleri belirler. Bir uçak, iniş yapan uçağın kuyruk girdabı ile etkileşime girerse dengeyi sağlayamama riskiyle karşı karşıya kalır. Bunu önlemek için, uçaklar arasında inen ve takip eden uçağın büyüklüğüne bağlı olarak minimum bir boşluk olmalıdır (Balakrishnan, Chandran, 2016) Ayırım sürelerini (Minimum takip süresi) uçakların aerodinamik yapıları belirlemektedir.

Uçak kuyruk türbülanslarının büyüklüğü, uçağın kütlesi arttıkça artar. Kuyruk türbülanslarının takip eden uçaklarda uçuş emniyetini bozduğundan, pekçok uçak kazasına neden olduğu belirtilmiştir.



Şekil 1.2: Kuyruk türbülansı
(https://www.skybrary.aero/index.php/Wake_Vortex_Turbulence)

Kuyruk girdaplarına etkilerinin farklı olması nediniyle, uçaklar büyüklüklerine göre sınıflara ayrılmaktadır. FAA tarafından belirlenen arda arda gelen uçaklar arası minimum ayırım süreleri, uçakların operasyon türü (havaalanına gelen, kalkan) ve uçak büyüklüğüne bağlantılıdır. İnen uçaklar için ayırma standardı ard arda gelen uçaklar arası mesafeye dayanırken, kalkan uçaklar için zamana dayanmaktadır (Beasley ve ark. 2001). Sınıflar arasındaki ayrım zamanları farklılık gösterdiğinden değişik sınıflardan uçakların oluşturduğu bir kümenin iniş için sıralandırılması neticesinde, farklı sıralandırmalar için toplam iniş zamanında farklılıklar oluşmaktadır. İyi bir sıralandırma kararı havalimanının pist kapasite kullanım oranını artırmaktadır. (Aydın, 2009)

Zorunlu minimum takip zamanlarını belirlemek için yetkili makamlar (Federal Aviation Administration – ABD , Civil Aviation Authority - İngiltere, Eurocontrol - Avrupa) uçakları temel üç veya dört kategori altında sınıflandırmış ve bu sınıflar arasındaki minimum takip zamanlarını belirlemiştir. Literatürdeki bir çok çalışma bu sınıflandırma temel alınarak yapılmıştır.

Tablo 1.1: Uçaklar Arası Minimum Ayırım Süreleri. (saniye olarak) (Ghoniem ve diğerleri, 2014)

Geliş – Kalkış Durumu			
Öncü / Takip Eden	Büyük	Orta	Küçük
Büyük	40	40	40
Orta	35	35	35
Küçük	30	30	30
Geliş – Geliş Durumu			
Öncü / Takip Eden	Büyük	Orta	Küçük
Büyük	99	133	196
Orta	74	107	131
Küçük	74	80	98
Kalkış – Kalkış Durumu			
Öncü / Takip Eden	Büyük	Orta	Küçük
Büyük	60	90	120
Orta	60	60	90
Küçük	60	60	60
Kalkış – Geliş Durumu			
Öncü / Takip Eden	Büyük	Orta	Küçük
Büyük	50	53	65
Orta	50	53	65
Küçük	50	53	65

1.3. ÇALIŞMANIN AMACI VE KAPSAMI

Uçak çizelgeleme problemlerine genel olarak bakıldığında literatürde farklı amaçlar kullanılmıştır. Uçak navigasyon servis sağlayıcıları (ANSP) kurumuna göre en yaygın olarak kullanılan amaçlar, toplam gecikmeyi (gecikme) en aza indirgemek, hedeflenen süreden toplam sapmayı en aza indirgemek (erkencilik ve gecikme), uçak başına ortalama gecikmeyi en aza indirgemek, pist kullanım oranınını yani verimi en üst düzeye çıkarmak ve maksimum gecikmeyi en aza indirmektir (Soykan, 2016).

Uçak çizelgeleme problemi için literatürde yer alan öncelik kuralları ise şu şekilde sıralanabilir (Aydın, 2009):

1. Erken varış zamanlı
2. İlk gelene ilk hizmet
3. Kısa varış zamanlı
4. Uzun varış zamanlı
5. Rassal öncelik kuralı
6. Dinamik değişkenli
7. Toplam varış zamanlı
8. Kalan varış zamanlı
9. Az uçuş süresine göre öncelik kuralı
10. Çok uçuş süresine göre öncelik kuralıdır.

Bu çalışmadaki uçak çizelgeleme probleminin amaç fonksiyonu, iniş ve kalkış sırasında hedeflenen sürelerden toplam sapma maliyetinin en aza indirilmesi olarak ele alınmıştır. Çalışmanın kapsamı genel olarak uçak sıralama ve dinamik durum için uçak çizelgelemeden oluşmaktadır. Ancak dinamik durumdan bahsedilirken statik durumla farkını ortaya koymak için kullanılan veriler statik durum için de C programında çalıştırılarak iki durum için kıyaslama imkanı elde edilmiştir. Çalışmada, uçak çizelgeleme problemini çözmek için, metaasezgisel algoritmalarla birlikte, özgün 2 seviyeli optimizasyon yöntemi kullanılmıştır. Literatürde de kullanılan 4 metasezgisel algoritma (Tabu Arama, Genetik Algoritma, Tavlama Benzetimi, Karınca Kolonisi Algoritması) özgün optimizasyon yöntemleri kullanıldıktan sonra internette

bulunan OR kütüphanesindeki test problemlerine uygulanmış ve çözüm sonuçları Beasley'in (Beasley ve diğerleri, 2001) sonuçlarıyla kıyaslanmıştır.

Çalışmada kullanılan optimize edilmiş algoritma sonuçları; sadece statik değil aynı zamanda dinamik durum için olmasına ragmen OR kütüphanesindeki küçük problemlerde Beasley (Beasley ve diğerleri, 2001), Awasthi (Awasthi, ve diğerleri, 2013), statik durum için gelen uçakların çizelgelenmesinde elde ettiği sonuçlardan genel olarak daha iyi sonuç vermiştir.

Optimize edilmiş algoritmaların üçü de (Tabu Arama, Genetik Algoritma, Tavlama Benzetimi); optimize edilmiş Karınca Kolonisi Algoritması hariç, optimum değeri bulmada etkili algoritmalar olmuştur. Parametere ayarına devam edilerek Karınca Kolonisi Algoritması'nın da performansı arttırılabilir. Özellikle dinamik durum için büyük test problemlerinde Beasley (2001), Pinol (2004) ve Awasthi ve diğerleri (2013) tarafından bahsedilen bilinen en iyi çözüm değerine ulaşmak ya da yaklaşmak için parametre ayarına devam edilerek çözüm değeri iyileştirelebilir.

Tezin içeriği olarak; ilk bölümde, uçak çizelgelemi problemi ve tezim amacı, kapsamı ve tezin katkılarından bahsedilmiştir. ikinci bölümde optimizasyon kavramı, uçak çizelgeleme problemi için çözüm yaklaşımları, tezde kullanılan metasezgisel yöntemler ve karşılaştırılması, literatürde yer alan çalışmalar ve araştırmalar özetlenmiştir. Üçüncü bölümde ise; geliştirilen matematiksel model, algoritmaya göre hazırlanan akış şemaları ve pseudo kod ile optimize edilmiş metasezgisel algoritmaların 10 uçak için çalışma şekli algoritmalarda kullanılan zaman atama ve sıralama fonksiyonlarıyla birlikte detaylı şekilde anlatılmıştır. Dördüncü bölüm bulgularda ise, C programında kodlanmış algoritmaların sonuçlarından bahsedilerek detaylı karşılaştırma tablolarıyla irdelenmiştir. Son bölüm olan tartışma ve sonuç bölümünde ise çalışmanının katkılarından ve gelecekteki çalışmalar için önerilerden bahsedilmiştir.

2. GENEL KISIMLAR

Bu bölümde teze temel teşkil eden konulardan bahsedilerek ve öncelikle optimizasyon kavramı ve sınıflandırılmasından bahsedilmiştir. Daha sonra tez çalışmasında kullanılan yöntemlerle ilgili olarak genelden özele bir sıralama yapılarak sezgisel, metasezgisel yöntemler ve tezde kullanılan metasezgisel yöntemlerden bahsedilmiştir. Son olarak da literatürde yer alan çalışmalar ve literatürdeki çalışmaları daha iyi inceleyip kıyaslayabilmek için literatür özet tablosuna yer verilmiştir.

2.1. OPTİMİZASYON

Optimizasyon, ya da en iyileme, belirli bir problemin belirli koşullar altında birden fazla çözümünün var olduğu durumlarda, oluşan alternatif çözümler içerisinde en iyi çözümün bulunması işlemi olarak bilinmektedir. Diğer bir deyişle, problem çözümlerinde belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problemi, en uygun biçime getirme yöntemi optimizasyon olarak adlandırılabilmektedir (Murty, 2003).

Kısaca optimizasyon, bir sistemin olası tasarımları arasında en iyisini bulmak demektir.(Akay, 2009)

Optimizasyon sürecinde en iyi ve en kötü bulunmaya çalışılır. Problem için en uygun çözüm kümesini bulmayı amaçlamaktadır. Ayrıca optimizasyon teorisi, en uygun değerin hesaplanmasında kullanılabilecek teknikler, metotlar, prosedürler ve algoritmalar bütünüdür (Antoniou ve Lu, 2007).

Optimizasyon modelinde maksimum veya minimum yapılmaya çalışılan fonksiyona amaç fonksiyonu denir. Karar değişkeni, sistem performansını etkileyen kontrol altında tutulması gereken değişkenlerdir. Kısıtlar, karar değişkenlerinin alabileceği aralığı belirleyen parametrelerdir (Winston, 2004).

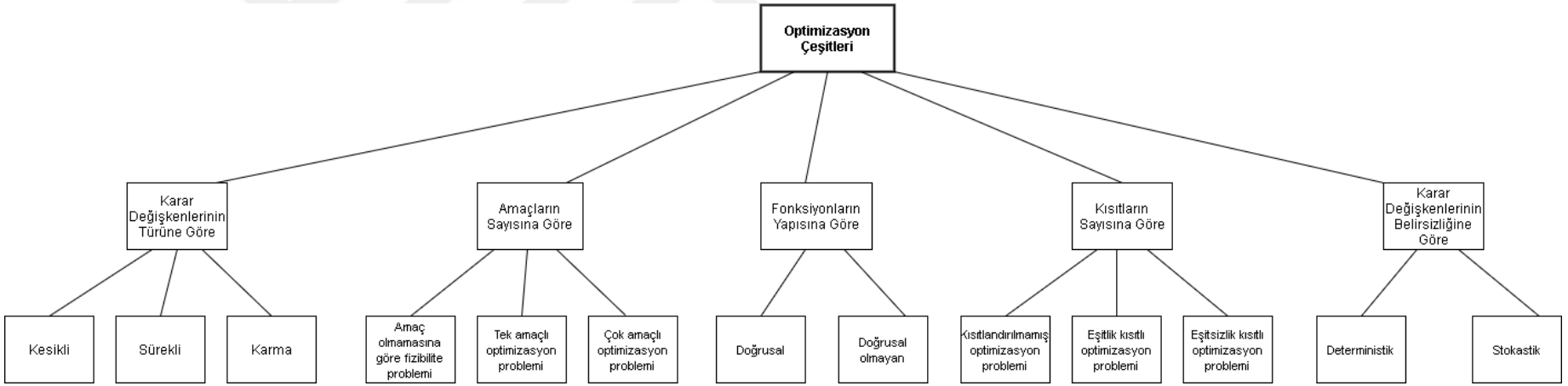
Optimizasyon, matematiksel modeller aracılığıyla amaç ve amaç fonksiyonlarına uygun karar üretmeyi sağlar. Bunun sağlanması yönünde de ilk adım problemin tüm detaylarını içerisinde bulunduracak modelleme işlemenin gerçekleştirilmesidir. (Çiftçioğlu, Doğan, 2017)

Optimizasyon sürecinde optimizasyon probleminin matematiksel modelini oluşturmak en önemli adımdır. Bu modeli doğru oluşturabilmek tasarım değişkenlerinin, kısıtların ve amaç fonksiyonunun doğru formüle edilmesine bağlıdır. (Çiftçioğlu, Doğan, 2017)

Optimizasyon modelinde maksimum veya minimum yapılmaya çalışılan fonksiyona amaç fonksiyonu denir. Karar değişkeni, sistem performansını etkileyen kontrol altında tutulması gereken değişkenlerdir. Kısıtlar, karar değişkenlerinin alabileceği aralığı belirleyen parametrelerdir (Winston, 2004).

2.1.1. Optimizasyon Modellerinin Sınıflandırılması

Optimizasyon modelleri amaç fonksiyonu, karar değişkenleri ve kısıtların özelliklerine göre sınıflandırılmaktadır (Çetin, 2007) (Mirjalili, 2016)

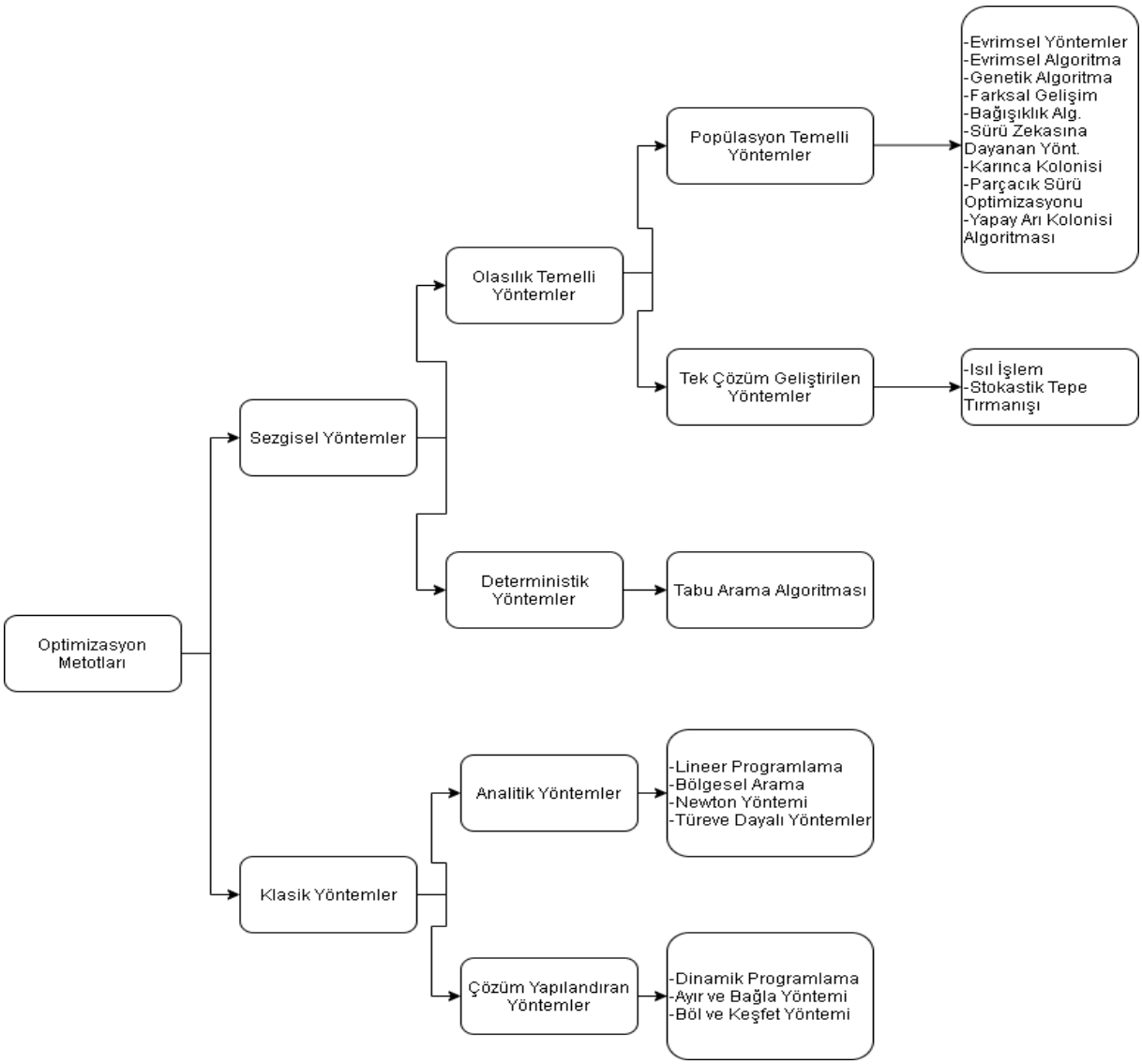


Şekil 2.1: Optimizasyon problemlerinin sınıflandırılması(Büyüksaatçi, 2015)

2.1.2. Optimizasyon Yöntemleri

Optimizasyon yöntemlerinde temel olarak belirlenmesi gereken hususlar; üzerinde çalışılan problemin veya sistemin temel bileşenlerinin oluşturulması, probleme veya sisteme ön kestirim parametlerinin atanması, hedeflenen amaç fonksiyonu belirlenmesi, hedef dışına çıkılmaması için gereken sınırlandırma kısıtlamalarının belirlenmesi ve son olarak probleme veya sisteme adapte edilecek en uygun optimizasyon yöntemin uygulanması olarak nitelendirilebilir (Karaboğa, 2014).

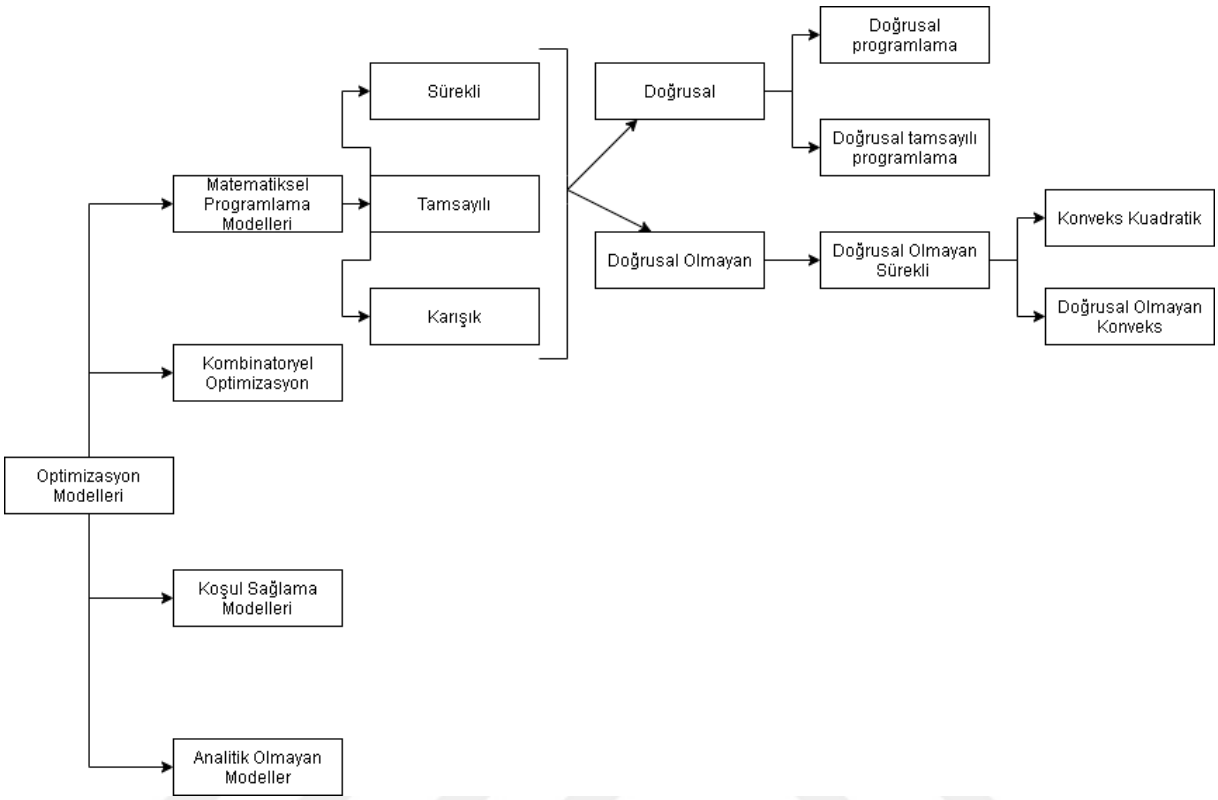
Optimizasyon yöntemleri, genel olarak kesin çözüm üreten klasik teknikler ve yaklaşık optimal çözüm üreten modern sezgisel teknikler olmak üzere 2 başlık altında toplanır. (Dumitrescu ve diğerleri, 2003)



Şekil 2.2: Optimizasyon metotlarının sınıflandırılması(Kıran, 2014)

2.1.2.1. Klasik Optimizasyon Modelleri

Sürekli ve türevlenebilen bir fonksiyonun diferansiyel matematik kurallarına göre en fazla ya da en az alabileceği değerinin incelendiği problemlere klasik optimizasyon problemi denir (Kahraman, 2010)

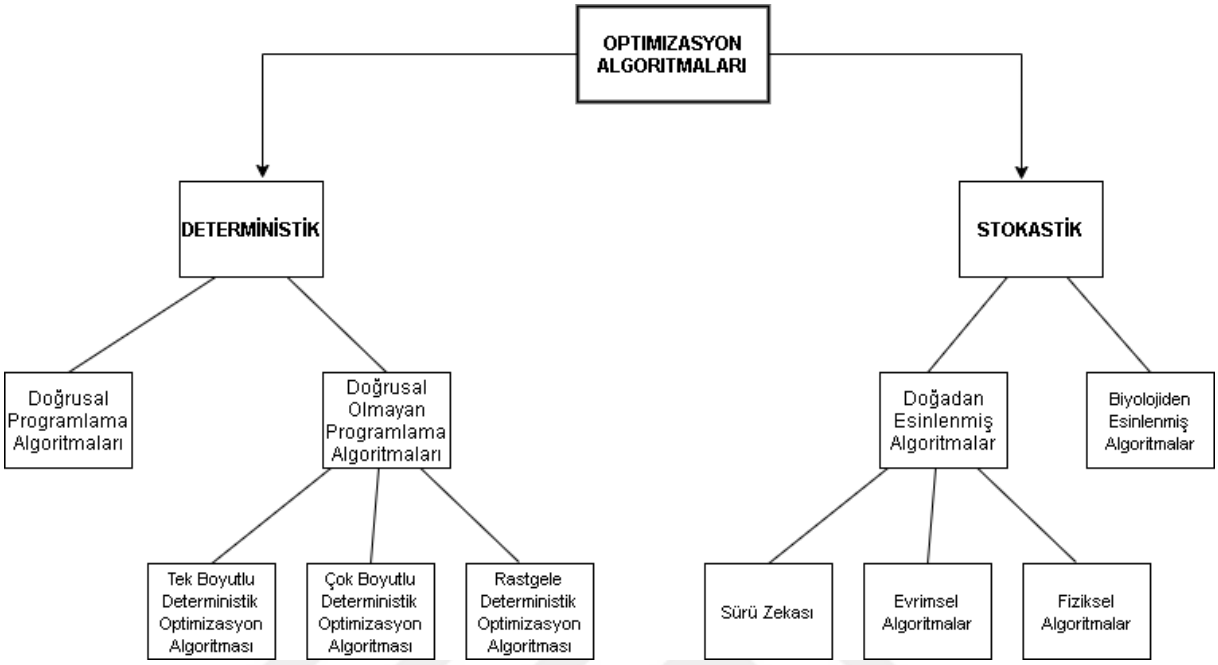


Şekil 2.3: Klasik Optimizasyon Modelleri (Talbi, 2009).

2.1.2.2. Optimizasyon Algoritmaları

Optimizasyon problemleri günlük yaşantıda sürekli olarak karşılaşılan problemlerdir. Bu problemler çözüm uzayının çok geniş olduğu, çözülmesi oldukça zor olan problemlerdir. Problemlerin çözümü için çeşitli optimizasyon algoritmaları geliştirilmiştir. Optimizasyon algoritmaları günlük yaşamdaki birçok problemi çözmeye yarayan ve genellikle büyük bir çözüm uzayına sahip problemlerde optimal çözümü tespit etmeye yarayan yaklaşımlar bütünüdür (Kaleli, 2019)

Optimizasyon yöntemleri ya da algoritmalar, çözüm sonuçlarına göre lokal (yerel) optimum çözüm ve global (küresel) optimum çözüme ulaşabilirler . Optimizasyon algoritmalarının genel amacı bir amaç fonksiyonun tanımlanmış arama veya çözüm bölgesinde daha geniş bir tarama yaparak tüm olabilecek çözümleri lokal çözüme bağlı kalmadan global minimuma veya maksimuma ulaşabilirliğini artırmaktır. Lokal optimimum çözüme ulaşan algoritmalarda ise, bir başlangıç çözümü ele alınarak global optimizasyona göre daha dar bir arama alanında minimum veya maksimum optimum çözüme ulaşılmak hedeflenir.



Şekil 2.4: Optimizasyon algoritmalarının sınıflandırılması (Büyüksaatçi, 2015).

2.1.2.3. Matematiksel model

Modelleme, problemlerin matematiksel olarak ifade edilmesine denir. Model kavramı, matematiksel düşünceleri ve süreçleri kavramsallaştırmayı göstermek için mühendislik ve temel bilimler gibi alanlarda sıkça kullanıllmaktadır. Matematiksel model, verilen bir durum veya problemle ilgili iki veya daha fazla değişken arasındaki ilişkinin matematiksel bir ifadesidir (Saka, 2016). Matematiksel model, doğrusal ya da doğrusal olmayan, dinamik ya da statik, açık ya da kapalı, ayrık ya da sürekli, deterministik ya da stokastik, tümden gelim, tüme varım ya da kaydırmalı şeklinde sınıflandırılabilmektedir (Gertsev ve diğerleri, 2004).

2.1.2.4. Matematiksel Modelin Bileşenleri

Bir optimizasyon problemini matematiksel modelleme süreci; karar değişkenlerinin belirlenmesi, optimizasyonun hedef ve kısıtlarının belirlenmesi, modelin formüle edilmesi ve son olarak da en uygun modelin seçimi ve geliştrilmesi olmak üzere 4 adımdan oluşmaktadır.Bu kavramlar çerçevesinde bir matematiksel modelin bileşenleri aşağıdaki gibidir:

Amaç fonksiyonu

Maksimum ya da minimum yapılmak istenen fonksiyon olarak tanımlanır. Örneğin, bir imalat ya da üretim işleminde kârı maksimum yada maliyet minimum yapılmak istenebilir. Deneysel verilerin bilinen bir modele uydurulması probleminde gözlenen veri ile tahmin edilen değer

arasındaki sapma minimum yapılmak istenebilir. Bu durumda modelin bilinmeyen parametreleri tahmin edilir (Panos, Douglas, 2000).

Karar değişkenleri

Amaç fonksiyonun değerini etkileyen ve amaca ulaşmak için control edilen değerlerdir. Değişkenlerin alacağı değerler, amaç fonksiyonunun değerinin ne olacağına belirler

Kısıtlar

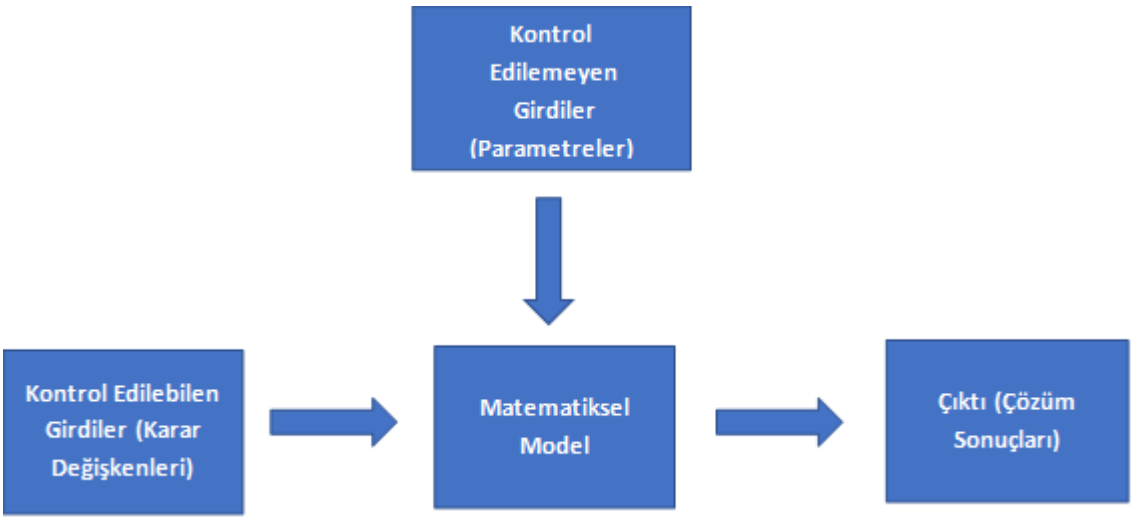
Karardeğişkenlerinin, belirli değerleri almasına ve belirli değerleri de almamasına yarayan faktörler yani bu değerler üzerindeki sınırlama ya da gereksinimlerdir. Amaç fonksiyonu gibi karar değişkenlerinin içerildiği matematiksel fonksiyonlar olarak ifade edilmektedirler.

Parametreler

Matematiksel modele etkisi olan ve aynı zamanda karar vericinin kontrol edemediği faktörlerdir.

Varsayımlar

Matematiksel modelin oluşturulması sırasında doğru oldukları varsayılan modelin bileşenleridir.



Şekil 2.5: Matematiksel Modelleme Süreci

Optimizasyon sürecinde optimizasyon probleminin matematiksel modelini oluşturmak en önemli adımdır. Bu modeli doğru oluşturabilmek tasarım değişkenlerinin, sınırlayıcıların ve amaç fonksiyonunun doğru formüle edilmesine bağlıdır.

Tipik bir optimizasyon probleminin matematiksel modeli aşağıdaki gibi gösterilebilir.

Amaç fonksiyonu,

$z = f\ (x)$ (2.1.1)

Kısıtlar,

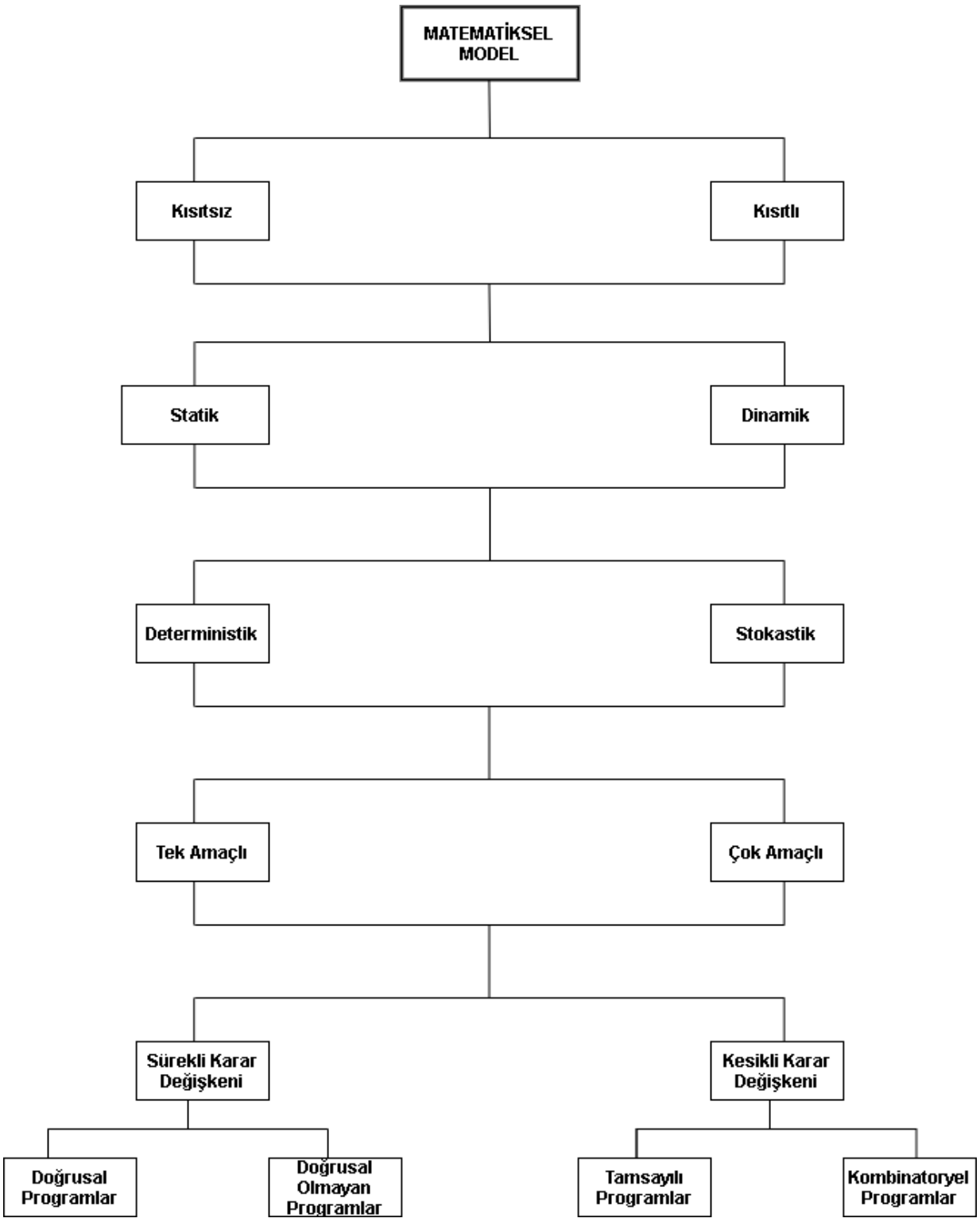
$h_j\ (x) = 0, j = 1,2, \dots ,0$ (2.1.2)

$g_k\ (x) \leq 0, k = p + 1, \dots , m$ (2.1.3)

$x_i \in X, X = (x_1 , x_2 , \dots , x_q)$ (2.1.4)

h_j eşitlik kısıtlarını ve g_k eşitsizlik kısıtlarını ifade etmektedir. X tasarım değişkenlerini, q değişkenlerin toplam sayısını ifade eder. Sınırlayıcı sayısı ise m ile ifade edilmiştir (Çiftçioğlu, Doğan, 2017).

Aşağıdaki şekilde problemin yapısına göre matematiksel model türleri gösterilmiştir.



Şekil 2.6: Problemin türüne göre matematiksel model çeşitleri (Eröz, 2020)

2.2. UÇAK ÇİZELGELEME PROBLEMLERİ İÇİN ÇÖZÜM YAKLAŞIMLARI

Optimizasyon metotları temel olarak iki kategori altında toplanabilir. Bu gruplar, kesin çözüm üreten klasik teknikler ile yaklaşık optimal çözüm üreten modern sezgisel tekniklerdir. Küçük boyutlu NP- zor olmayan problemlerin çözümünde tam sayılı programlama ve algoritmalar kullanılır. Büyük boyutlu problemlerin çözümünde ise daha hızlı ve mantıklı olduğu için herzaman kesin çözümü vermese de sezgisel algoritmalar kullanılmaktadır.

2.2.1. Kesin Çözüm Yöntemleri

Kesin çözüm yöntemleri, uygun bir süre içerisinde optimum bir çözüm bulmayı amaçlar ancak problem büyüklüğü arttıkça en uygun çözümü bulmak zorlaşır.

Dal-sınır yöntemi, kesme düzlemi yöntemi, dinamik programlama yaklaşımı, tam sayılı programlama gibi yöntemler literatürde sıklıkla karşılaşılan kesin çözüm yöntemleridir.

2.2.2. Yaklaşık Çözüm Yöntemleri

2.2.2.1.Sezgisel Yöntemler

Sezgisel yöntemler, çözülmek istenen problem boyutunun elverişli optimal çözüm algoritmaları ve hesaplama araçları limitleri üzerinde olduğunda veya optimal çözüme ulaşma maliyetine katlanılmak istenmediğinde kullanılmaktadır. İyi tasarlanmış ve modellenmiş olan sezgisel yöntemler ile optimum veya optimuma yakın kaliteli sonuçlar elde edilebilmektedir. (Beasley, Sonander, Havelock, 2001)

Sezgisel yöntemler üç ayrı başlık halinde incelenmektedir. Birinci grup sezgisel yöntemler, öncelik kuralları kullanarak çizelge oluşturan yöntemlerdir. Bir defa uygulanarak ulaşılan çözüm nihai çözüm olarak kabul edilir. İkinci grupta ise, öncelik kuralları kullanarak veya örnekleme yolu ile birden çok uygulama yapılır ve birden çok çözüm elde edilir. En iyi çözüm nihai çözüm olarak kabul edilir. Meta-sezgisel yöntemlerin son senelerde öne çıkışı ve yaygın kullanımı nedeni ile bu sezgisel yöntemleri ayrı bir başlık altında üçüncü grup olarak incelenmektedir (Köksalan & Erkip, 2000).

Geleneksel olarak kullanılan optimizasyon algoritmaları genel olarak global minimum veya maksimum ulaşamayıp dar bir çözüm bölgesinde kalırlar. Sezgisel teknikler global optimum çözümü garanti edememelerine rağmen, makul bir hesaplama maliyeti ile optimum veya optimuma yakın bir çözümü elde edebilirler. Klasik tekniklerin dezavantajlarına karşılık sezgisel tekniklerin avantajları da aşağıda maddelenmiştir. (Kiran, 2014)

i) Bir probleme özgü değildirler ve birçok probleme kolaylıkla uyarlanabilirler.
ii) Türev vb. ağır matematiksel hesaplamaları gerektirmezler.
iii) Kâr veya maliyet amaç fonksiyonları ile çözümlerin kalitesini ayırdedebilirler.
iv) Problemin boyutuna bağlı olmaksızın makul sürelerde optimum veya optimuma yakın bir çözümü elde edebilirler.
v) Kesin matematiksel formüllerle yapılan tanımlamalar için gerçek dünya problemlerinde amaçlar, kısıtlar, problem verisinin toplanması gibi zorluklar mevcuttur. Model için giriş parametrelerinin belirlenmesinde kullanılan verinin hatalı olması, sezgisel yöntemle elde edilebilecek alt optimal çözümden daha fazla hataya neden olabilir.
vi) Hesaplama maliyetleri düşüktür
Sezgisel algoritmalar en iyi çözümü garanti edemezler bu tür algoritmalar yakınsama özelliğinden dolayı çözüm arama bölgesi içerisinde, en kısa sürede bulunan optimum çözümün civarındaki bir çözümü seçerler. Sezgisel algoritmalarda çözümün doğruluğunu kanıtlama yetkinliği güdülmez (Karaboğa, 2014).
Sezgisel algoritmaların başlıca kullanım amaçları arasında, araştırmacılar için karmaşık yapıdaki problemleri basitleştirip anlaşılabilirliğini artırmak, optimizasyon problemlerinde kesin özümün tanımlanamadığı bir yapıda veya sistemde olması, kesin çözümü bulmayı amaçlayan işlemlerde bir ön öğreniminin gerçekleşmesi olarak sayılabilir (Karaboğa, 2014).

2.2.2.2.Metasezgisel Yöntemler

Metasezgisel algoritmalar, klasik sezgisel algoritmaların doğadan esinlenerek geliştirilmiş hali olarak görülebilir (Laporte ve ark., 2000). Optimizasyon problemlerinin çözümü için kullanılan metasezgisel yöntemler aynı zamanda zeki yöntemler veya yapay zeka optimizasyon yöntemleri olarakta bilinmektedir ve literatürlerde her üç başlık kullanılmaktadır. (Önal, 2020) Ayrıca, yöneylem araştırmaları, yapay zeka ve doğadan esinlenilerek geliştirilmektedir.

Metasezgisel yöntemler kısa zamanda optimum veya optimuma yakın sonuçlar elde edebilmek amacıyla kullanılan yöntemlerdir. Yani optimum çözümün elde edilmesini garanti etmeyen yöntemlerdir. Ancak geliştirilen algoritmanın her çalıştırılmasında optimuma veya optimuma yakın sonuçlara ulaşma olasılığının yüksek olduğu durumlarda geliştirilmiş olan algoritmanın güçlü ve uygulama alanında kullanılabilir olduğu söylenebilir. Metasezgisellerin en önemli sınıflandırması kısaca aşağıdaki gibidir (Blum ve Roli, 2003):

Doğadan esinlenen veya doğadan esinlenmeyen algoritmalar

Popülasyon tabanlı veya tek bir nokta arama

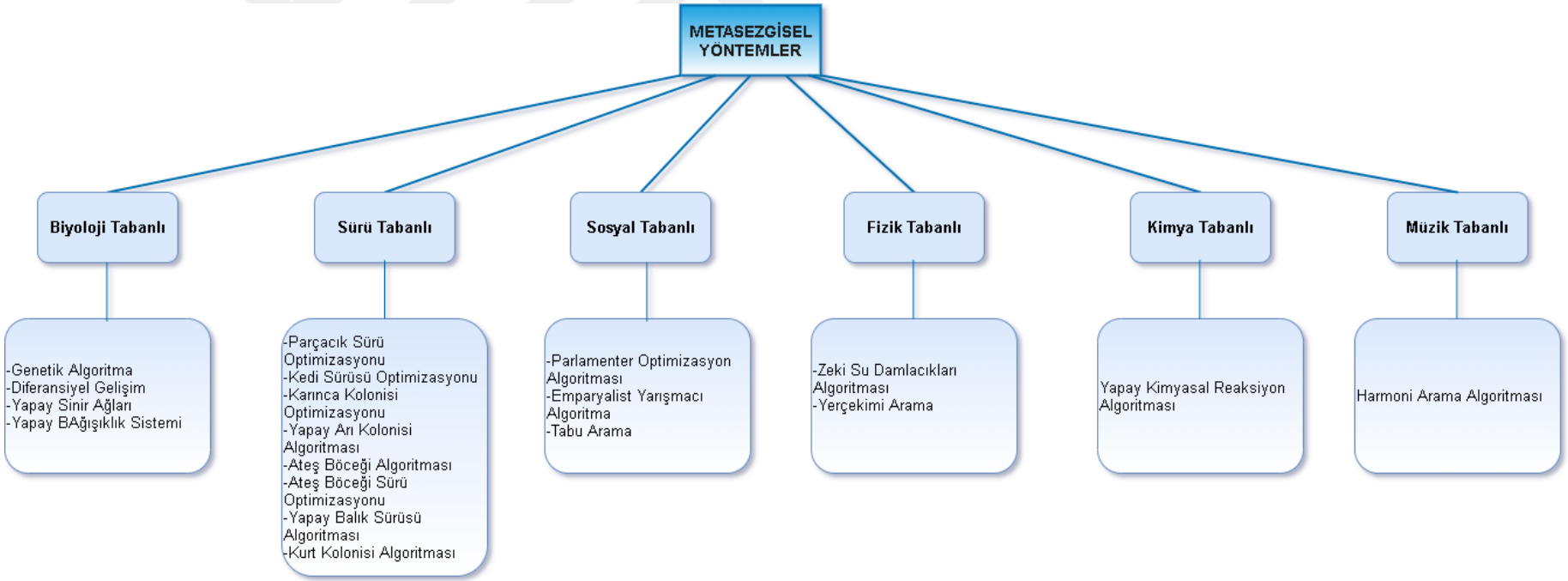
-Populasyon Tabanlı Metasezgisel Yöntemler (Genetik Algoritma, Karınca Kolonisi Algoritması, Taklitçi Algoritma, Dağınık Arama vs..)

-Tek Bir Nokta Arama Yöntemleri (Benzetim Tavlaması, Tabu Arama vs..)

-Dinamik veya statik amaç fonksiyonu

Bir veya çeşitli komşuluk yapıları

Bellek kullanan veya kullanmayan yöntemler



Şekil 2.7: Metasezgisel Yöntemler (Önal, 2020).

2.3. TEZDE KULLANILAN METASEZGİSEL ALGORİTMALAR

2.3.1. Genetik Algoritmalar

İlk olarak 1975 yılında Holland tarafından tanıtılmıştır. Genetik Algoritma (GA), iyi bir çözüm bulmak için doğal seleksiyon ve evrimsel teoriye dayanan populasyon temelli bir metasezgisel algoritmadır. Bu prensiplerden esinlenerek, düşük kaliteli çözümler popülasyondan elimine edilir ve hasta bireyler başarılı yeni bireyleri (çocuk birey) garanti etmek için yeniden üretirler. Yeni birey ebeveynlerinin iyi genlerine sahip olabildiği gibi kötü genlerine de sahiptir. Ortama uyum sağlamayan kötü yeni bireyler hayatlarını devam ettiremeyeceklerdir. (Park, Choi ve Kim, 2003).

GA, yerel minimuma erken yakınsamayı engellemek için; olası tek bir çözümü ele alan tavlama benzetimi, tabu arama gibi diğer yerel arama yöntemlerinin aksine arama esnasında çözüm popülasyonu kullanır (Park, Choi ve Kim, 2003). Ayrıca populasyondaki her birey kromozom olarak adlandırılır ve kromozomlar olurlu bir çözümü temsil eder. Yeni bireyler üretmek için çaprazlama ve mutasyon operatörleri kullanılır. Üretilen pek çok yeni nesilden sonra popülasyon yerel veya global optimum olabilecek bir çözüme yakınsar (Gen ve Cheng, 1975).

GA geleneksel yöntemler ile çözümü zor veya imkansız olan, çözüm uzayının çok büyük olduğu problemlerde kullanılmaktadır. Çözüm uzayının taranması geleneksel metotlarla çok uzun sürebilmektedir. GA ile kısa bir sürede optimuma yakın ve kabul edilebilir sonuçlar bulunabilmektedir. (Aydın, 2009)

Genetik algoritmanın diğer algoritmalara göre; karmaşık problemleri çözme yeteneği ve parallelik gibi iki avantajı bulunmaktadır. Amaç fonksiyonu sürekli veya süreksiz, sabit veya geçici, doğrusal veya doğrusal olmasına bakılmaksızın, genetik algoritmalar tarafından kolaylıkla ele alınabilmektedir. Birden fazla gen yapısı paralel uygulamalar için de uygun olabilmektedir (Koziel ve Yang, 2011).

Genetik algoritmayı diğer arama yöntemlerinden farklı kılan özellikler aşağıda sunulmuştur:

· Genetik algoritma, parametre setlerinin kodları ile ilgilenir, parametrelerin kendileri ile doğrudan ilgilenmez.

· Genetik algoritmanın arama alanı, yığının veya popülasyonun tamamıdır. Tek nokta veya noktalarda (çözüm bölgesinin daraltılmış bölgelerinde) arama yapmaz.
· Genetik algoritmalarda, amaç fonksiyonu kullanılır, sapma değerleri veya diğer hata faktörleri kullanılmaz.
· Genetik algoritmaların uygulanmasında kullanılan operatörler, stokastik yöntemlere dayanır, deterministik yöntemler kullanılmaz.

GA' nın diğer metasezgisellere kıyasla bir takım avantajlı özellikleri vardır (Dao, Abhary ve Marian, 2017):

- Kısıtları ve kalite ölçütlerini tanımlama esnekliği,
- Sürekli ve kesikli değişkenlerle çalışma yeteneği,
- Geniş arama uzayı kabiliyeti,
- Çoklu optimum / iyi çözümler sunma becerisi ve
- İşlem süresini kısaltmak için paralel hesaplama tekniklerini kullanması gibi.

GA, optimum olmadığı zamanlarda da optimum yakın çözümleri garanti eder. Problemlerin GA ile çözümünde uygulanan işlem adımları aşağıdaki gibidir (Gen, 1996):

1. Arama uzayındaki bütün olası çözümler, dizi olarak kodlanır. Bu diziyi oluşturan her bir elemana gen denir. Birden fazla genin bir araya gelerek oluşturduğu diziye de kromozom adı verilir. Her bir dizi arama uzayında belirli bir bölgeye tekabül eder.
2. Genellikle rassal bir çözüm seti seçilir ve başlangıç popülasyonu olarak kabul edilir. Popülasyon kromozomlardan oluşan topluluktur.
3. Her bir dizi için bir uygunluk değeri hesaplanır, bulunan uygunluk değerleri dizilerin çözüm kalitesini gösterir.
4. Bir grup kromozom belirli bir olasılık değerine göre rassal olarak seçilip üreme işlemi gerçekleştirilir.
5. Üreme işleminde, çeşitli genetik operatörler kullanılabilir.
6. Yeni bireylerin uygunluk değerleri hesaplanarak çaprazlama ve mutasyon işlemlerine tabi tutulur.
7. İterasyon nesil sayısına ulaşınca işlem bitirilir. Uygunluk değeri en yüksek olan dizi seçilir.

Genetik Algoritmanın işlem adımlarına göre oluşturulmuş sözde (pseudo) kodu aşağıdaki gibidir:

Amaç fonksiyonunu yaz f(x), $x= (x_{1, \ldots\ldots} x_{2)}^{T}$
Sonuçları kromozomların içerisine kodlayarak yaz (0-1dizi şeklinde)
Uygunluk F değerini belirle (örn., F ve F(x) maksimizasyon için)
ilk popülasyonu oluştur
Başlangıç çaprazlama olasılığı (p_c) ve mutasyon olsılığı (P_m) içindeğerleri ata
while t < en çok üretim değeri
Çaprazlama ve mutasyon yöntemlerini kullanarak yeni sonuçlar oluştur
if p_c < rand ise çaprazlama işlemini yap; **end if**
if p_m < rand ise mutasyon gerçekleştir; **end if**
Uygunluk değeri artarsa eğer, yeni çözümleri kabul et
Yeni nesil için bulunan en iyi değeri seç
end while
Sonuçları ve görselleştirmeyi yorumla

Şekil 2.8: Genetik algoritma sözde kodu (Yang, 2010).

2.3.2. Tavlama Benzetimi (Simulated Annealing)

Kirkpatrick, Gelatt ve Vecchi (1983) ve Cerny (1985) tarafından birbirinden bağımsız şekilde ortaya konmuştur. Tavlama benzetimi, kombinatoryal optimizasyon problemlerinde global optimum veya yakın optimumu bulmak için sık kullanılan, eski metasezgisel yöntemlerden biridir. Tavlama benzetimi ismi, katıların fiziksel tavlanma süreci ile olan benzerlikten ileri gelmektedir (Johnson ve diğ., 1989). Tavlama benzetimi, metal gibi fiziksel bir maddenin, ısıtılarak yüksek enerji düzeyine çıkarılıp sonradan yavaş şekilde soğutulması sürecini ifade eder. Sürecin amacı ise daha düşük bir enerji seviyesine ulaşmaktır.

TB algoritması bir başlangıç çözümle ve yerel minimumlardan kaçınmak için yüksek bir sıcaklık (T) değeri ile başlamaktadır. Her bir hesaplama adımında algoritma, bir önceki çözümün komşuları arasından çok sayıda çözümler üretmekte ve sıcaklığı önceden belirlenen

bir kurala göre azaltmaktadır. Üretilen komşu çözümler arasından rastgele bir tanesi seçilmektedir. Bu seçilen çözüm ya yeni çözüm olarak kabul edilmekte ya da reddedilmektedir. İşlemler yeni çözümle (eğer kabul edilmişse) veya bir önceki çözümle (eğer kabul edilmemişse) tekrar başlamaktadır. Algoritmayı durdurmak için pek çok farklı kriter (iterasyon sayısı, sıcaklığın minimum değeri vb.) kullanılabilir. (Büyüksaatçi, 2015)

Tavlama benzetimi algoritması dört temel bileşene ihtiyaç duyar:

1. Konfigürasyon: İyi bir cevap için yapılan araştırmada, problem çözme olasılığı.
2. Hareket kümesi: Lokal minimumdan kaçarak bütün uygun konfigürasyonlara erişmeye izin verecek hareketler kümesi.
3. Maliyet fonksiyonu: Amaç fonksiyonuna bağlı olarak azaltılacak miktar.
4. Soğutma çizelgesi: Ne zaman ve ne kadar azaltılacağını, ne zaman soğutmanın sona erdirileceğini belirlemek için bir başlangıç maliyet fonksiyonu ve kuralları. (Aydın, 2009)

Tavlama benzetimi algoritması çalışma prensibi olarak, lokal arama metoduna benzer. Lokal arama metodunun en büyük dezavantatı, global optimum yerine local optimumu bulmasıdır. Tavlama benzetimi, bu olumsuzluğu ortadan kaldırır. Kontrol parametrelerine bağlı olarak, komşulık değerlerindeki maliyet fonksiyonu değişimini kabul eder. Teorik olarak tavlama benzetiminin çalışma prensibi, markov zincirlerinin modellenmesi ile elde edilir. Algoritmanın adımları ise (Aydın, 2009) ;

1. Rassal olarak ya da seçilmiş bir çözüm kümesini, başlangıç çözümü olarak kabul et.
2. Başlangıç çözümüne eniyi çözüm olarak bir değere ata.
3. Başlangıç çözüm değerinin maliyet fonksiyonunu hesapla.
4. Başlangıç sıcaklığı belirle.
5. Başlangıç sıcaklığı bir değere ata.
6. Durma kriterleri sağlanmadı ise aşağıdaki işlemleri gerçekleştir.
    a. Markov zinciri uzunluğunu belirle.
    b. Başlangıç sıcaklığını azalt ve altıncı maddeye geri dön.

7. Eniyi çözüm çizelgesini oluştur ve dur.

Tavlama Benzetimi Algoritması sözde (pseudo) kodu ise şu şekildedir:

Amaç fonksiyonunu yaz f(x), $x= (x_{1, \ldots..} x_n)^T$

Başlangıç sıcaklığını (T_0) ve başlangıç tahmini değerini (x^0) ata

Sıcaklığın son değerini (T_f) ve en fazla iterasyon sayısını N'i ayarla

Soğutma işlemini belirle $T \longrightarrow \alpha T$, $(0 < \alpha < 1)$

while $T > T_f$, $n < N$

Yeni bölgelere rastgele git : $x_{n+1} = x_n$ + randn

$\Delta f = f_{n+1} (x_{n+1}) - f_n(x_n)$'i hesapla

Bulunan sonuç daha iyiyse onu kullan

if daha iyi değilse

Bir r rassal sayı oluştur

$p = \exp [-\Delta f / T] > r$ ise eğer, kabul et

end if

En iyi x_x ve f_x değerlerini güncelle

$n = n+1$

end while

Şekil 2.9: Tavlama benzetimi algoritması sözde kodu (Yang, 2010).

2.3.3. Tabu Arama

Tabu Arama Algoritması, Tavlama Benzetimi'ne benzer şekilde, daha önce gözlemlenen çözüm noktaları arasında döngü yaratan hareketleri cezalandırarak yerel bir minimumdan kaçınmaya çalışan yerel arama iyileştirme yöntemidir. Algoritma içindeki bu yasak hareketlere "tabu" denir. Tabu algoritması, belirli sayıdaki tekrarlama için bu hareketlerin listesini tutar. Algoritmadaki döngü, tabu listesinin uzunluğuna bağlıdır. Tabu listesinin uzunluğu küçükse, işlemin döngüye girme olasılığı yüksek olabilir (Lee vd, 1997).

Tabu arama, hedefe kısıt ve tabu kullanarak belirli bir olasılıkla ulaşır. Hedefe ulaşma olasılığının yükselmesi için kısıt ve tabu sayısını yani yasak sayısını artırmak gerekir. Algoritmanın adımları ise şöyle sıralanabilir (Aydın, 2009):

1. Başlangıç çözümünü rassal olarak seç ve bu çözüm için amaç fonksiyonu değerini hesapla.
2. Değişim operatörünü seç. Bu operatöre göre pozisyonlar arasında değişimi gerçekleştir.
3. Yeni değişimin amaç fonksiyonu değerini hesapla. Başlangıç değeri ile veya bir önceki değer ile karşılaştır. Değişimin yasak olup olmadığını belirle, eğer değişim kabul edilebilir ise yasak değildir.
4. İterasyon sayısı kadar ikinci maddeye giderek işlemleri tekrarla. Hesaplanan eniyi çözümü belirle.

Tabu Arama Algoritmasının performansının, yerel bir optimumda sıkışmaya meyilli olduğu için zor problemlerin üstesinden gelme konusunda geliştirilmesi gerekmektedir. Yerel optimumdan kaçabilmek için, yoğunlaştırma ve çeşitlendirmeye yönelik ek bileşenler çalışmalara dahil edilmelidir.

Tabu Arama Algoritmasıyla elde edilen çözüm alanınında, yeni alanlara yönelmek için çeşitlendirme kabiliyeti olması ve en umut verici alanları yoğun bir şekilde keşfetmek için yoğunlaştırma yeteneği olması önemlidir. Ancak bu iki uygulama arasında bir denge bulunması da çok önemlidir.

Yoğunlaşma ve çeşitlendirme mekanizmalarının dengelenmesi, genellikle, sabit uzunlukta tabu listeleri kullanıldığında veya tabu kullanımının kontrol edilmesiyle tabu listesinin uzunluğunun kontrol edilmesiyle yapılır.

Tabu Arama Algoritması en genel sözde (pseudo) kodu şu şekildedir :

$s - s_0$; başlangıç çözümü

Tabu listesini başlat, orta vade ve uzun vade hafızaları tanımla;

Repeat

Çözümlerin "A" kümesini oluştur;

A kümesinin en iyi s' komşusunu bul; tabu olmayan %'leri ya da aspirasyon kriteri tutulur

$s - s'$;

Tabu listesini ve aspirasyon durumlarını güncelle

if $f(s) < f(s_0)$ **Then** $s_0 - s$;

Until Durdurma kriteri sağlanana kadar devam et

Output: s0 bulunan en iyi çözüm.

Şekil 2.10: Tabu arama algoritması sözde kodu (Alharkan ve diğerleri, 2019)

2.3.4. Karınca Kolonisi Algoritması

Genetik Algoritma'da olduğu gibi karınca kolonisi algortiması da popülasyona dayalı bir metasezgisel algoritmadır. Yiyecek ararken karıncalar bulundukları ortamla iletişime geçerler ve feromon salgılayarak izlerini bırakırlar ve bu izler sayesinde diğer karıncalar tarafından takip edilirler. Tesadüfi ilk yiyecek aramadan sonra feromon yoğunluğu değişir ve karıncalar feremonun yüksek olduğu rotayı takip ederler. Kolonideki karıncaların yaklaşık olarak geçtikleri yola eşit seviyede feromon maddesi bıraktıkları kabul edilirse, feromon miktarı artan karınca sayısı ile çoğalmaktadır. Birçok karıncanın aynı rotayı takip etmesi sonucunda o rota tercih edilen yol haline gelmektedir. Dolayısıyla olumlu bir geribildirim mekanizması sonucunda bu yollar, genelde en kısa ya da daha etkili olarak ortaya çıkacaktır (Yang, 2010).

Dorigo ve Gambardella (1997) bu özelliği kombinatoryal optimizasyon problemlerini çözmede kullanmışlardır. Stokastik yerel bir arama politikası uygulayarak, her karınca bir çözüm üretebilir. Bir tur, kolonun tüm karıncaları farklı kalitede çözümler ürettiğinde sona erer. Her turun sonunda toplanan bilgi, küresel bir feromon güncelleme kuralı aracılığıyla güncellenir. Bir sonraki turdaki bilgileri kullanarak karıncaların daha iyi çözümler üretmesi beklenir. (Hançerlioğulları, 2013)

Karınca Kolonisi Algoritması sözde (pseudo) kodu şu şekildedir:

Amaç fonksiyonunu yaz f(x), $x= (x_{1, \ldots\ldots} x_n)^T$

[ya da Rotalama problemi için $f(x_{ij})$, $(i,j) \in \{1,2, \ldots, n\}$]

Feromon buharlaşma oranı γ'ye karar ver

while

for tüm n boyutu (veya düğümleri) üzerinde döngü

Yeni çözümler üret

Yeni çözümleri gözden geçir

Feromon $\delta\varphi_{ij}$ 'li daha iyi bölgeleri işaretle

Feromonu güncelle $\delta\varphi_{ij} \longleftarrow (1 - \gamma) \delta\varphi_{ij} + \delta\varphi_{ij}$

end for

Güncel en iyi değeri elde edinceye kadar algoritma işlemlerine devam et

end while

En iyi sonuçlar ve feromon dağılımını işle

Şekil 2.11: Karınca Kolonisi Algoritması sözde kodu (Yang, 2010).

2.3.5. Kullanılan Metasezgisel Yöntemlerin Karşılaştırılması

Literatüre göre, populasyon tabanlı ve tek nokta arama tabanlı olmak üzere iki tür metasezgisel algoritma vardır. Genetik algoritma ve Karınca Kolonisi gibi algoritmalar popülasyona dayalı algoritmalardır ve yeni çözümler üretmek için bir grup çözüm ile çalışır. Benzetim tavlaması ve Tabu arama algoritmalar ise tek nokta arama tabanlı algoritmalardır ve tek bir çözümle başlayıp komşuluk arama yoluyla sonucu iyileştirmeye çalışırlar.

Tek nokta arama tabanlı algoritmalar, bir başlangıç çözüm ile başlar ve tekrarlı olarak daha fazla çözüm üretir. Popülasyon tabanlı algoritmalarla karşılaştırıldığında Tavlama Benzetimi ve Tabu Arama yöntemini uygulamak için gereken süreler daha kısadır. Genel olarak, hem Tabu Arama hem de Benzetim Tavlaması Algoritmasının, makul bir süre içerisinde iyi kalitede sonuçlar üretebilmektedir. Ancak genellikle, Benzetim Tavlaması Algoritması daha kolay kabul edilir ve daha az bilgisayar belleğine ihtiyaç duyar. Genetik Algoritma ise her tekrarda birden fazla çözümü izlediği için çözüme ulaşması daha yavaş olmaktadır. Genetik Algoritma

içerisinde kullanılan komuşuluk arama yöntemleri nedeniyle tek bir çözüm yerine bir çözüm setini esas alır. Böylece Genetik Algoritma, aramayı komşuluk bölgerindeki en çok umut veren bölgelere yoğunlaştıramamaktadır.

Kim, Lim ve Park'ın çalışmasına göre; tavlama benzetimi algoritmasına, ekleme komşuluk arama algoritması eklenmesiyle elde edilen sonuçların diğer algoritmalardan daha iyi sonuç verdiği gösterilmiştir.

Parthasarathy ve Rajendran (1998) ilk çözümün belirli bir kuralla sağlandığı Tavlama benzetimini önermiştir. Sonuçlar, Tabu Arama da dahil olmak üzere diğer buluşsal yöntemlerle karşılaştırılmış ve Tavlama Benzetimi Algoritması en iyi kuralları üretmiştir. Vallada ve diğ. (2008) toplam ağırlıklı geçiciliği en aza indirgemek amacıyla atölye çizelgeleme problemi için sezgisel ve metasezgisel yaklaşımların kapsamlı bir incelemesini sağlamıştır.

Metasezgisel algoritmaların perfomansı karar verilecek konuya, problemin yapısı ve büyüklüğüne göre değiştiği için algoritma seçiminde bu kriterler dikkate alınıp seçilmelidir. Algoritma üzerinde parametre ayarı yapılması, problemin türü ve büyüklüğü, algoritma karşılaştırılması da yapılan çalışmaların sonucunu etkiler.

Tablo 2.1: Metasezgisellerin kullanım alanlarına ve çalışma prensiplerine göre sınıflandırılması (Fink, Voss, 1999)

META-SEZGİSEL YÖNTEM	**GENETİK ALGORİTMA**	**TABU ARAMA**	**KARINCA SİSTEMLERİ**	**BENZETİM TAVLAMASI**
Hafıza Yapısı	Popülasyon	Tabu Listesi	İzlerin Yönetimi	Sınırlı Popülasyon
Yoğunlaşma Stratejisi	Uyumluluk ve Çaprazlama	İyi çözümlerin komşulukları	İyi çözümlerden diğer ajanların da haberdar olması	Tavlama planı
Çeşitlendirme Stratejisi	Mutasyon	Yeni çözüme gitme	Alt-Sezgisel kullanımı	Tavlama planı

Tablo 2.2: Metasezgisellerin kullanım alanlarına ve çalışma prensiplerine göre sınıflandırılması (Fink, Voss, 1999) (devam)

META-SEZGİSEL YÖNTEM	GENETİK ALGORİTMA	TABU ARAMA	KARINCA SİSTEMLERİ	BENZETİM TAVLAMASI
Arama Yöntemi	Çözümleri iyileştirici	Çözüm iyileştirici	Çözüm yapıcı	Çözüm iyileştirici
	Rastsallık var	Rastsallık yok	Rastsallık var	Rastsallık var
	Gradyan değil	Gradyan	Gradyan	Gradyan
	Çözümlerin kombinasyonu	Strateji var	Strateji ve sinerji var	Strateji var
Çözüm Uzayı	Yapılandırılmamış	Yapılandırılmış (Uzaklık Matrisleri ve Komşuluk Yapıları)	Yapılandırılmış	Yapılandırılmış
Optimum Çözüm	Çözüm popülasyonu	Tek çözüm	Çözüm popülasyonu	Tek çözüm
Kaynak	Doğal, Evrimsel, Amp	Doğal, Hafızanın zekice kullanımı, Amp	Doğal, Karınca davranışları	Doğal
Kullanım Alanları	Çizge Oluşturma, Kuadratik Atama Problemleri, Bilgisayar Destekli Tasarım ve İmalat Problemleri, Görüntü İşleme, Elektronik Devre Tasarımı, Çizelgeleme.	Min. Örten Ağaç, Atölye Çizelgeleme, Üretim Çizelgeleme, Kaynak Planlama, Haberleşme, Finansal Analiz, Planlama, Enerji Dağıtımı, Moleküler Mühendislik, Lojistik, Esnek Üretim, Biomedikal Analiz	Gezgin Satıcı Problemi, Kuadratik Atama Problemleri, İş Çizelgeleme	Gezgin Satıcı Problemi, Finansal Analiz, Bilgisayar Tasarımı, Görüntü İşleme, Moleküler Fizik Ve Kimya, Çizelgeleme Problemleri, Haberleşme Şebekeleri

2.4. LİTERATÜR TARAMASI

Bu bölümde, uçakların çizelgelenmesi konusunda yayınlamış makale, bildiri ve kitaplardan bahsedilmiştir. Ayrıca uçakların çizelgelenmesi konusunda özellikle son yıllarda yapılan çalışmalara yönelik detaylı bilgiler, özet tablo şeklinde gösterilmiştir.

Literatüre baktığımızda genel olarak uçak çizelgeleme problemlerinin 4'e ayrıldığını söyleyebiliriz. Bunları; havayolu çizelgeleme, uçak bakım planlaması, filo atama ve mürettebat atama olarak gruplandırabiliriz.

Bugüne kadar, çok amaçlı problemleri çözebilen çok pistli uçakların çizelgelenmesi (ASP) için literatürde birçok uçak çizelgeleme problemi türü önerilmiştir. En yaygın çözüm yaklaşımları şunlardır: (1) karışık-tamsayı programlama (MIP) formülasyonları, standart çözücü ile çözülmüştür; (2) dal ve sınır (B & B) algoritmaları; (3) dinamik programlama (DP) yaklaşımları ve (4) sezgisel çözüm yaklaşımları.

Uçak sıralama problemi (ASP) için hem tam hem de sezgisel algoritmalar önerilmiştir. Kesin yöntemler, sorunun küçük bir örneği olan birçok araştırma makalesinde en uygun çözümü sağlar. Büyük problemler için en uygun çözümlere ulaşmanın uzun sürme ihtimali nedeniyle, son zamanlarda yaklaşık algoritmalar dikkati çekmektedir.

Literatürdeki uçak çizelgeleme problemleri (ASP) için çok sayıda sezgisel ve metasezgisel algoritma önerilmiştir. Örneğin, popülasyona dayalı sezgisel yöntem (Beasley ve diğerleri, 2001), dağılım arama algoritması(scatter sarch algorithm) (Pinol, Beasley, 2006), gelişmiş karınca kolonisi algoritması (improved ACO) (Bencheikh ve diğerleri, 2011), hem iniş hem de kalkış operasyonları için açgözlü sezgisel ve metasezgisel algoritmalar (Hancerliogullari ve diğerleri, 2013), ayrışmaya dayalı uyarlanabilir büyük komşu arama sezgisel (Vadlamani, Hosseini, 2014), yinelenen yerel arama algoritması (Sabar, Kendall, 2015), hibrit parçacık sürü optimizasyonu ve yerel arama (Girish, 2016), karınca kolonisi algoritması ve yerel sezgisel arama algoritma kombinasyonlu taklitçi algoritma (memetic algorithm) (Bencheikh ve diğerleri, 2016) Kuyruk türbülansı (Wake vortex) modellemeye dayalı karınca kolonisi algoritması (Xu, 2017), tavlama benzetimi (Salehipour ve diğerleri, 2013), yapay arı koloni algoritması (Luo, Duan, 2014). Bu çalışmaların çoğu birden fazla pistin olduğu durumlar için yapılmıştır. Dinamik durumda çizelgelem yapıldığı zaman test verisi çözümleri her durumda

optimum elde edilememiştir. Bu çalışmada karşılaştırmak için elde ettiği sonuçları kullandığımız Beasley, inen uçak çizelgeleme problemini (ALP) yer değiştirme problemi olarak dinamik bir şekilde çözmek için doğrusal programlamaya dayalı sezgisel bir algoritma ve popülasyona dayalı sezgisel bir yöntem kullanmıştır (Beasley ve diğerleri, 2004).

Ayrıca havayolu firmalarına ait birbirinden farklı konularla ilgili çalışmalar yapılmıştır. Bu konular genel anlamda; uçuş çizelgeleme, uçuşların atanması, mürettebat ataması, uçus rotalama, bakım planlama olarak ele alınabilir.

Ancak daha önceki çalısmalara bakıldığında uçus atama probleminin filo atama problemi olarak daha genel ve çözülmesi kolay bir problem olarak yapılan çalışmalarda daha sıklıkla yer edindiği görülmektedir. Bunun yanı sıra filo veya uçuş atama probleminin çok büyük boyutlu ve çözülmesi zor bir problem olması sebebiyle yer alan çalışmalarda genellikle global optimum değerine ulaşma kaygısı olmadan sezgisel ve metasezgisel yöntemler kullanılarak uygun süreler içerisinde çözülmeye çalışılmıştır. Buna rağmen bazı çalışmalarda dal-sınır yöntemi gibi en iyi sonuca ulaşmaya olanak sağlayan yöntemlerin yer aldığı, ancak bunların sadece küçük boyutlu problemler üzerinde etkili olduğu gösterilmiştir.

Bununla birlikte her geçen gün gelistirilen yöntemler ve hızla gelisen teknolojilerin sayesinde daha büyük boyutlu problemlerde çözüme daha kısa sürede ulaşılabildiği görülmektedir.

Çalışmaların çoğunda, uçak çizelgeleme problemi atölye tipi iş çizelgeleme problemi mantığı ile ele alınmıştır. Pistler ile tezgahlar, uçaklar ile de yapılacak işler benzetilmiştir. Her uçağın en erken iniş zamanı, işin hazır olma zamanı olarak değerlendirilmiştir. Buna göre uçak çizelgeleme problemi; işlem zamanlarının sıralamaya bağımlı (hazırlık zamanları sıfır) olduğu veya hazırlık zamanlarının sıralamaya bağımlı olduğu (işlem zamanları sıfır) olduğu, atölye tipi iş çizelgeleme problemi olarak da görülebilir. Literatürde genel olarak; uçakların en son iniş zamanının minimize edilmesi ya da hedeflenen iniş zamanından önceki ya da sonraki inişlerin yani sapmaların minimizasyonu amaç fonksiyonu olarak kullanılmıştır. Makine çizelgeleme problemlerinde olduğu gibi, toplam işlem zamanının (Cmax) minimize edilmesi de bir diğer amaç fonksiyonunu oluşturabilmektedir.

Uçakların yönetilmesi ile ilgili yapılan çalışmalara bakıldığında havaalanlarında uçakların çizelgelenmesi, uçuşların atanması ya da mürettebat ataması gibi konularda kullanılan yazılım sistemlerinin bulunduğu görülmektedir

Literatürde çalışmaların hedeflerine ulaşmasını engelleyebilecek düzensiz durumlara bakıldığında, çalışmaya bağlı olarak havayolu ile ilgili kaynak kısıtlarından bahsedildiği, bazı çalışmalarda ise hava alanı ve hava sahası ile ilgili kapasite sıkıntısından kaynaklanan kısıtlardan bahsedildiği görülmektedir. Çalışmaların bir kısmında ise düzensiz durumlar göz ardı edilmiş ya da çok az değinilmiştir.

Literatür incelemesine göre, uçakların iniş ve kalkışları dahil havaalanlarındaki tüm operasyonların aksamasına neden olarak; kısıtlayıcı hava koşulları, bakım sorunları, uçakların sayıca az olması, personel sıkıntısı, mürettebatın bulunmaması, gelen uçuşların zamanlamasındaki gecikmeler sayılabilir. Karşılaşılan bu durumlar, zaman zaman uçuş gecikmelerine, uçuş iptallerine ve pistlerin kapatılmasına neden olabilecek kadar büyük boyutlarda olabilmektedir.

Daskin ve Panayotopoulos'ın (1989) çalışmalarında, tek bir topla-dağıt serim yapısında uçakları rotalarına atayarak işletme kârının enbüyüklenebileceği tam sayılı doğrusal programlama yardımıyla gösterilmiştir. Ayırca, planlanan güzergahlardan elde edilecek karı çoklamak için güzergahlara en az maliyetle uçak atamayı hedeflemişlerdir. Lagrangian gevşetmesi yöntemi ve sezgisel yöntemler birlikte kullanılmıştır. Elde ettikleri sonuçlarla göre, Lagrange gevşemesinin kazançların üst sınırını sağlamada etkili olduğunu ve hedefteki tüm rotaları programda uçması gereken maksimum uçak sayısının mevcut uçakların sayısından az veya ona eşit olması durumunda iyi çözümler üretebildiğini göstermeye çalışmışlardır.

Lohatepanont ve Barnhart (2004), çalışmalarında Uçuş yapılacak bölgelerin seçimini ve seçilen uçaklara uçak tipleri atama optimizasyonu sağlayan entegre modeller ve çözüm algoritmalarından bahsetmişlerdir. Amerikada bir havayolu için gerçekleştirilen çalışma sonucunda uçakların kullanımındaki potansiyel iyileşme ve gelirde önemli bir artışın olduğundan bahsedilmiştir.

Feo ve Bard (1989), çalışmalarında Maliyetleri kontrol etmek için havayolları, tasarruf için potansiyel bir kaynak olarak bakım operasyonlarının öneminden bahsedip bakım merkezlerinin yerleştirilebileceği, kontrol gereksiniminin daha iyi karşılanabileceği uçuş programı geliştiren bir yöntem geliştirmeye çalışmışlardır. Problem, integral kısıtlamaları ile American Havayolları için minimum maliyetli çok amaçlı bir akış ağı olarak formüle edilmiş ve iki aşamalı bir buluşsal yöntem kullanılarak çözülmüştür. Çalışmanın amacı, planlayıcılar tarafından hem bakım istasyonlarının bulunması hem de çevrimsel bakım talebini daha iyi karşılayan uçuş çizelgelerinin geliştirilmesi için kullanılabilecek bir model sunmaktır. Problemde, doğrusal problemin gevşetilmesi çözüm için çok büyük olduğundan, iki aşamalı sezgisel bir yöntem kullanılmıştır.

Hane v.d., (1995), filo atama problemini yan kısıtlarıyla birlikte büyük çok ürünlü akış problemi olarak zaman genişletilmiş serim yapısında modellemiştir. Problemi çözmek için kullanılan yöntem, iç nokta algoritması, çift en dik kenarlı simpleks, maliyet pertürbasyonu, model toplama, küme-bölme kısıtlamaları üzerine dallanma ve dallanma sırasını önceliklendirme kullanılmıştır. Yazarlar ayrıca problemin boyutunu küçültmek için bazı serim ön süreç yöntemleri kullanmıştır.

Clarke v.d., (1997), uçak rotasyon problemi (ARP) için matematiksel bir formülasyon sunmuşlardır ve asimetrik seyir satıcısı problemi ile olan benzerliğini tartışmışlardır Aynı zamanda, uçak rotasyon problemini, bakım kısıtlarını sağlayan ve bağlantılı uçuşların kesintisiz uçuş değerini enbüyükleyen yan kısıtlı Eulerian tur problem olarak modellemiştir. Problem, ön süreç teknikleriyle basitleştirilmiş, Lagrangian gevşetmesi ve kısmi optimizasyon ile çözülmüş ve bir havayolunun gerçek verileri üzerinde test edilmiştir.

Desaulniers v.d., (1997), makalelerinde, günlük havaaracı rotalama ve çizelgeleme problemi ele almışlardır. Çalışmada, heterojen bir filoya ait uçaklardan elde edilen tahmini karları en yükseğe çıkaran günlük çizelgeleri belirlemekten amaçlanmış ve iki model sunulmuştur. Birinci modelin doğrusal gevşetilmesi sütun türetme tekniği kullanılarak, ikinci modelin doğrusal gevşetilmesi Dantzig-Wolfe ayrıştırması yaklaşımı kullanılarak çözülmüştür. Genel soruna tam çözümler dal-sınır yoluyla türetilmiştir. İki farklı havayolu şirketi tarafından sağlanan verilere

ilişkin hesaplama sonuçları elde etmişlerdir. Çalışmanın sonunda uygulanan yöntemlerle, kâr artışı sağlanabileceği ve bunun makul bir CPU süresi süresinde elde edilebileceği belirtilmiştir.

Gopalan ve Talluri (1998), uçak rotalama problemini US Airways için modellemiş ve üç günlük bakım kontrolü ve dengeli kontrol gereksinimi için polinom zamanlı bir algoritma kullanmıştır. Modelde, uygun uçak bakım rotalaması elde edebilmek için uçuşlar ve hatta filolar arasında değiş-tokuş yapılmıştır. Algoritmanın statik sonsuz-ufuk modeli ve dinamik sonlu-ufuk modeli olmak üzere iki farklı modelde üç günlük bakımın yanı sıra denge kontrol ziyaret gereksinimlerini de karşılayan, gün içindeki rotaları sabit olan bir grafikte bir uçak rotasını bulmak için kullanıldığından bahsetmişlerdir.

Barnhart v.d., (1998), Bütünleşik filo atama ve uçak rotalama problemi için bir dizi tabanlı model ve dal-fiyat (branch-and-price) algoritması geliştirmişlerdir. Uçak rotalama problemini ele aldıklarından dolayı, uçak rotalarının minimum maliyet kümelerini bulmayı amaçlamışlardır. Problemi çözerken, havayolu şirketleri tarafından sağlanan verilerden faydalanmışlar ve combine filo atama ve uçak rota sorunları için hesaplama sonuçları sunmuşlardır. Geliştirdikleri modelin dezavantajı, yüzlerce uçuşa sahip uçuş programının milyonlarca diziye sahip olmasıdır.

Rexing v.d., (2000), problem mevcut filo atama modellerine basit bir varyant olarak modellemişler, her uçuşa bir zaman penceresi atamışlar ve sonra her pencereyi ayrık hale getirerek modelin kalkış saatini seçmesini sağlamışlardır. Uçuş bacaklarına filo tipi ataması gerçekleştiren bütünleşik filo atama ve çizelgeleme problemini ABD içinde bi havayolu firması için incelemişlerdir. Her test senaryosunda model, temel modelden çok daha düşük maliyetli bir filo ödevi üretir ve ayrı bir analizde, modeli uçakların tasarruf potansiyeliyle programın sıkılaştırılması için kullanılır. Çalışmanın sonucunda işletmenin yıllık 50 milyon dolar civarında tasarruf ettiği ve operasyon maliyetlerini azalttığından bahsetmişlerdir.

Sriram ve Haghani (2003), çalışmalarında uçakların yeniden atanması ve bakım planlaması için bir formülasyon ve sezgisel bir yöntem sunmuşlardır. Uçakların uçuş bölümlerine yeniden atanması sırasında oluşan bakım masraflarını ve masrafları en aza indirgemeyi amaçlamışlardır. Yazarlar, hem A hem de B tipi kontrolü modellemede dikkate almış ve çözüm için rassal

araştırma ve derin ilk araştırma (depth first search) yaklaşımlarını karma olarak kullanan bir sezgisel yöntem kullanmıştır.

Sarac v.d., (2006), çalışmalarında uçakların yönlendirme sorununa odaklanmışlar ve uzun vadeli rota planlamasına yer vermişlerdir. Bununla birlikte, şiddetli hava değişiklikleri, ekipman arızaları, değişken bakım zamanı veya Federal Havacılık İdaresi'nin (FAA) zorunlu olduğu yeni tüzükler gibi stokastik olaylar, bu uzun vadeli planlar üzerinde tahribat yaratır. Bu sorunu çözmek için, bakım kaynak kullanılabilirliği kısıtlamalarını içeren operasyonel bir uçak bakım yönlendirme problemi formülasyonundan bahsetmişlerdir. Bakım slotları (slots) ve bakım istasyonlarında kullanılabilir adam-saatini dikkate alan operasyonel uçak bakım rotalama problemini modelleyerek çözmüşlerdir

Orhan (2007), uçak rotalarının bakım gereksinimleri ile birlikte ele almış, böylece uçakların etkin kullanımının yanı sıra bakım maliyetlerinin enküçüklenmesini hedeflemiştir. Belirtilen hedef doğrultusunda, uçakların bakıma girmeden önceki yasal kullanılabilir uçuş zamanlarını da enküçükleyen bütünleşik çok-amaçlı karma tamsayılı doğrusal bir modelleme yaklaşımı önermiştir. Yaklaşım, uçuş sayılarını ve/veya sürelerini de karar vericinin öncelikleri doğrultusunda dengelemekte, böylece uçak kullanım oranları ve bakım maliyetlerinin olabildiğince eşit tutulmasını sağlamıştır.

Mathaisel (1997), çalışmasında havayolu şirketlerinin uçus planlarını ve uçaklarının çizelgelemelerini yaparken yardımcı olacak ve farklı senaryolar altında da çalışmalarını sürdürebilecek optimizasyon destekli bir karar destek sistemi kurulmakla beraber havayolu firmalarının bu ve buna benzer çalısmalara olan ihtiyacının ne kadar fazla olduğunu ortaya koymuşlardır.

Jarrah ve Strehler (1999), yapılan çalısmada özellikle küçük yerleşim yerleri arasında seyahat eden yolcuların gidecekleri yere direkt uçamaması ve farklı uçaklar arasında aktarma yapma gerekliliklerini göz önünde bulundurarak, günümüzde bazı havayolu firmalarının da önemle üzerinde durdugu aktarma olmayan uçuş-şların en iyilemesi konusunu incelemiştir. Yapılan bu çalışma sonucuna göre A.B.D.'nin önde gelen bir havayolu firması için en kârlı direkt uçuş rotaları belirlenmiş ve böylece yıllık 10 milyon doları geçen bir kazançlarının olabileceği hesaplanmıştır.

Hsu ve Wen (2000), yaptıkları çalısmada öncelikle müsteri memnuniyetini dikkate alan ve servis seviyesini artırmaya yönelik bir ag tasarımı gerçeklestirmisler. Hsu ve Wen'in yapmış oldukları ağ tasarımına göre, sistemdeki belirsizliklere karsı etkin bir matematiksel yöntem olarak kullanılan Grey Teorisini ve çok amaçlı programlamayı birlikte kullanarak uçusların havayolu firmasına olan maliyeti yada yolculara olan maliyetini azaltmayı amaçlayan bir model olşturmuşlardır.

Bolat (2000), uçakların havaalanlarındaki giriş kapılarına atanmasının uçuş çizelgelerine, uçusların ise bu çizelgeye uygun sekilde gerçeklestirilebilinmesine, uçakların hizmet ihtiyaçlarının farklılıgına ve körük kapasitelerine bağımlı olarak gerçekleştirilebildiğinin göz önüne alındıgı bir sorunu incelemiştir. Problemde giriş kapılarının kullanılmadan sürelerin varyansını en aza indirebiceke karma ikili (0-1) matematiksel model kullanıllanmışlardır.

Bruinsma ve ark. (2000), makalelerinde daha küçük Avrupa sehirlerinden Londra, Paris, Frankfurt ve Amsterdam gibi önemli bağlantı noktaları kullanarak yapılan kıtalararası uçusları göz önünde tutarak bu büyük Avrupa sehirlerindeki havaalanlarının performanslarını karşılaştırıp incelemişlerdir. Karşılaştırma yaparken ulasım maliyeti ve süresi, yeniden çizelgeleme sürelerini dikkate almışlardır. Çalışmalarının sonucuna göre uçuş frekanslarının arttırılmasının yeniden çizelgeleme maliyetlerini yükseltmesi nedeni ile toplam maliyetlerin azalmasında önemli bir etkisinin olmadığı ortaya çıkmıştır.

Drezner T., Drezner Z. (2001), havayollarında karşılaşılan önemli aktarma merkezlerinin seçimi ile ilgili problemi ele almışlardır. Çalışmanın giriş kısmında öneme sahip merkezler arasında yapılacak olan uçusların, öneme sahip olmayan iki nokta arasında yapılacak olan uçuşlardan daha düşük maliyete sahip olduğunu belirtmişlerdir. Bu duruma rağmen, çalışmada yolcuların en fazla bir önem merkezini kullanmaları ve bu sayede kat ettikleri mesafeyi en aza indirmeleri hedeflenmiştir. Problemi çözmek için kullanılan algoritma sayesinde birçok önem merkezi arasından yolcuların minimum seyahat mesafesi hedefini gerçekleştiren merkezler tespit edilmiştir. Çalışmanın sonunda problemin yolcu sayısının maksimize edilmesi veya önem merkezlerindeki trafiğin minimize edilmesi gibi farklı amaç fonksiyonlarını ele alacak biçimde rahatlıkla değiştirebileceğinden bahsetmişlerdir.

Stojkovic ve ark. (2002), yayımlanan bu makalede uçuş çizelgelenmesiyle ilgili problem uçuş personeli, dinlenme süreleri, yolcu aktarmaları ve bakım gibi konular ele alınarak incelenmiştir. Yapılan bu çalışmada uçak ve uçuş personelinin planlanması ağ modelleri kullanılarak ele alınmıştır ve modelin sistemde beklenmeyen karmaşıklıklara karşı etkin olarak çalıştırılabileceği öngörülmüştür. Makalenin uygulamalarla ilgili bölümünde, hazırlanan bu modelin hem küçük hem de büyük ölçekli havayolu şirketlerinin uçus planlamalarında iyi sonuçlar verebileceği belirtilmiştir.

Pulugurtha, Nambisan (2003), çalışmasında havayolu kârlılık yönetimi için genetik algoritma kullanılarak bir karar destek sistemi kurmuşturr. Problemi daha gerçekçi olarak ele alabilmek amacıyla taleplerdeki değişkenliklerin de hesaba katırlmasını sağlamıştır. Problemin modelleme bölümünde doğrusal tamsayılı programlama kullanılmıs ve çözüm kısmında genetik algoritmalardan faydalanılmış ve C programlama dili kullanılarak kodlanmıştır. Çalışmanın son bölümüne göre birçok vaka çalışması yapılmış ve kurulan karar destek sisteminin beklenen sonuçları verdiğinden bahsedilmiştir.

Hsu ve Wen (2005), yapmıs oldukları çalışmalarında, uçuşların sıklıklarının belirlenmesinde kullanılacak bir modelleme üzerinde durmuşlardır. Modellerini olustururken havayolu pazar payı modeli, uçus frekansı belirme modeli ve bulanık mantık tabanlı rekabet etkilesim modeli olmak üzere üç alt model kullanmışlardır. Ana modelin isleyisine göre; bir önceki modelin sonucunun bir sonraki modelde kullandıkları, birbirleriyle sürekli etkileşimli olan üç alt modelin kullanılarak ana modeli çalıştırdıkları görülmektedir. Çalışmanın sonuçlarından modelin özellikle orta boy işletmelerin uçuşlarının planlanmasında uygulanabilir olduğu görülmektedir. Ayrıca rekabetçi ve belirsiz pazar şartlarında da havayolu şirketlerine karar vermede esneklik kazandırdığından bahsetmişlerdir.

Wu (2005), bir havayolu firmasının uçuş planlamalarının hedeflenen zamanda gerçeklemesi ve operasyonel güvenilirlikleriyle ilgili değerlendirme yapmak için yeni bir yaklasım ortaya koymuştur. Çalışmada genel olarak uçuşlar sırasında yaşanabilen gerçek gecikme verileriyle simülasyon çalışması sonucunda ortaya konulan farklı senaryolara ait doğal gecikmeler arasındaki farklar ele alınmıştır. Çalışmanın sonucuna göre havayollarında uçuş çizelgelemelerin dışarıdan gelen etkilerle nasıl etkilenebildiği, ne kadar hassas oldugu ve bu

çizelgelemelerin ancak sınırlı güvenirlik-esneklik oranını sağlayarak olusturulabilecegi ortaya çıkmıştır.

Sherali ve ark. (2006), çalışmalarında havayolu filosu atama, modelleri ve algoritmaları üzerinde durmuşlardır. Filo atama problemini haftalık filo modeli, birleştirilmiş uçuşlar için filo atama modeli, optimizasyon harici filo atama çözümleri gibi farklı alt çalışmalar yaparak ele almışlardır.

Wang ve Regan (2006), çalışmalarında yapılacak bir uçuş için uçaklar arasında değişime gidildiği zamanki durumlar için dinamik kârlılık yönetimi üzerinde çalışma yapmışlardır. Çalışmalarında öncelikli olarak aynı kokpite sahip olan filodaki uçaklar arasında değişime gidilmesi gerektiğinden uçuşu hiçbir şekilde etkilemeyecek şekilde eşleştirilme sağlanmıştır. Daha sonra ele alınan problem üzerinde, uçaklar arası değişiklik ihtimalinin sabit bir değer olduğu varsayılarak çalışma yapılmıştır. Çalışmanın sonucunda elde edilen sayısal sonuçlar, geliştirilen yöntem sayesinde havayolu sirketlerinin kârlılık yönetimindeki yeteneklerini elde edilen gelirin arttırılması yönünde büyük ölçüde arttırabildiği gösterilmiştir.

Bélanger ve ark. (2006), çalışmalarında periyodik olarak filo atama problemini inceleyip, ele almışlardır. Yapılan çalışmada amaçlanan, uçuşlara atanacak en kârlı uçak tipini belirlemek ve uçuş çizelgelemesini de bu doğrultuda gerçekleştirmektir. Problem lineer olmayan tam sayılı ağ akış modeli olarak kurulmasına rağmen daha sonra bu problem için dal sınır yönteminden geliştirilerk oluşturulan dal-maliyet yöntemiyle çözülmüştür. Çalışmalarının son bölümünde model üç gerçek veri seti kullanılarak test edilmiştir ve sonuçlar geliştirilen yöntemin gerekli uçak sayısında azalmayı sağlayabildiğini ve bunun da direkt olarak kârlılık oranını arttırdığı tespit edilmiştir.

Yan ve ark. (2007), çalışmalarında Tayvan Havayolları filosu için daha iyi rota ve uçuş çizelgelemesi belirlemeye yardımcı olabilecek değişken olabilen pazar paylarını da ele alan kısa dönemli uçus çizelgelemesi modeli oluşturmuşlardır. Çalışmalarında kullanılan model lineer olmayan karma tamsayılı programlama olarak kurulmuş, ardından modelin etkin bir biçimde çözülebilmesi için sezgisel bir yöntem kullanılmıştır. Elde edilen sonuçlar incelendiğinde sezgisel yöntem kullanılarak bulunan sonuçların eniyi sonuçlara çok yakın olduğu gözlemlenmiştir.

Bennell ve ark. (2011), uçak iniş ve kalkışlarını programlamak için kullanılan dinamik programlama, şube ve sınırlama, buluşsal yöntemler ve metasezgiseller gibi operasyon araştırma tekniklerinin kapsamlı bir incelemesinin yapıldığı Uçak çizelgeleme problemleri (ASP) hakkındaki son bir anketi incelemişlerdir.

Bianco ve ark. (1987, 1997) Terminal Manevra Alanı (TMA) içinde gelen uçakları sıralamak için tamsayı programlama kullanmışlardır. Uçak sıralama ve kombinatoryal eniyileme probleminin NP zor olduğunu göstermişlerdir. Ayrıca, hazır zamanlar sıfır olduğunda, problemin Asimetrik Seyyar Satış Uzmanı Problemi'ne (ATSP) indirgendiğini bulmuşlardır. Çözüm olarak, Lagrangian alt sınırlama teknikleri kullanan bir şube-bağlı strateji ve çözüm sürecinde elde edilen alt-dizilerin özelliklerine dayanan bir bölümleme yaklaşımı önermişlerdir. Yine de çalışmada uçak operasyonları için zaman kısıtlamalarını ve belirli ardışık olmayan operasyonlar arasında gerekli ayırma süreleri ihmal edilmiştir.

Beasley ve ark. (2000), uçak (düzlem) iniş çizelgeleme problemini ele almışlardır. Tekli pist için sorunun karışık tamsayı sıfır bir formülasyonunu sunmuşlardır ve onu çoklu pist haline genişletmişlerdir. Problemi, doğrusal programlama tabanlı ağaç araştırması kullanılarak çözmüşlerdir. Tekli ve çoklu uçaklarda uçağın sıralama problemi için karışık bir tamsayılı doğrusal program (MILP) modeli önermiş ve uçak iniş problemleri için İlk Başlamalı İlk Hizmetin (FCFS) bir sezgisel algoritması uygulanmıştır (ALP).

Dear ve ark. (1989, 1991), statik ve dinamik Uçak İniş Problemi (ALP) için Kısıtlı Konum Değiştirme (CPS) sezgisel yöntemini sunmuşlardır. Statik ve dinamik durumlarda uçak iniş problemi üzerinde çalışmışlardır. Kısıtlı Konum Değiştirme (CPS) metodolojisi incelenmiş ve etkinliği test edilmiştir. Hızlı zaman simülasyonu kullanılarak algoritmanın etkinliğini göstermek için, CPS ve FCFS arasındaki performans karşılaştırması gerçekleştirebilimiştir. CPS'nin iki basamağı vardır: önce, akışı maksimize eden dizileri arar, sonra minimum gecikme ile maksimum akışlı çözüm seçilir. 500 uçak ve bir pisti içeren hesaplama sonuçları, sezgisel olarak FCFS yaklaşımından daha küçük gecikmeler elde edildiğini göstermektedir

Neuman ve Erzberger (1990) çalışamlarında hava trafik kontrol sistemlerinde kullanılan FCFS yaklaşımının performansını, gelişmiş zaman (TA) tekniği ve CPS sezgisel yöntemini kullanarak

değerlendirdiler. Öncelikli olarak başlangıç olarak gelişler için en basit sıralama stratejilerinden biri olan FCFS kullanılmıştır.

Optimizsayon aşaması için, hava trafiğinin yoğun olduğu zamanlarda belirli ana uçakların hızlandırılması için gelişmiş zaman tekniği (TA) kullanılmıştır. Cps algoritması uçakların farklı uçak sınıfları için farklı ayırma gerekliliklerinden yararlanması kuralını aynı zamanda optimizasyon amaçlı olarak da kullanmaktadır. Algoritmaların istatistiksel özelliklerini belirlemek için, rastgele seçilen trafik örnekleri oluşturulmuştur. FCFS'nin adil bir düzen oluşturduğu saptanmıştır; TA metodu, planlanan her bir uçağın gecikmesini azaltmak için makul sonuçlar vermektedir; CPS, büyük uçak gruplarıyla yoğun trafiğin en etkili olduğu ve uçaklar başına ortalama gecikmeyi en düşük seviyede tutmada başarılı olduğu anlamına gelir.

Venkatakrishnan ve ark. (1993) havaalanında ayrım zamanlarını gözlemleyerek, hava trafik kontrol prosedürlerini iyileştirip, mevcut kapasitenin daha verimli kullanılması konusunda çalışmışlardır. Havayolu çizelgeleme problemini hava sahasında bekleyen benzer uçakların karşılaştırılması olarak ele almıştır. Tek bir pisti göz önüne alarak uçak inişleri için hava trafiği gecikmelerini incelemişlerdir. ASP için statik ve dinamik modeller sunmuşlardır. Statik model açısından, zaman penceresi kısıtlamalarını değiştirerek Psaraftis'in (1980) çalışmalarını uygulamışlardır. En erken ve en geç zamanları dikkate alarak sıralama yapıp gelişmenin olduğunu gözlemlemişlerdir.

Bianco ve arkadaşları (1997), TMA'daki uçakların birden çok pistte çalıştırılması için sıraya bağlı kurulum süreleri ve çıkış tarihlerine sahip deterministik bir iş atölyesi çizelgeleme modeli önermişlerdir. Çalışmalarında En hızlı arama sezgiselliği (CSH) adı verilen hızlı dinamik bir yerel buluşsal algoritma geliştirmişlerdir. Modelin performansı ve algoritması, uygulamanın yapıldığı havaalanlarının TMA'ları için gerçek veri setlerini kullanarak analiz etmişlerdir. Yapılan analizler sonucunda, FCFS yöantemi ile kıyaslanınca ortalama olarak gecikmenin en az %40 oranında azaltıldığını ve TMA'nın kapasitesinin yaklaşık% 30 oranında arttığını göstermektedir.

Carr ve diğ. (2000), hava trafik kontrol otomasyonu için sıralamada ve planlama algoritmalarında havayolu varış tercihlerini göz önüne alan, öncelikli zamanlama kavramını geliştirmişlerdir. Bu yöntem, piste tahmini varış zamanına dayalı olarak FCFS sırası yerine,

tercih edilen bir sıraya göre gelen bir uçak kümesini zamanlama yöntemidir. Metodun fizibilitesini değerlendirmek için hızlı zamanlı simülasyon kullanılmıştır. Yapılan analizlere göre, önerilen zamanlama yönteminin, banka varış sırasındaki sapmaları azaltmak için FCFS zaman çizelgesinden daha başarılı olduğunu, ancak zamanlama verimliliğinde çok az veya hiç azalmaya neden olmadığını göstermektedir.

Bauerle ve ark. (2007), havaalanına gelen uçakların sıralanmasını ve havalimanı kapasitesi üzerindeki etkilerini incelemişlerdir. Geliş sürelerini Poisson süreci ile modellenebileceğini varsayarak kullanmışlardır. Sadece bir pisti modellemek için bir M / SM / 1 kuyruğu (bağımlı servis zamanlarına sahip) kullanmışlardır. İki pist durumunu incelerken ise rastgele bölme, adi para sayma, en az yükleme gibi farklı sezgisel yöntemler kullanmışlardır. Kullanılan yöntemlerin performasnları neden oldukları gecikmeye göre karşılaştırılmıştır.

Soomer ve Franx (2008), havayollarının maliyetini dikkate alarak gelen uçakların çizelgesinin belirlemesi problemini tek pist için çözmüşlerdir. Modelin karışık tamsayılı programlama formülasyonu yanında, beklenen geliş zamanlarına göre uçuşların sıralmasıyla elde edişlen olası başlangıç çözümünü dikkate alane yerel arama sezgiseli geliştirmişlerdir. Başlangıç çözümünü geliştirmek için ikili yer değiştirme ve komşuluk değiştirme yöntemleri uygulanmıştır.

Balakrishnan ve Chandran (2010), CPS yönteminin uçak operatörleri arasında eşitliği korumaya yardımcı olduğunu ve iniş sürelerinin doğru tahmin edilme derecesini arttordığını yazmışlardır. CPS ve diğer kısıtlamalar altında pist planlaması için dinamik programlama algoritmaları geliştirmişlerdir. Önemli bir hedef olarak, bir uçak dizisinin tamamlanma süresini en aza indirerek, pist çıkışı veya eşdeğer olarak en üst düzeye çıkarılmasını hedeflmişlerdir. Çalışmalarını tek pist için yapmış olmalarına rağmen, çoklu pist koşullarına da çözüm sağlamışlardır.

Yu ve ark. (2011) tek bir pistli ALP için Cellular-Automata tabanlı Optimizasyon (CAO) adlı bir algoritma önermişlerdir. Algoritma; uçak iniş sürecinin Hücresel Otomasyon (CA) modeli kullanılarak simüle edildiği, iyi bir iniş sırası elde etmek ve ileri optimizasyon olarak, önceki aşamada elde edilen iniş sırasının stokastik bir yerel arama yoluyla gerçekleştirilmesi gibi 2 ana adımdan oluşmaktadır. Geliştirilen yöntem, LP tabanlı Ağaç Arama, Karınca Kolonisi

Optimizasyonu (ACO), ve Biyonomik Algoritma (BA) gibi birkaç yöntemle karşılaştırlmıştır. CAO'nun çoğu durumda çözümün kalitesi ve hesaplama hızı açısından daha üstün olduğu sonucuna varılmıştır.

Boysen ve Fliedner (2011), tek piste inişlerin yerdeki personelin iş yükü üzerine etkilerini araştırmışlardır. Zemin personelinin iş yükünün azaltılması için inen uçak içinde taşınan yolcu sayısı, her havayolunun inen uçak sayısı, havayolu şirketi başına yolcu sayısı gibi hedeflerin azaltılması amaçlanmıştır. Uçakların, en erken ve en erken iniş saatleri dikkate alınmamıştır ve normalde türbülans etkisine bağlı ayrı hesaplanan ayırma zamanları eşit alınmıştır. Ve sonuç olarak, her bir amaç fonksiyonu için, matematiksel modellerle birlikte sezgisel çözümler gösterilmiştir.

Salehipour, Modarres ve Naeni (2013) çalışmalarında gelen uçakları pistlere atarken karma tam sayılı hedef programlama modeli geliştirmişlerdir. Sonra da NP-Zor olan problemin karmaşıklığına göre benzetim tavlamasını kullanarak, hibrid metasezgisel algortima geliştirmişlerdir. Çalışmalarında beklenen iniş süresi ile gerçekleşen iniş süresi arasındai sapmayı minimize etmeyi amaçlamışlardır. Çalışmalarının sonucunda geliştirdikleri algoritmanın 100 uçak sayısına kadar olan problemler için optimal sonuç bulabildiğini ve aynı zamanda 500 uçağa kadar ve 5 pist kapasitesine sahip problemler için kısa sürede diğer sonuçlarla karşılaştırılabilir, iyi kalitede sonuçlar elde edebildiklerinden bahsetmişlerdir.

Bennell, Mesgarpour ve Potts (2017) uçak iniş problemini statik ve dinamik durumun her ikisi için de incelerek ele almışlardır. Ele aldıkları problemi tek piste inen uçaklar için çalışmışlardır. Her gelen uçak için ard arda inişlerde zaman penceresi kısıtlarını, uçakların büyüklüğüne göre minimum ayırım zamnlarını dikkate almışlardır. Geliştirdikleri çok çok amaçlı formül ile piste gelen uçakların akışını, gecikme ve erken gelme sürelerini ve uçak manevralarında ortaya çıkan yakıt maliyetini dikkate almışlardır. Dinamik çizelgeleme yaklaşımı sayesinde, geliştirdikleri algoritma yeni uygun durumdaki uçakları hesaba katarak, önceki çizelgelemede periodik güncellemeler yapmaktadır. Londra Heathrow havalimanından rastgele test verileri ve gerçek veriler kullanarak dinamik programlama ve yerel arama uygulamalarını statik ve dinamik problem üzerinde araştırmışlardır.

Xu (2017), çalışmasında gelen uçak çizelgeleme problem için wakevortex modelleme (WVM) yöntemine dayanan bir karınca kolonisi (AC) algoritması önermektedir. Elde ettiği sayısal sonuçların yeni geliştirdiği yöntemin CPLEX ve genel Karınca Kolonisi yaklaşımına göre daha iyi performans gösterdiğini doğruladığından bahsetmiştir.

Sama, D'Ariano, Paciarelli(2017) , trafik sıkışıklığı durumuna göre yoğun havaalanlarında kalkış ve iniş işlemlerinin gerçek zamanlı optimizasyonunu ele almışlardır. Çalışmalarında karışık tamsayılı doğrusal programlama formülasyonları; yüksek performans modelleme hassaslığı ile emniyet kısıtlamalarını göz önünde bulundurarak, pratik çeşitli performans göstergeleri arasındaki dengeyi araştırmak üzere önerilmiştir. Milano Malpensa ve Roma Fiumicino'nun iki önemli havalimanı için çeşitli rasgele iniş ve kalkış uçaklarının bozukluklarını simüle ederek denemeler gerçekleştirilmiştir. Pratik boyutlu örnekler, ticari olarak da kullanılan bir çözücü aracılığıyla çözülmüştür. Eniyileştirilmiş çözümler, yaygın olarak kullanılan bir çizelgeleme kuralı ile karşılaştırılmıştır.

Jamili A. (2017), çalışmasında, filo atama problemini göz önünde bulundurarak, entegre uçak rotalama ve çizelgeleme için yeni ve karışık bir tamsayı matematiksel model önermiştir. Büyük ölçekli problemler için rasyonel bir zaman aralığında iyi çözümler bulmak için Benzetim tavlaması (SA) temelli sezgisel bir algoritmadan bahsetmiştir. Ayrıca bazı örnekleri rastgele oluşturmuş ve önerilen buluşsal algoritmayı, sonuçları optimum olan çözümlerle karşılaştırarak doğrulamaya çalışmıştır. Sağlam ve sağlam olmayan çözümlerin etkileri incelenmiş ve son olarak, SA ve Parçacık Sürüsü Optimizasyonu (PSO) ile karşılaştırıldığında daha etkili bir çözüm getiren bir hibrid algoritma oluşturulmuştur.

Lieder, Stolletz (2016) çalışmalarında genel pist yapılandırmalarıyla birlikte uçak çizelgeleme problemi için bir optimizasyon yöntemi sunmuşlardır. Heterojen ve birbirine bağımlı pistler için uçak çizelgeleme problemini (ASP) analiz etmişlerdir. Problemi modellemek için karışık tamsayılı programlama ve dinamik programlama yaklaşımını uçak çizelgeleme problemini optimum şekilde çözmek için kullanmışlardır. Büyük ölçekli problemleri çözmek için dönme planlama ufku sezgiselini (RPH) sezgiselini geliştirmişlerdir. RPH sezgisel yönteminin çok kısa sürede optimum yakın sonuçlar verdiğinden bahsetmişlerdir.

Wu, Sun, Qu (2016), çalışmalarında bir uçak takımının inişi için modelleme ve sıralama yaklaşımı önermişlerdir. Çalışmada öncelikle inen uçak takımının iniş sıralama probleminin (SPLTA) iniş misyonunun her prosedürü tanıtılmıştır. Terminal alandaki trafik ve uçakların karaya iniş başarısızlığına (FLA), SPLTA ile yakın ilişkisi sebebiyle daha fazla önem verilmiştir.

Sonra Uçaktaki tek bir uçağın durum değişikliği ve terminal alanındaki uçaklar arasındaki alışverişler dikkate alınır. Genel SPLTA, gerçekçi kısıtlamalara tabi bir maliyet fonksiyonu ile bir optimizasyon problemi olarak formüle edilmiştir. SPLTA'yı çözmek için, bir uçak takımının en uygun sırayla inmesini sağlamak için karınca kolonisi yöntemi (DSAAC) kullanan dinamik bir sıralama algoritması önerilmiştir. Deneylerde, SPLTA, "en az yakıtla ilk hizmet" (LFFS) ilkesine dayanan yöntem, karınca koloni optimizasyon algoritması (ACO) tabanlı yöntem, karınca kolonisinin statik sıralama algoritması (SSAAC) ve DSAAC kullanılarak çözülmüştür. DSAAC'ın, maliyet fonksiyonunu ve iniş zamanı tüketimini en aza indirgemede diğer üç yöntemden daha iyi performans gösterdiğini göstermiştir.

Bencheikh, Boukachour, Alaoui (2016) uçak iniş problemini dinamik durum için ele almışlardır. Problemin doğrusal bir tamsayı olarak matematiksel formülasyonunu sunmuşlardır. Problemi çözmek için karınca koloni algoritmasını ve yerel gelişme sezgisel algoritmasını birleştiren bir memetik algoritma uygulamışlardır. Yapıcı özelliğinden dolayı problemi çözmek için yapay arı kolonisi algoritmasından yararlanmışlardır ve bu algoritma, statik durumda ALP'de uygulanan Geliştirilmiş ACO'nun bir uyarlamasıdır. Geliştirdikleri algoritmayı 10-50 uçak için ve 1-5 pist içeren havalanlarıları için internetteki örnek veri setlerinden faydalanarak test etmişleridir ve bulgularını Beasley ve ark.'nın sunduğu üç yaklaşımla karşılaştırmışlardır.

Heidt, Helmke, Kapolke, Liers ve Martin (2016) çalışmalarında, belirsiz koşullar altında pist çizelgeleme problemini incelemişlerdir. İlk olarak, belirsizlikleri göz ardı eden matematiksel optimizasyon modelleri sunmuşlardır. uçağın planlanmış olup olmadığını, hangisinin planlandığını her ayrık noktaya göre hesaplamışlardır. Ardından, her planlama adımında belirsizlikler hesaba katılmışlardır. Daha sonra, kararlı planlara götüren çözüm yaklaşımlarını tasarlamak için farklı sağlam optimizasyon yöntemleri uygulamışlardır. Bu optimizasyon

yaklaşımları, bir simülasyon aracıyla bütünleştirilir ve farklı trafik senaryolarında değerlendirilmiştir.

Faye (2015), çok pistli uçaklar açma sorununu incelemiştir Amaç, gelen havayolu şirketlerini havaalanında mevcut pistlere atamaktır. İniş saatleri önceden tanımlanmış zaman aralıklarında bulunur ve iki ardışık iniş arasındaki güvenlik ayırma kısıtlarının sağlanması gerekir. Bu tarz problemleri çözmek için yeni bir yaklaşım önmermiştir ve yöntem, ayırma zamanı matrisinin yaklaşık ve tam zaman ayrımına dayanmaktadır. Ayırma matrisi, 2 matrisi derecelendirme yöntemi ile yakınlaştırılır. Bu durum, yakınlaştırma matrisinin seçimine bağlı olarak alt sınırlar veya üst sınırlar sağlar. Bu sınırlar, kısıt oluşturma algoritmasında, tam veya sezgisel olarak, sorunu çözmek için kullanılır. Kullandığı yönmtemin etkinliğini göstermek için 500 uçağa kadar olan açık problemler üzerinde hesaplama testleri yapmıştır.

Gürkan, Gürel, Aktürk (2016) çalışmalarında havayolu planlaması, planlama problemleri arasındaki derin ilişkileri vurgulamayı amaçlamışlardır. Beklenmedik gecikmelerin etkisini azaltmak için; çizelgeleme tasarımı, filo atama ve uçak rotalama problemlerini yolcuların bağlantıları sırasında günlük planlama ufku içinde entegre ederek bağlantılı uçuş yolcularının bağlantı hizmetlerine şans kısıtları da dahil edilmiştir. Hizmet seviyeleri tesadüfi kısıtlamalarla sağlanmaktadır. Doğrusal olmayan yakıt tüketimi ve ikinci derece konik reformasyon ile işlenen objektif fonksiyondaki CO2 emisyon maliyet terimleri nedeniyle doğrusal olmayan karmaşık tamsayılı programlama model önermişlerdir. Çalışmanın temel katkısı, havayolu operasyonlarının bu üç aşamasının entegre bir modelinde ilk kez seyir süresi kontrolünü hesaba katmaktır. Ayrıca, daha büyük problemleri çözmek için 2 sezgisel yöntem önermişlerdir. Ve Amerika'da bulunan büyük bir havayolundan elde edilen gerçek veriler kullanılarak hesaplama sonuçları, önerilen çözüm yöntemlerinin uygulanmasında potansiyel karlılığı göstermek için sunulmuştur.

Samà, D'Ariano, Pacciarelli (2014) çalışmalarında; uçak çizelgeleme ve yönlendirme gerçek zamanlı problemini ele almışlardır. Makalede, yoğun bir Terminal kontrol alanı (TCA)'da hava trafiği izleme ve denetimi için etkili karar destek araçlarının geliştirilmesi sorunuyla ilgilenmişlerdir. Bu amaçla, merkezi ve dönen ufuk trafiği kontrol paradigmaları uygulanmış ve karşılaştırılırmıştır. Matematiksel formülasyon olarak, alternatif grafikler temelli TCA'daki

hava trafik akışlarının ayrıntılı bir modelini kullanmışlardır ve bu da genelleştirilmiş ayrık grafiklerdir.

Uçak planlaması ve (tekrar) yönlendirme yaklaşımları için, hava trafik kontrolörlerinin davranışını yerine getiren İlk giren ilk çıkar (FIFO) kuralı, sabit rotalar için dal sınır algoritmasının da dahil olduğu, uçak çizelgeleme ve yeniden yönlendirme için kombine bir dal sınır algoritması, tabu arama algoritması ve eş zamanlı çizelgeleme ve yönlendirme için karışık bir tamsayılı doğrusal programlama formülasyonu gibi çeşitli optimizasyon tabanlı algoritmalar ile karşıaştırılmışlardır. Gerçek dünyadaki bir havaalanı vaka çalışması Milano Malpensa için çeşitli varsayımsal bozulma senaryolarını simüle etmişlerdir ve önerilen zamanlama ve yönlendirme yaklaşımları, farklı senaryolardaki performanslarına göre karşılaştırmışlardır. Trafik durumlarının bozulması, birden fazla gecikmeli gelen / kalkan uçak ile geçici olarak bozulan bir pist simüle edilerek üretilmiştir. Genel olarak, optimizasyon yaklaşımlarının uçak gecikmesinin minimize edilmesi açısından FIFO'ya kıyasla çözümleri önemli ölçüde geliştirdiği bulunmuştur. Bununla birlikte, pratik uygulamalar için doğru yaklaşımın ve paradigmaların belirlenmesinde bazı karmaşıklıklar bulunduğundan bahsetmişlerdir.

Samà, D'Ariano, Corman, Pacciarelli, (2016), makalelerinde, kaliteli çözümlerin hızlı bir şekilde bulunma olasılığını artırmak için AGLIBRARY çözücüsünde (karmaşık yönlendirme ve çizelgeleme problemleriyle uğraşmak için gelişmiş optimizasyon çözücüsü) uygulanan bir dizi algoritmik iyileştirmeyi sunmaya çalışmışlardır. Önerilen çerçeve, kesilmiş dal ve sınır algoritması ile hesaplanan, sabit rotalarla uçak çizelgeleme problemi (her havaaracı tarafından geçilecek kaynaklar göz önüne alındığında) için iyi bir başlangıç çözümü ile başlar. Daha sonra terminal kontrol alanında bazı uçakların yeniden yönlendirilmesiyle çözümü geliştirmek için bir metaheuristik uygulamışlardır. Değişken komşuluk arama, tabu arama ve hibrid şemalar üzerine kurulu yeni meta-sezgisel algoritmalar tanıtılmıştır. Hesaplamalı deneyler, bir çok uçağın gecikmesi ve geçici olarak aksatılmış pistler de dahil olmak üzere çeşitli istikrarsızlıklar altında bir İtalyan havaalanı kontrol bölgesi üzerinde gerçekleştirmilerdir. Kullandıkları meta-sezgisel algoritmayı, ticari bir çözücü ile karşılaştırdıklarında ve Aglibrary'ın daha önceki sürümleriyle karşılaştırdıklarında, küçük bir hesaplama süresi içinde oldukça iyi sonuçlar ürettiğinden bahsetmişlerdir.

Lieder, Briskorn, Stolletz (2015), makalelerinde pozitif hedef iniş saatleri ve sınırlı zaman pencereleri bulunan uçaklar için çok sayıda bağımsız pist üzerinde en uygun iniş çizelgelerini oluşturabilen yeni bir algoritmayı sunmuşlardır. Algoritma, pozitif hedef iniş süreleri ve sınırlı zaman pencereleri ile birden fazla pistte farklı uçak sınıflarıyla etkili bir şekilde Uçak iniş problemelerini (ALP) çözebilen dinamik bir programlama mantığı üzerine çalışmaktadır. Yaptıkları sayısal inceleme sonucuna göre, algoritmanın hızlı ve optimal bir şekilde büyük problem örneklerini çözdüğünü ve standart MIP çözücü CPLEX'i geçtiğinden bahstemişlerdir. Çalışmada aynı zamanda alfgortimada kullnadıkları yeni üstünlük kriterinin dinamik programlama yaklaşımının performansını önemli ölçüde geliştirdiğini göstermişlerdir. Yaptıkları çalışmaya göre, 100 taneye kadar uçağın bulunduğu problemlerin birkaç saniye içinde en iyi çözüme ulaşabildiğinden bahsetmişlerdir.

Ghoniem, Farhadi, Reihaneh (2015) çalışmalarında çok pistli havaalanlarına uçak sıralaması problemleri için etkili bir dal-fiyat (B&P) algoritmasını sunmuşlardır. Bu yaklaşım, hesaplama zorlu örnekler seti üzerinde klasik bir 0-1 karışık tamsayılı formülasyonu çözülürken, sorunun işlemlenebilirliğini birkaç derecede arttırır. Algoritma, B&P algoritmasının hesaplama etkinliğinin merkezinde yer alan sütun üretim altproblemini, gelişmiş bir dinamik programlama prosedürü kullanarak, uçak zaman pencereleri ve ayrımsız ayırma zamanlarıyla temel bir en kısa yol problemi olarak çözmektir. Ayrıca çalışmalarında, önerilen dinamik programlama prosedürünü ve dolayısıyla genel B & P algoritmasını hızlandırmaya katkıda bulunan algoritma özelliklerini değerlendirdiklerinden bahsetmişlerdir.

Xu (2017), yaptığı çalışmada başlangıçta gelen uçakların sıralanması problemi (ASP) için kanat iz türbülansı (wake vortex) modelleme (WVM) yöntemine dayanan bir Karınca Kolonisi (AC) algoritması önermektedir. Bu durum modellemede iz türbülansının 2. derece matrisiyle modellenmesiyle oluşturulmuştur. Sayısal sonuçları kıyaslayarak geliştirmiş oldukları algortimanın, CPLEX, genel AC algoritması ve yaklaşım algoritmasından daha iyi performans gösterdiğinden bahsetmişlerdir.

Girish (2016) makalesinde, uçak iniş problemini (ALP) çözmek için, döner bir ufuk çerçevesinde hibrid bir parçacık sürüsü optimizasyon algoritması (RH-HPSO-LS) önermektedir. Hava trafik kontrolünde ALP önemli bir optimizasyon problemidir ve NP-hard

olarak iyi bilinir. Sorun, gelen uçakları havalimanındaki pistlere atamak ve her uçağa iniş saati atamaktan ibarettir. Her bir uçağın yakıt tüketiminde en hızlı yakıt tüketim hızına bağlı olarak belirlenen optimum hedef iniş süresi vardır ve bu mesafeden sapma sapma miktarı ile orantılı bir cezaya neden olur. Her bir uçağın iniş saati belirli bir zaman aralığında sınırlandırılmış ve önceki uçakları ile minimum ayırma süresi gereksinimini karşılamalıdır. Amaç, uçakların iniş saatlerinin ilgili hedef iniş saatlerinden sapması nedeniyle toplam ceza maliyetini en aza indirgemektir. Önerilen algoritmanın performansı, 500 uçak ve 5 piste kadar bir dizi kıstas örneği üzerinde değerlendirilmiştir. Önerilen RH-HPSO-LS algoritmasının performansı, RH çerçevesi olmadan uygulanan HPSO-LS algoritması ve mevcut literatürdeki yaklaşımlar ile karşılaştırılmıştır. Ayrıca, önerilen algoritmanın performansı, literatürden uyarlanan iki diğer PSO varyantı ile ve CPLEX yazılımı kullanılarak elde edilen sonuçlar ile de karşılaştırılmıştır.

Sabar ve Kendal (2015) makalelerinde gelen uçakların çizelgelenmesi problemi (ALP) için yinelenmiş bir yerel arama (ILS) algoritması önermişlerdir. ILS, yerel bir optimum çözüm bulmak için bir yerel arama prosedürünü ardı ardına başlatan tek bir çözüm tabanlı arama metodolojisidir. Aynı zamanda bir pertürbasyon operatörü kullanarak yerel optimumdan kaçıp ve yerel arama prosedürü için yeni bir çözüm sağlamak için mevcut çözümü değiştirmişlerdir. Farklı problemler ve / veya durumlar farklı özelliklere sahip olduklarından, ILS'nin başarısı yerel arama, pertürbasyon operatörü (lar) ve pertürbasyon kuvveti üzerine oldukça bağımlı ve önemli olduğundan bahsetmişlerdir. Bu sorunları çözmek için dört pertürbasyon operatöründen ve algoritma ilerledikçe değişen zamanla değişen bir pertürbasyon gücünden yararlanmışlardır. Yerel arama olarak değişken mahalle iniş algoritması kullanmışlardır. Önerdikleri ILS.nin , hem çözüm kalitesi hem de hesaplama süresi açısından verimliliğini gösteren, bilimsel literatürden alınan ALP ölçüt örnekleri için yüksek kaliteli çözümler üretebildiğinden bahsetmişlerdir.

Vos, Santos, Omondi (2015), uçak çizelgesini geliştirme soruna (ASRP) yenilikçi bir dinamik modelleme çerçevesi sunmuşlardır. Bu konuda yapılan önceki çalışmaların sıklıkla statik bozulma test senaryolarından yararlandıklarından ve bir kerelik bir değerlendirmede bozulan olayların simülasyonunu taklit ettiklerinden fakat bu çalışmada, bozulma belirleme çözücüsü (DSS) olarak adlandırılan modelleme çerçevesinin yenilikçi bir özellik olduğu ve uçak çizelgesindeki bozulmaları dinamik bir şekilde ele aldığından bahsetmişlerdir. Geliştirdikleri

çerçeve, verimli bir uçak seçimi algoritmasının ve paralel uçak özel zaman-mekan ağlarına dayanan bir doğrusal programlama modelinin kombine kullanımı üzerine kuruludur. ASRP'yi çözmek için kullanılan optimizasyon modelinin amacının, operasyonel, yolcu gecikme ve iptal maliyetleri de dahil olmak üzere maliyetleri en aza indirmek olduğunu vurgulamışlardır. Ve bu durum da kullanılan karar değişkenleri, uçuş iptali, uçuşların ertelenmesi ve uçakların uçuşlar arasında değiştirilmesi ile ilgilidir. Son olarak da elde edilen sonuçların; geleneksel statik yaklaşımın pratik meydan okumayı ihmal ederek ve bozulma maliyetlerini olduğundan daha az tahmin ederek güvenilmez çözümlere yol açabileceğinden ve çalışmada önerilen dinamik DSS çerçevesinin, gerçek zamanlı operasyonlar için uygun bir zaman penceresi içinde gerçek uçak çizelgesi bozulma problemlerini çözebileceğinden bahsetmişlerdir.

Samà, D'Ariano, Pacciarelli (2017), çalışmalarında trafik tıkanıklığı durumunda meşgul havaalanı terminal kontrol alanında kalkış ve iniş işlemlerinin gerçek zamanlı optimizasyonunu ele almışlardır. Terminal alanlarının, özellikle yeni altyapı inşa etme imkânının sınırlı olduğu büyük Avrupa havalimanlarında tüm hava trafik kontrol sisteminin darboğazı haline geldiğine değinmişlerdir. Ayrıca, karmaşık tamsayılı doğrusal programlama formülasyonları, yüksek performans modelleme hassaslığı ile emniyet kısıtlamalarını göz önünde bulundurularak, pratik ilgi çeken çeşitli performans göstergeleri arasındaki dengeyi araştırmak üzere önerilmiştir. Milano Malpensa ve Roma Fiumicino'da iki önemli havalimanı için çeşitli rasgele iniş ve kalkış uçaklarının bozukluklarını simüle ederek denemeler gerçekleştirilmiştir. Pratik boyutlu örneklerin, ticari bir çözücü kullanılarak en iyi duruma getirildiğinden bahsedilmiştir. Eniyileştirilmiş çözümler, yaygın olarak kullanılan bir zamanlama kuralı ile de karşılaştırılmıştır. Kapsamlı bir hesaplamalı analiz, çeşitli göstergeler arasında iyi bir uzlaşı bulabilen çözümlerin seçimini mümkün kıldığı belirtilmiştir.

Mukherjee , Hansen (2009) hem havaalanının kendisi hem de yaklaşma rotaları olumsuz havaya maruz kaldıklarında bir havalimanına gelen hava trafiğini yönetmek için bir stokastik tamsayı programlama modeli sunmuşlardır. Modelde, yerden erteleme kararları statik, yeniden yönlendirmede olanlar dinamiktir. Modeldeki karar değişkenleri, çeşitli kapasite kısıtlı kaynaklara ulaşmayı planlayan toplam uçuş sayısıdır. Model, varış saatlerini bireysel uçuşlara doğrudan atamadığı için, dinamik yeniden yönlendirme modelinden gelen çözümler, doğrudan yönlendirmeleri ve serbest bırakma sürelerini bireysel uçuşlara ya da onları işleten

havayollarına atayan bazı kaynak ayırma algoritmasına doğrudan eklenebilir olduğundan bahstmişlerdir. Kötü hava koşulları engellediğinde veya terminal yaklaşım güzergahlarının kapasitesi ciddi ölçüde sınırladığında, uçuşları diğer yaklaşımlara göre yeniden yönlendirmek, yüksek yerdeki gecikmelerin hafifletilmesi ile önemli yararlar sağlamaktadır. Gerçekleştirdikleri deneylerin sonucuna göre, yeniden yönlendirme kararlarının dinamik bir şekilde, statik yeniden yönlendirme ile karşılaştırıldığında %10-15 gecikme maliyeti azaltımı ve "saf" yer tutma stratejisine kıyasla (Yeniden yönlendirmenin olmaması gibi..) karşılaştırıldığında yaklaşık %50 daha az gecikme maliyeti sağladığını göstermişlerdir. Statik yeniden yönlendirmeden farklı olarak, dinamik yeniden yönlendirme yeteneği, gerçekleştirilen hava koşullarına daha iyi uyan yeniden yönlendirme kararları vermeye neden olduğunu vurgulamışlardır. Düşük toplam beklenen gecikme maliyeti, alternatif rotalar arasındaki ayrışma noktasına ulaşıncaya kadar uçuşlar için yeniden yönlendirme kararlarını erteleyerek ve bu kararları alırken güncellenmiş hava durumu bilgilerini elde ederek sağlandığından bahsetmişlerdir.

Ahmed, Mansour, Haouari (2016), makalelerinde dayanıklı hava aracı çizelgeleri yapılması üzerine odaklanmışlardır. Uçakların çizelgelenmesi konusu içersinde dayanıklık kavramını, uçakların yeniden yönlendirilmesi ve uçuş kalkış zamanlarının yeniden ölçülmesi yoluyla uçak ve yolcuların bağlantılarına etkili tampon sürelerini atayarak programa dahil etmişlerdir. Ayıca çalışmada; PSO algoritması, iki aşamalı MIQP modelleri, Monte Carlo simülasyonu ve evrimsel bir algoritmayı birleştiren iki seviyeli bir yaklaşım geliştirmişlerdir. İlk aşamada, iki karışık-tamsayı kuadratik programlama modelini, arama sürecini yönlendirmek için bir Parçacık Sürüsü Optimizasyonu (PSO) prosedürüyle birlikte bir simülasyon prosedürünü çağırarak sağlamlığı denenen uygulanabilir deneme çözümleri oluşturmak için tekrarlı olarak çözmüşlerdir. İkinci aşamada, evrimsel bir algoritma, birinci aşamada üretilen çözümü tekrar tekrar geliştirmek için kullanılmıştır. Gerçek very setleri ile çalışarak, 1278 uçuşla kapsamlı hesaplama deneylerinin sonuçlarını sunmuşlardır.

Hancerliogullari, Rabadi, Al-Salem, Kharbeche (2013), makalelerinde, uçak iniş ve kalkış operasyonlarında toplam ağırlıklı gecikmesini en aza indirgemek amacıyla, karışık modlu operasyonlar altında, birden fazla pistteki Uçak Sıralaması Problemini (ASP) incelemişlerdir. Problem, NP-Zor olduğundan makul hesaplama zamanlarında çözümler elde etmek için aç

gözlü sezgisel ve metasezgisel algoritma uygulanmışlardır. Algoritmaların çözümleri optimal çözümlerle karşılaştırılmış ve performansları çözüm kalitesi ve işlemci süresi açısından değerlendirilmiştir.

Shenpeng, Cao, Zhang (2011), makalelerinde tek pist Uçak Çizelgeleme Problemini ele alınmıştır. Önerilen yöntem Hücresel otomasyon Yöntemi (CA) ve Genetik Algoritma (GA)'dan ve aynı zamanda Uçak iniş çizelgeleme (ALS) problemini kısa sürede çözebilen deterministik bir algoritmadan oluşmaktadır. Yapılan deneylere göre geliştirdikleri yöntemin geleneksel optimizasyon yöntemlerine kıyasla gerçek zamanlı ALS problemi için daha hızlı ve uygun olduğundan bahsetmişlerdir.

Bencheikh, Boukachour, Alaoui (2011) , Makalelerinde, çoklu pist durumunda uçak iniş problemi (ALP) incelenmişlerdir. Bir öncelik kuralı tarafından belirlenen uçak kalkış sürelerini tek bir pistte planlamak için yeni bir buluşsal yöntem önermişlerdir. Sezgisel testler için birkaç öncelik kuralını karşılaştırmışlardır. Öncelikle, problemin doğrusal ve doğrusal olmayan bir amaç fonksiyonu ile matematiksel bir formülasyonunu sunmuşlardır. Daha sonar, tüm verilerin önceden bilindiği ve tek bir pistte uçak iniş planlaması için yeni bir sezgiselliğin sunulduğu sorunun statik durumunu ele almışlardır, bu buluşsal yöntem, birden fazla pist durumunu çözmek için bir karınca koloni algoritmasına dahil edilmiştir. Doğrusal olarak nesnel durumda, algoritmanın toplam örnek sayısının %80'inde optimal çözümlerle çakışan çözümler sağlarken, kalan örneklerin %20'si için en iyi çözümlerden ortalama %5 sapma gösterdiğinden bahsetmişlerdir.

Beasley, Krishnamoorthy, Sharaiha, Abramson (2004), yer değiştirme problemi için genel bir karar problemini ele almışlardır. Yer değiştirme problemi, bir karar sırası oluşturulması gerektiğinde ortaya çıkmakta ve yapılması gereken her yeni kararın, yapılan önceki karara geri dönmesini sağlamaktadır. Yer değiştirme sorununun ortaya çıktığı bir durum, bir havalimanında uçak inişlerinin dinamik olarak planlanması ile ilgilidir. Uçakların ve iniş pistlerinin iniş saatlerine ilişkin kararlar, zaman geçtikçe ve operasyonel çevre değiştikçe dinamik bir biçimde alınmalıdır. Çalışmada, yer değiştirme sorununun dinamik uçak iniş problemine uygulanması gösterilmektedir. Hesaplama sonuçları, 500 uçak ve beş pistten oluşan bir dizi kamuya açık test problemi için sunulmuştur.

Beasley, Sonander, Havelock (2001), İngiltere'nin Ulusal Hava Trafik Hizmetleri tarafından Londra Heathrow'da pist kullanımını iyileştirmeye yönelik yapılan bir soruşturma üzerine yapılan araştırma çalışmasında, karaya çıkmayı bekleyen uçakların zamanlamasını iyileştirmek için bir algoritma geliştirmeyi amaçlamışlardır. Makalede, Geliştirilen sezgisel algoritma (nüfus buluşsal yöntemi) tartışılmış ve sonuçlar Londra Heathrow'daki uçak inişleriyle ilgili fiili operasyonel veriler kullanılarak sunulmuştur. Çalışma sonucuna göre, benzer durumlarda hava trafik kontrol kararlarında algoritmanın, dikkate alınan tüm uçakların iniş süresinin kısaltılması açısından % 2 ± 5 oranında iyileşme göstereceğinden bahsetmişlerdir.

Ernst, Krishnamoorthy, Storer (1998) Uçak inişlerinin bir veya birden fazla pistte planlanması sorunuyla ilgilenmişlerdir. Amaç, bir veya daha fazla sayıda pistte bir dizi uçağın aralarındaki olması gereken ayrım kriterleri sağlanarak en iyi şekilde karaya indirmektir. Çalışmada, kısmi sıralama bilgilerine dayanarak iniş sürelerini çok hızlı bir şekilde değerlendiren özel bir simpleks algoritması sunulmuştur. Yöntem, daha sonra bir problem alanı arama sezgisel yönteminde ve aynı zamanda hem tek, hem de çok pistli problemler için bir dal ve sınır yöntemi içerisinde kullanılır. Algoritmanın etkinliği, literatürden elde edilen bazı standart test problemleri kullanılarak test edilmiştir.

Diallo, Ndiaye, Seck (2012), Havaalanında tek pist olması durumunda statik uçak iniş probleminin formülasyonu ve simülasyonları üzerinde çalışmışlardır. Bu çalışmada da, diğer gelen uçakların çizelgelenmesi problemlerinde olduğu gibi amaç, hedeflenen iniş süresinden sapmaların maliyetini en aza indirgemektir. Problemi çözerken karma ve 0-1 tamsayıya dayalı doğrusal programlama yönteminden faydalanmışlardır. Aynı zamanda, pratik karar desteği için gerçek very setlerini kullanark simülsayon tekniklerinin Afrika ve Madagaskar'daki Hava Seyrüsefer Güvenliği Ajansına (ASECNA) Léopol Sédar Senghor (LSS) uluslararası havalimanında gerçek dünyada nasıl başarılı bir şekilde uygulandığını göstermişlerdir. (LSS) havalimanının gerçek veri kümelerine dayanan sayısal deneyler ve karşılaştırmalar sınmuşlardır. Kullandıkları yönteme göre pist sisteminin verimini maksimize edebildiklerinden bahsetmişlerdir.

García, Quintero (2011) , hava taşıtı operasyonlarındaki maliyetleri ve yolcu gecikme sürelerini azaltmak için, kalabalık havalimanlarındaki kalkış ve inişlere katılmak için bir strateji

oluşturma üzerine çalışmışlardır. Önerdikleri stratejiyle, uçağa katılmak için ilk önce gelen ilk hizmet esasına dayanan geleneksel kural, işletme maliyetlerini ve yolcu gecikmelerini azaltan bir sıralama ile değiştirilmiştir. Her bir uçağa verilen dikkat sırası, olası tüm dizilerin numaralandırılmasını gerektirmeyen sezgisel bir algoritma kullanılarak elde edilmiştir. Bu stratejiyi kullanırken elde edilen sonuçların, işletme maliyetlerinde %47.6'ya, yolcu gecikmelerinde %73.2'ye kadar önemli oranlarda düşüşleri gerçekleştirebileceğini göstermişlerdir.

Lee, Balakrishan(2008), ayrılma ölçütleri, varış zamanı pencereleri, FCFS diziliminden sapma sınırları ve öncelik koşulları gibi kısıtlamaların bulunduğu bir varış programının toplam gecikme maliyetini en aza indirgemek için bir algoritma üzerinde çalımışlardır. Bu algoritma ortalama gecikme ve çıktı arasındaki dengeyi değerlendirmek için kullanılmaktadır. Önerilen yaklaşım, geçiş noktası maksimize etme ve gecikmeleri en aza indirgeme zamanlarının karşılaştırılması yoluyla terminal alan çizelgelerindeki dengeleri değerlendirmeyi mümkün kıldığından bahsetmişlerdir. Uçakların ETA yani kendileri için planalanan sürelerinden önce inmeleriyle ilgili kanunları araştırdıkları sırada, algoritmayı aynı zamanda toplam yakıt maliyetlerini azaltmak için de kullanmışlardır. Minimum yakıt maliyeti programlarını hem gecikme maliyetini hem de hızlandırmanın ekstra yakıt maliyetini göz önüne alarak belirlemişlerdir. Yaptıkları analize göre, en pratik senaryolarda 3 dakikaya kadar bir zaman ilerlemesi yapılabileceğinden bahsetmişlerdir.

Litertaür çalışmasına göre uçak çizelgeleme problemleri içinde tezin doğrudan konusuna benzer olanlar benzerliklerin ve farklılarındaha detaylı incelenmesi açısından içeriklerine göre gruplanarak aşağıdaki tablolar şeklinde özetlenmiştir.

Tablo 2.3: Literatür Özet Tablosu

Kaynak	Çalışma Adı	Amaç	Amaç Fonk	Öngörülemeyen Durumlar (Disruptions)	Çözüm Metodu	Sayısal Veri	Açıklama (Remarks)	Pist Sayısı
Ernst, Krishnamoorthy, Storer (1999)	Heuristic and Exact Algorithms for Scheduling Aircraft Landings	Bir veya daha fazla sayıda pistte bir dizi uçağın aralarındaki olması gereken ayrım kriterleri sağlanarak en iyi şekilde karaya indirmek	Toplam gecikmelerin minimizasyonu		Dal sınır Algoritması, Problem alanı arama sezgisel yöntemi, Simpleks yötemine dayalı	Literatürdeki test problemleriyle test edilmiştir		Tekli ve Çoklu Pist
Beasley, Krishmanoorthy, Saharaiha, Abramson (2000)	Scheduling Aircraft Landings-The Static Case	Toplam uçak maliyetlerini en aza indirgeme	Hedef zamanlardan sapma maliyetinin minimizasyonu	Problemin büyüklüğü	Karışık Tamsayılı Programlama, Dal Sınır Algoritması,Genetik Algoritma	OR Kütüphanesi (Beasley 1990)	Genel Olarak Karışık Tam Sayılı Programlama (MIP) kullanılmıştır.	çoklu pist
Beasley, Sonander, Havelock (2001)	Scheduling aircraft landings at London Heathrow using a population heuristic	Havalanına inmeyi bekleyen uçakların zamanlamasını iyileştirmek için bir algoritma geliştirme	Uygunluk fonksiyonunun (Fitness Function)maksimizasyonu	Bağlantılı uçuşları kaçıran yolcular, aynı uçak ya da mürettebatın sonraki uçuşa gecikmesi	Populasyon Sezgiseli	Heathrow Havalimanı gerçek veri setleri	Modeli baz alınan ve en çok referans gösterilen makalelerden biri	çoklu pist
Beasley, Krishnamoorth, Sharaiha, Abramson (2004)	Displacement problem and dynamically scheduling aircraft landings	Zorunlu durumlarda uçakların en az maliyetle yer değişimini sağlamak	Yer değiştirme maliyetinin minimizasyonu		Sezgisel algoritma	500 Uçak		Çoklu Pist
Pinol, Beasley (2006)	Scatter search and bionomic algorithms for the aircraft landing problem	Pistleri etkin şekilde kullanarak hedef zamanlardan sapan uçak iniş maliyetlerini en aza indirgeme	Hedef zamanlardan sapma maliyetinin minimizasyonu		Karışık Tamsayılı Programlama, 2 populasyon tabanlı sezgisel	OR Kütüphanesi 50 uçak ve 4 pist (Beasley 1990)		
Bäuerle, Engelhardt-Funke, Kolonko (2007)	On the waiting time of arriving aircrafts and the capacity of airports with one or two runways	Bir havalimanına gelen uçakların kuyruklama sürecini ve havalimanı kapasitesine etkilerini inceleme	-	Hava koşulları, teknik sebepler, uçakların geç gelmesi	Olasılık, simülasyon	Bir havalimanında gerçek veri setleri kullanmışlardır		Tekli ve Çoklu Pist

Tablo 2.4: Literatür Özet Tablosu (devam)

Kaynak	Çalışma Adı	Amaç	Amaç Fonk	Öngörülemeyen Durumlar (Disruptions)	Çözüm Metodu	Sayısal Veri	Açıklama (Remarks)	Pist Sayısı
Soomer, Franx (2008)	Scheduling aircraft landings using airlines' preferences	Maliyetleri de dikkate alarak tercih edilen varış zamanından sapmayı en aza indirme	Ölçeklendirilmiş maliyetlerin toplamının maliyetlerin minimizasyonu	Uçkaların geç kalkışı	Yerel arama sezgiseli, tamsayılı programlama, simülasyon	Avrupadaki büyük bir havalimanının gerçek veri setleri kullanılmıştır		Tek Pist
Lee, Balakrishan (2008)	Fuel Cost, Delay and Throughput Tradeoffs in Runway Scheduling	Piste gelen uçakların akışının arttırılması ve uçak iniş maliyetinin azaltılması	Uçakların toplam inme maliyetinin minimizasyonu		Dinamik Programlama	30 Uçak		Tek Pist
Balakrishnan, Chandran (2010)	Algorithms for scheduling runway operations under constrained position shifting	Listelenen tüm kısıtlamaları yerine getirirken Cmax.ı en aza indirgeyen bir zaman çizelgesi oluşturma	Toplam maliyet minimizasyonu	Uçakların erken gelmesi ve erken kalkışı, hava trafik yoğunluğu	Dynamic programming, simülasyon	100 rastgele üretilmiş örneklem ve her veri için 10,20,30,40,50 uaçak		Çoklu Pist
Yu, Cao, Zhang (2011)	A real-time schedule method for aircraft landing scheduling problem based on cellular automation	Uçak iniş problemlerini hızlı çözmek	Toplam gecikmenin minimizasyonu	Pist Sayısı, uçak sayısı,	Hücresel otomasyon modeli, Yerel Arama Yöntemi, Roisimli deterministik operatör,	13 veri seti, OR Kütüphanesi		Tek Pist
Bencheikh , Boukachour,Alaoui (2011)	Improved Ant Colony Algorithm to Solve the Aircraft Landing Problem	Toplam ceza maliyetlerini minimize etmek	Tüm uçakların iniş saatleri ile hedef iniş saatleri arasındaki sapmanın ceza maliyetinin minimizasyonu		Karınca Kolonisi	Beasley (1990) 10-50 Uçak ve 1-5 pist	Uçaklar arası simetrik ayırım süreleri kabulu matrisi	
Shenpeng, Cao, Zhang (2011)	A Real-time Schedule Method for Aircraft Landing Scheduling Problem Based on Cellular Automaton	Uçakların gecikmesinin ve toplam maliyetlerin minimize edilmesi	Toplam maliyetin minimizasyonu		Hücresel otomasyon ve Genetik Algoritma	13 veri seti, OR Kütüphanesi		Çoklu Pist

Tablo 2.5: Literatür Özet Tablosu (devam)

Kaynak	Çalışma Adı	Amaç	Amaç Fonk	Öngörülemeyen Durumlar (Disruptions)	Çözüm Metodu	Sayısal Veri	Açıklama (Remarks)	Pist Sayısı
Diallo, Ndiaye, Seck (2012)	Scheduling Aircraft Landings at LSS Airport	Hedeflenen zamanlardan sapma maliyetini en aza indirme	Hedeflenen zamanlardan sapma maliyetinin minimizasyonu		Karışık Tam Tamsayılı Programlama, Simülasyon	2 veri seti, 16-36 Uçak		Tek Pist
Salehipour, Modarres ve Naeni (2013)	n efficient hybrid meta-heuristic for aircraft landing problem	Amaç, uçakların iniş saatinin toplam hedef zamanından sapmasını en aza indirmektir (hem erkencilik hem de gecikme) böylece uygun çizelgeleme yaparak uçakların iniş maliyetini en aza indirmek	Hedef süreden iniş sapmasının toplam maliyetinin minimizasyonu		Benzetim tavlamasına dayalı melez meta-sezgisel, karma tamsaysılı hedef programlama	100 -500 uçak arası deneme yapılmış ve 100 uçağa kadar optimum sonuç elde edilmiş (Beasley 1990)		çoklu pist
Hancerliogullari, Rabadi, Al-Salem, Kharbeche (2013),	Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem	uçak iniş ve kalkış operasyonlarında toplam ağırlıklı gecikmesini en aza indirgemek	toplam ağırlığa bağlı gecikmenin minimizasyonu	Problem büyüklüğünün artması	Aç gözlü sezgisel ve meta sezgisel algoritma, Benzetim Tavlaması	15-25Uçak Seti, 55 Proplem Örneği (Ghoniem Veri Seti)		çoklu pist
Samà, D'Ariano, Pacciarelli (2014)	Optimal aircraft scheduling and routing at a terminal control area during disturbances	Hava trafiğinin yoğun olduğu terminal kontrol alanlarında (TCA), optimizasyon tabanlı çözücüler tarafından elde edilebilecek potansiyel gecikmenin azaltılması	maksimum ardışık gecikmenin minimizayonu	Hava koşulları, Uçak sayısı	Dal sınır algoritması, tabu arama,karışık tamsayılı doğrusal programlama	60 ATC-TCA geciken uçak		Tek Pist
Vadlamani, Hosseini (2014)	A novel heuristic approach for solving aircraft landing problem with single runway	En uygun şekilde uçakların inişlerini çizelgeleme	Uçakların geç ve erken gelmesinden kaynaklana ortalama gecikmenin minimizasyonu		Uyarlanabilir Geniş Komşuluk Araması (ALNS), CPLEX	150 Uçak, OR Kütüphanesi		Tek Pist

Tablo 2.6: Literatür Özet Tablosu (devam)

Kaynak	Çalışma Adı	Amaç	Amaç Fonk	Öngörülemeyen Durumlar (Disruptions)	Çözüm Metodu	Sayısal Veri	Açıklama (Remarks)	Pist Sayısı
Farhadi, Ghoniem, Al-Salem (2014)	Runway capacity management – An empirical study with application to Doha International Airport	Hava sahasına giren uçakların hazır oldukları zamanda en kısa süre piste inmeleri ve sapmaların az olması	Toplam Yakıt Maliyetinin Minimizasyonu		Karışık Tam Sayılı Programlama	Doha Havalimanında ferçek verilerle uygulama, 20-40 Uçak		Tek Pist
Sabar, Kendall (2015)	An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem	Her bir uçağın tercih edilen hedef zamanından sapmasını en aza indirme	Beklenen iniş saatinden iniş sapmalarının toplam cezasını en aza indirme	Uçak sayısı, problem büyüklüğü	Yinelemelei Yerel Arama Serzgiseli	13 ALP Örneği, 10-500 Uçak, OR Kütüphanesi	Küçük veri setleri için zaman endeksli MIP hızlı çalışıyor	Çoklu Pist
Ghoniem, Farhadi, Reihaneh (2015)	An accelerated branch-and-price algorithm for multiple runway air craft sequencing problems	Zaman penceresi kısıtlaması ve dal-fiyat algoritmasıyla çoklu pistlerde uçakların sıralanmasının araştırılması	Toplam ağırlıklı başlangıç zamanlarını en aza indirme	uçak sayısı, uçaklar arası ayırım süresi, pist sayısı	Dal-fiyat (B & P) algoritması	15-50 Uçak	Birbirinden bağımsız pistler için etkili ve kesin bir yöntem	Çoklu pist
Lieder, Briskorn, Stolletz (2015)	A dynamic programming approach for the aircraft landing problem with air craft classes	Uçak inişlerinin toplam gecikmesini veya ilgili maliyeti en aza indirme	Uçakların beklenen zamandan geç inmesiyle oluşan uçak inişlerinin toplam gecikme maliyetini en aza indirme	sınırlı zaman pencerelri, havaalanı pist sistemleri, uçak sayısı, uçaklar arası ayırım zamanları	Dal-Paha Algoritması, Karışık Tamsayılı Programlama, dinamik bir programlama	50-100 uçak Bianco ve diğerleri (1999)	Sadece takip eden(ardıl) uçaklar için ayrım süresi dikkate alınmıştor	Çoklu Pist
Faye (2015)	Solving the Aircraft Landing Problem with time discretization approach	Uçakları mümkün olduğunca tercih edilen iniş zamanına yakın hatta daha öncesinde yani mümkün olduğunca erken karaya indirmek	Gelen uçakları mümkün olduğunca erken karaya indirirken gerekli tüm maliyetleri minimize etmek	Uçak sayısı, uçaklar arası ayırım süresi	Uçaklar arası ayırma zamanı matrisinin yaklaşık ve tam zaman ayrımınına dayalı yöntem ve sezgisel algoritma	500 Uçak Beasley (1990)	Zaman indeksli MIP küçük ölçekli problemler için hızlı çalışıyor	Çoklu Pist

Tablo 2.7: Literatür Özet Tablosu (devam)

Kaynak	Çalışma Adı	Amaç	Amaç Fonk	Öngörülemeyen Durumlar (Disruptions)	Çözüm Metodu	Sayısal Veri	Açıklama (Remarks)	Pist Sayısı
Girish (2016)	An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem	Uçakların iniş saatlerinin ilgili hedef iniş saatlerinden sapması nedeniyle toplam ceza maliyetini en aza indirgeme	Toplam ceza maliyetinin minimizasyonu	Statik durum, dinamik durum, uçak sayısı	Döner ufuk çerçevesinde hibrid bir parçacık sürü optimizasyon yerel arama algoritması (RH-HPSO-LS)	500 Uçak		Çoklu Pist
Samà, D'Ariano, Corman, Pacciarelli (2016)	Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas	Terminal Kontrol Alanı (ATC-TCA) probleminde Hava Trafik Kontrolü için amaç fonksiyonların ve hesaplama süresinin değeri açısından, çözümlerin kalitesini büyük ölçüde arttırma	Hedeflenen iniş ve kalkış saatlerinden maksimum pozitif sapmanın en aza indirgenmesi, Cmax.ın minimizasyonu	Birden fazla uçakta gecikmeler, geçici olarak bozulmuş pist, hava trafik tahmininin zaman ufku,	Değişken komşuluk arama, tabu arama ve hibrid şemaları	60 ATC-TCA geciken uçak, Ortalama 20 örneklem Samà ve diğerleri (2014),		Çoklu Pist
Heidt, Helmke, Kapolke, Liers, Martin (2016)	Robust runway scheduling under uncertain conditions	Planlanan zamanlardan sapmaların toplamını en aza indirme	Maliyeti artırmayan tüm çözümler arasında sağlamlığı en üst düzeye çıkarma	Hava trafik yoğunluğu	simülasyon	50 Uçak		Tek Pist
Bencheikh, Boukachour, Alaoui (2016)	A memetic algorithm to solve the dynamic multiple runway aircraft landing problem	Havaalanına geç gelen tüm uçaklardan kaynaklanan toplam ceza maliyetini düşürme	Toplam ceza maliyetini ve geçerli çözümün önceki durumdan toplam harekete karşılık gelen yer değiştirme fonksiyonunu en aza indirme	Pist Sayısı, uçak sayısı, inen uçaklar arasındaki ayırım zamanları,	Karınca koloni algoritmasını ve yerel gelişme sezgisel algoritmasını birleştiren bir memetik algoritma	10-50 Uçak		Tek ve Çoklu Pist
Wu, Sun, Qu (2016)	A sequencing model for a team of aircraft landing on the carrier	Maliyet fonksiyonunu ve iniş zamanı tüketimini en aza indirgeme	Toplam maliyet minimizasyonu	Yakıt sıkıntısı, uçak gövde hasar derecesi	Karınca kolonisi yöntemine dayalı dinamik bir sıralama algoritması	10-20 Uçak		Tek Pist
Lieder, Stolletz (2016)	Scheduling aircraft take-offs and landings on interdependent and heterogeneous runways	Uçak kalkış ve inişleri,bir piste atama ve sıraya bağlılık gereksinimlerini karşılayıp, gecikmelerden kaynaklanan maliyetleri en aza indirgeme	Gecikmiş işlemler için toplam ceza maliyetini en aza indirme	Zaman penceresi, pist sayısı	karışık tamsayılı programlama, dinamik programlama, dönme planlama ufku sezgiseli (RPH)	30-62 Uçak	Çoklu Kriter Amaçları, Kesin yaklaşım,	Çoklu Pist

Tablo 2.8: Literatür Özet Tablosu (devam)

Kaynak	Çalışma Adı	Amaç	Amaç Fonk	Öngörülemeyen Durumlar (Disruptions)	Çözüm Metodu	Sayısal Veri	Açıklama (Remarks)	Pist Sayısı
Xu (2017)	An efficient Ant Colony algorithm based on wake-vortex modeling method for aircraft scheduling problem	Yayılım zamanını (Cmax) minimuma indirme	Toplam ceza maliyetlerinin minimizasyonu	Uçak sayısı, uçaklar arası ayırım süresi, pist sayısı	kanat iz türbülansı (wakevortex) modelleme (WVM) yöntemine dayanan bir Karınca Kolonisi (AC) algoritması	50 Uçak		Çoklu Pist
Benell, Mesgarpour, Potts (2017)	Dynamic scheduling of aircraft landings	Pist akışı maksimizasyonu, gecikme ve erken gelme minimizasyonu, yakıt maliyeti minimizasyonu ile ilgili gereksinimleri karşılayan bir model geliştirmek ve ATC'lere fayda sağlayacak çözümler üreten dinamik / online çizelgeleme algoritması tasarlamak	Piste gelen uçakların akışının maksimizasyonu, gecikme ve erken gelme sürelerini minimizasyonu ve uçak manevralarında ortaya çıkan yakıt maliyetiminimizasyonu	Uçakların geç gelmesi, Hava trafiği	Dinamik programlama, benzetim tavlaması, döner ufuk yaklaşımı	Heathrow Havalimanı gerçek veri setleri	Çok amaçlı amaç fonksiyonu ve dinamik çizelgeleme	Çoklu Pist
Ng, Lee, Chan, Qin (2017)	Robust aircraft sequencing and scheduling problem with arrival/departure delay using the min-max regret approach	min-max pişmanlık yaklaşımını kullanarak; tüm en kötü senaryolarda optimallik ve maksimum yayılım zamanı arasındaki sapmayı en aza indirmek	Tüm kötü senaryoları dikkate alarak maksimum pişmanlık değerinin minimizasyınu	Uygun olmayan pist kullanımına bağlı olarak meydana gelen kazalar, uçakların gecikmesi, terminaldeki hava koşulları, Rüzgar hızı, hava trafiğinde öngörülemeyen durumlar,	Karma tip (gelen-giden uçaklar) paralel pist operasyonları için min-max pişmanlık yaklaşımı kullanılmıştır. Modelde, sürü zekasına dayalı algoritma kullanılarak geliş ve kalkış gecikmeleri dikkate alınmıştır.		Yapay Arı Kolonisi ve min. Max. Pişmalık yaklaşımı	çoklu pist
Ahmed, Alam, Barlow (2018)	A Cooperative Co-Evolutionary Optimisation Model for Best-Fit Aircraft Sequence and Feasible Runway Configuration in a Multi-Runway Airport	Pist kullanım kapasitesini en üst seviyeye çıkarabilen, uygun bir pist konfigürasyonuyla birleştirilmiş bir uçak sıralamasını elde etmek için bir model geliştirmektir.	En uygun pist konfigürasyonunu en uygun şekilde seçerken, ardışık uçaklar arasındaki boşluğu en aza indirerek optimum bir sıra elde etmek (Pist kullanım oranının maksimizasyonu ve kalkış varış saatlari arası toplam sürenin minimizasyonu)		Genetik algoritmaya dayanan, popülasyona dayalı metasezgisel yaklaşım kullanılmıştır. Ayrıca C# programı kullanılmıştır		Geç- erken kalkış ve gelişler için ceza maliyetleri uygulanmamış	Çoklu Pist
Prakash, Piplani, Desai (2018)	An optimal data-splitting algorithm for aircraft scheduling on a single runway to maximize throughput	Pist kullanım oranını arttırma	Toplam yayılım zamanının minimizasyonu		Veri Bölme Algoritması, 0-1 Tam sayılı programlama, dallandırma kesme yaklaşımı			Tek Pist

3. MALZEME VE YÖNTEM

Bu bölümde, uçak çizelgeleme problemleri (UÇP) için literatürdeki çalışmalara göre seçilen 4 metasezgisel algoritmanın (Tabu Arama, Tavlama Benzetimi, Genetik Algoritma ve Karınca Kolonisi Algoritması) geliştirilen özgün sezgisel yöntemlerle performanslarının arttırılarak UÇP problemlemine uyarlanışından bahsedilecektir. Geliştirilen sezgisel yöntem OR kütüphanesindeki (Beasley, 2000) veri setleri kullanılarak 10 uçak için detaylıca anlatılacaktır. Ayrıca UÇP için geliştirilen ve hem gelen hem de kalkan uçaklar için aynı anda kullanılabilen matematiksel model, algoritma akış şeması, sözde kod gibi konular da detaylı bir şekilde anlatılacaktır.

3.1. MATEMATİKSEL MODEL

Matematiksel model oluşturulurken Beasley vd. (2000) makalesinde kullandıkları model baz alınarak geliştirilmiş ve statik durum ve gelen uçaklar ve pist sayısının belirli olmadığı durumlar için kullanılan model uçakların büyüklük katsayıları ve kalkan uçaklar ve gecikmeleri de dahil edilerek dinamik ve statik problemlere uyarlanabilir ve tek pist için geçerli olacak şekilde oluşturulmuştur. Geliştirilen matematiksel model, optimizasyon ve modelleme için kullanılan GAMS programında test edilerek doğruluğu teyit edilmiştir.

İndeks ve Notasyonlar

P: Uçak sayısı

E_{L_i}: i uçağı için en erken iniş zamanı (i = 1, …, P)

L_{L_i}: i uçağı için en geç iniş zamanı (i = 1, …, P)

E_{D_i}: i uçağı için en erken kalkış zamanı (i = 1, …, P)

L_{D_i}: i uçağı için en geç kalkış zamanı (i = 1, …, P)

T_{L_i}: i uçağı için hedef iniş zamanı (i = 1, …, P)

T_{D_i}: i uçağı için hedef kalkış zamanı (i = 1, …, P)

S_{ij}: arka arkaya inen i ve j uçakları arasında gerekli olan ayrım zamanı (i, j'den önce inmektedir), i = 1, …, P; j = 1, …, P; i ≠ j

Karar Değişkenleri

x_i: i uçağı için gerçekleşen iniş zamanı (i = 1, …, P)

y_i: i uçağı için gerçekleşen kalkış zamanı (i = 1, …, P)

w_i: i uçağının büyüklüğüne ve operasyon tipine bağlı değişen katsayısı (i = 1, …, P)

a_i: i uçağının gecikme süresi (i uçağı için hedeflenen iniş zamanından sonra geçen süre; $a_i = \max(0, x_i - T_{L_i})$)

b_i: i uçağının havaalanına erken gelme süresi (i uçağı için hedeflenen iniş zamanına kalan süre; $b_i = \max(0, T_{L_i} - x_i)$)

c_i: i uçağının geç kalkış süresi (i uçağı için hedeflenen kalkış zamanından sonra geçen süre; $c_i = \max(0, y_i - T_{D_i})$

d_i^+: i uçağı için geç iniş maliyeti

d_i^-: i uçağı için erken iniş maliyeti

f_i^+: i uçağı için geç kalkış maliyeti

g_{ij}: i uçağı j uçağından önce inerse 1 değerini alır, aksi durumda 0 değerini alır

h_{ij}: i uçağı j uçağından sonra kalkarsa 1 değerini alır, aksi durumda 0 değerini alır

Matematik Formülasyonu

$$Min\ z = \sum_{i=1}^{P} w_i(a_i.d_i^+ + b_i.d_i^- + c_i.f_i^+) \quad (3.1)$$

$$E_{L_i} \leq x_i \leq L_{L_i} \quad (3.2)$$

$$E_{D_i} \leq y_i \leq L_{D_i} \quad (3.3)$$

$$x_i - T_{L_i} = a_i - b_i \quad (3.4)$$

$$y_i - T_{D_i} = c_i \quad (3.5)$$

$$T_{L_i} - x_i \leq a_i \leq T_{L_i} - E_{L_i} \quad (3.6)$$

$$x_i - T_{L_i} \leq b_i \leq L_{L_i} - T_{L_i} \quad (3.7)$$

$$T_{D_i} - y_i \leq c_i \leq T_{D_i} - E_{D_i} \quad (3.8)$$

$$x_j - x_i \geq S_{ij} \qquad \text{i, j = 1, …, P; i} \neq \text{j} \quad (3.9)$$

$$y_j - y_i \geq S_{ij} \qquad \text{i, j = 1, …, P; i} \neq \text{j} \quad (3.10)$$

$$x_j - y_i \geq S_{ij} \qquad \text{i, j = 1, …, P; i} \neq \text{j} \quad (3.11)$$

$$y_j - x_i \geq S_{ij} \qquad \text{i, j = 1, …, P; i} \neq \text{j} \quad (3.12)$$

$$g_{ij} + g_{ji} = 1 \quad (3.13)$$

$$h_{ij} + h_{ji} = 1 \quad (3.14)$$

$$g_{ij} + h_{ij} = 1 \quad (3.15)$$

$$g_{ij}, h_{ij} \in \{0,1\} \qquad \text{i, j = 1, …, P; i} \neq \text{j} \quad (3.16)$$

$$x_i, y_i, x_j, y_j, a_i, b_i, c_i, d_i^+, d_i^-, f_i^+, T_{L_i}, E_{L_i}, L_{L_i}, T_{D_i}, E_{D_i}, L_{D_i}, S_{ij} \geq 0 \quad (3.17)$$

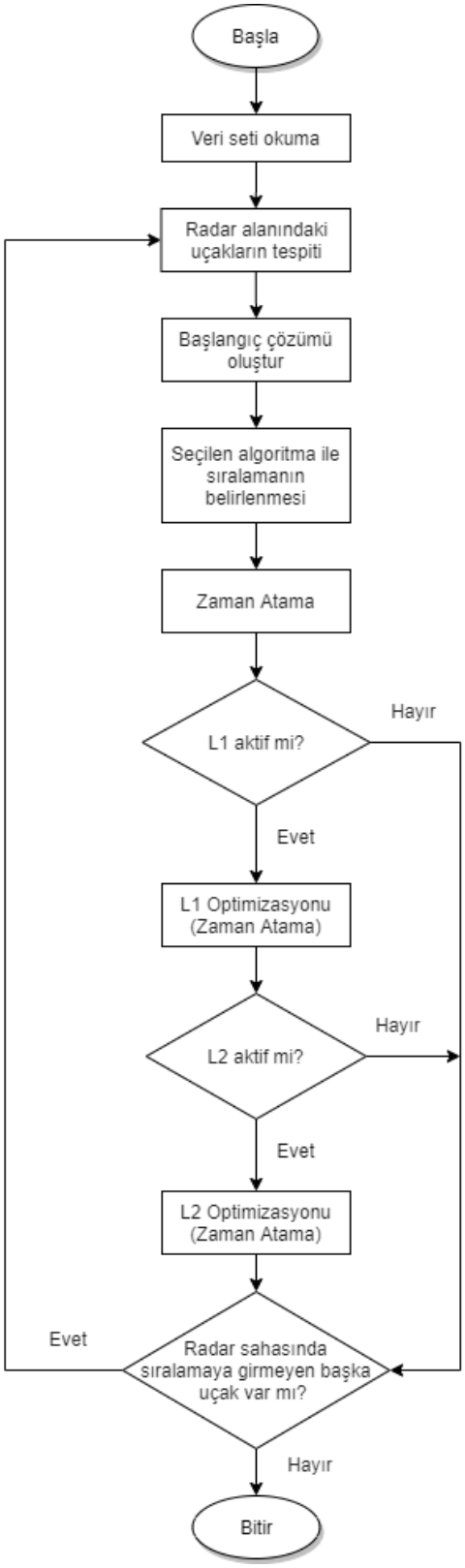
Amaç Fonsiyonu (1), uçakların büyüklüğü de dikkate alınarak iniş ve kalkışlardaki hedeflenen sürelerden sapmaların maliyetinin minimize eder.

2 ve 3 Kısıt denklemeleri sırasıyla her uçağın kendi zaman penceresi içersinde iniş ve kalkışını temsil etmektedir.
4 ve 5 kısıt denklemleri; x_i ve y_i karar değişkenleriyle T_{L_i} ve T_{D_i} parametlerinin a_i ve b_i karar değişkenleriyle bağlantısını göstermektedir.
6 denklemi, havaalanına geç gelen i uçağının gecikme süresinin hangi süre aralıklarında olması gerektiğini ifade etmektedir.
7 denklemi havaalanına erken gelen i uçağının erken gelme süresinin hangi süre aralıklarında olması gerektiğini göstermektedir.
8 denklemi, havaalanından geç kalkan i uçağının hangi süre aralıklarında geç kalkabileceğini temsil etmekdtedir.
9, 10, 11 ve 12 denklemler inen uçaklar, kalkan uçaklar ve aynı piste inip kalkan uçaklar için minimum takip mesafesini göstermektedir.
13, 14,15 denklemleri aynı piste inen ve kalkan uçakları ve esnekliğini ifade eder.

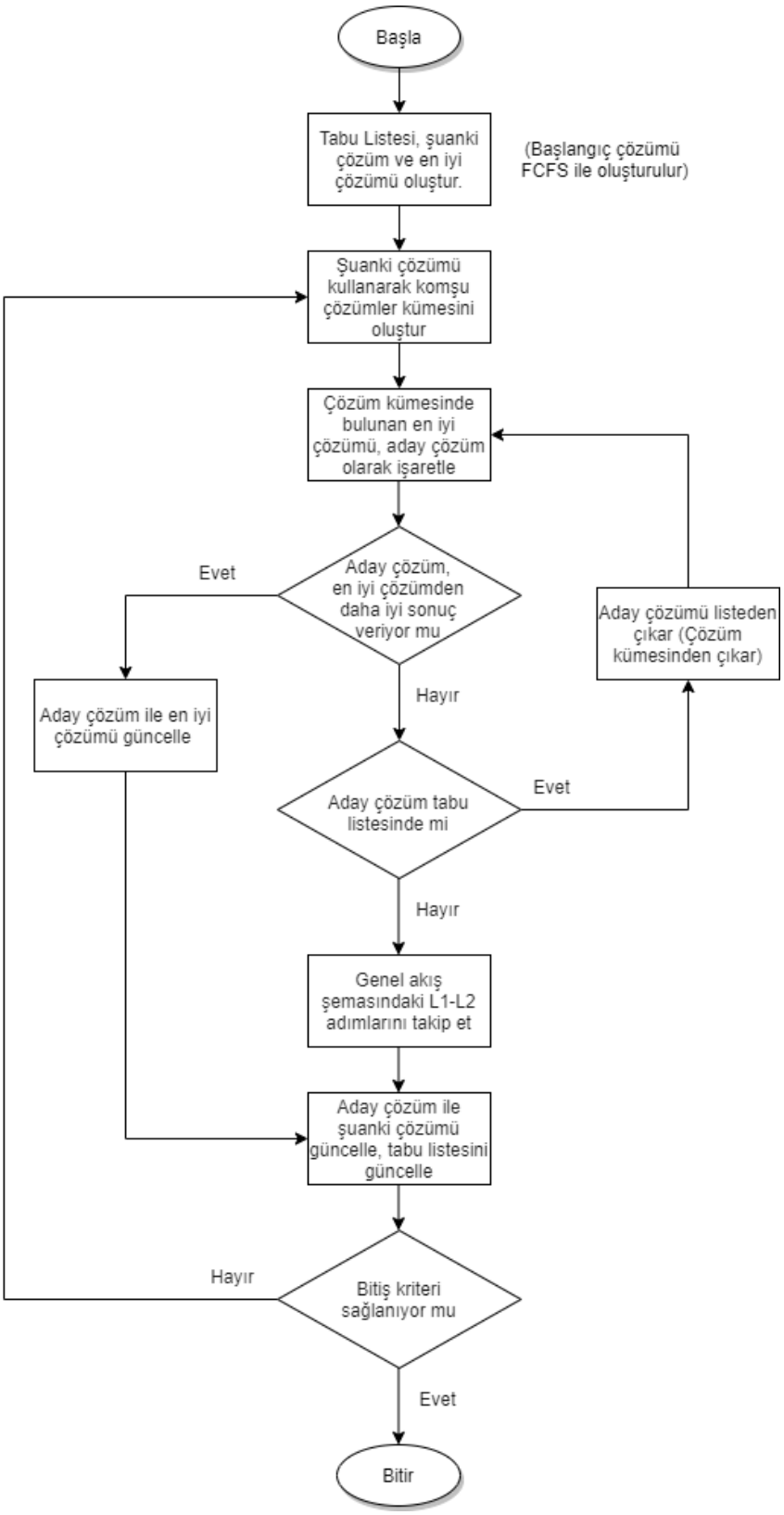
3.2. AKIŞ ŞEMALARI ve SÖZDE KOD

3.2.1. Akış Şemaları

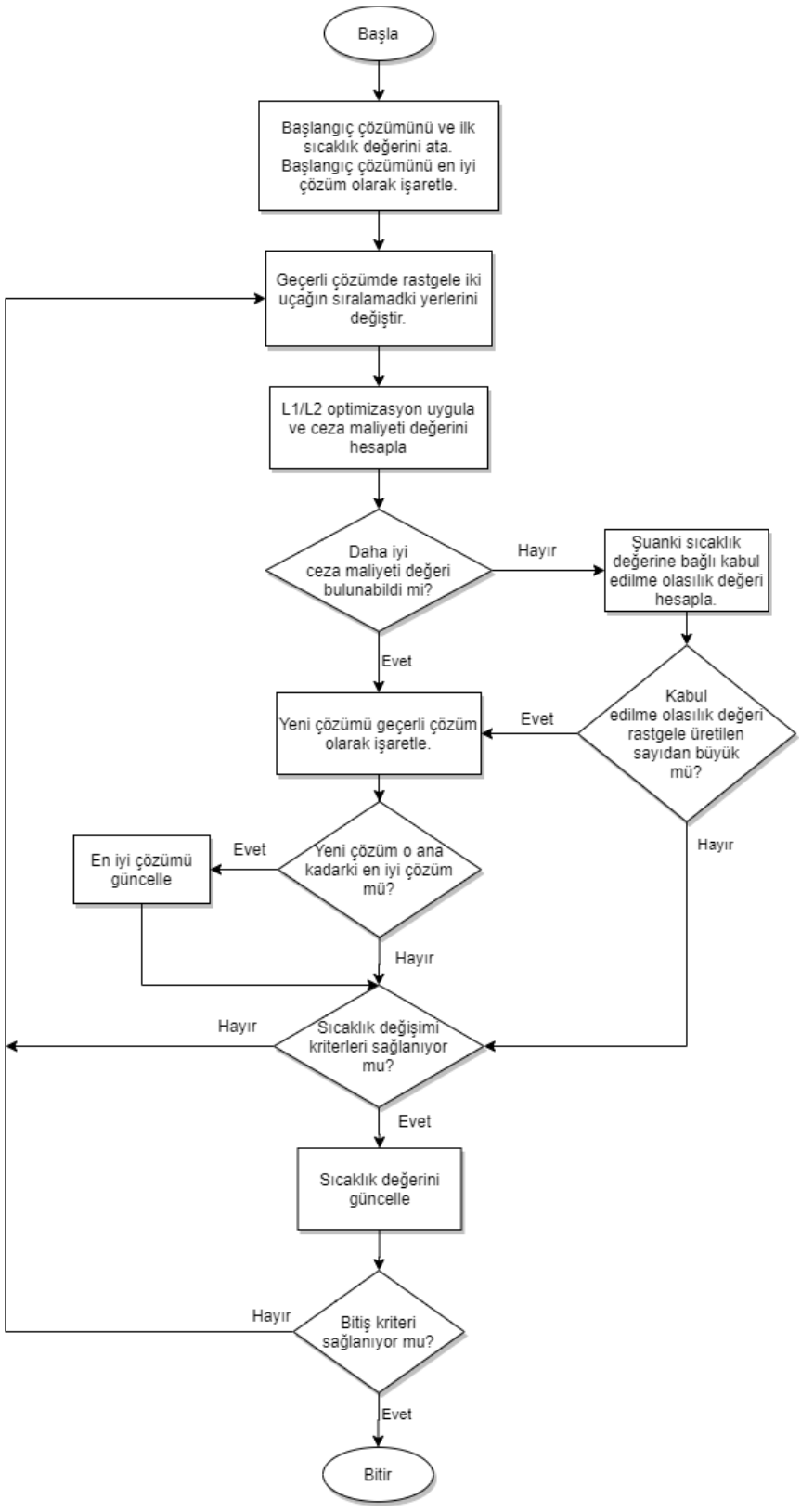
Modele ve özgün sezgisel yöntem kullanılarak geliştirilen algoritmalara göre akış şemaları oluşturulmuştur. Akış şemaları, geliştirilen algoritmanın genel mantığını anlatması açısından öncelikle genel akış şeması ve özgün seviye 1, seviye 2 optimizasyonların 4 metasezgisel algoritmaya uygulanışlarını gösteren akış şemaları olarak ayrılmış ve her biri aşağıdaki şemalar halinde gösterilmiştir.



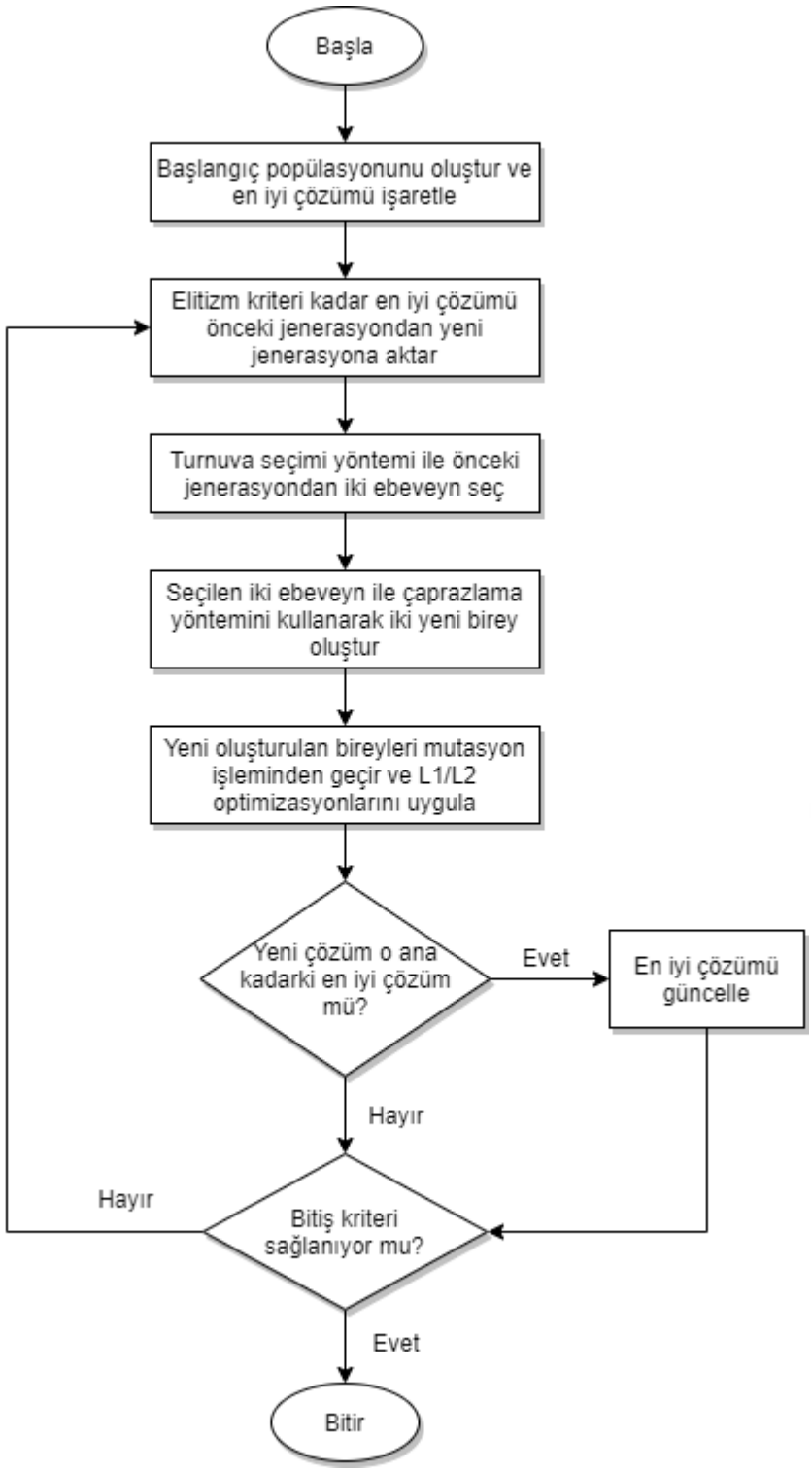
Şekil 3.1: L1 (Seviye1 Opt.)-L2 (Seviye2 Opt.) Algoritma genel akış şeması.



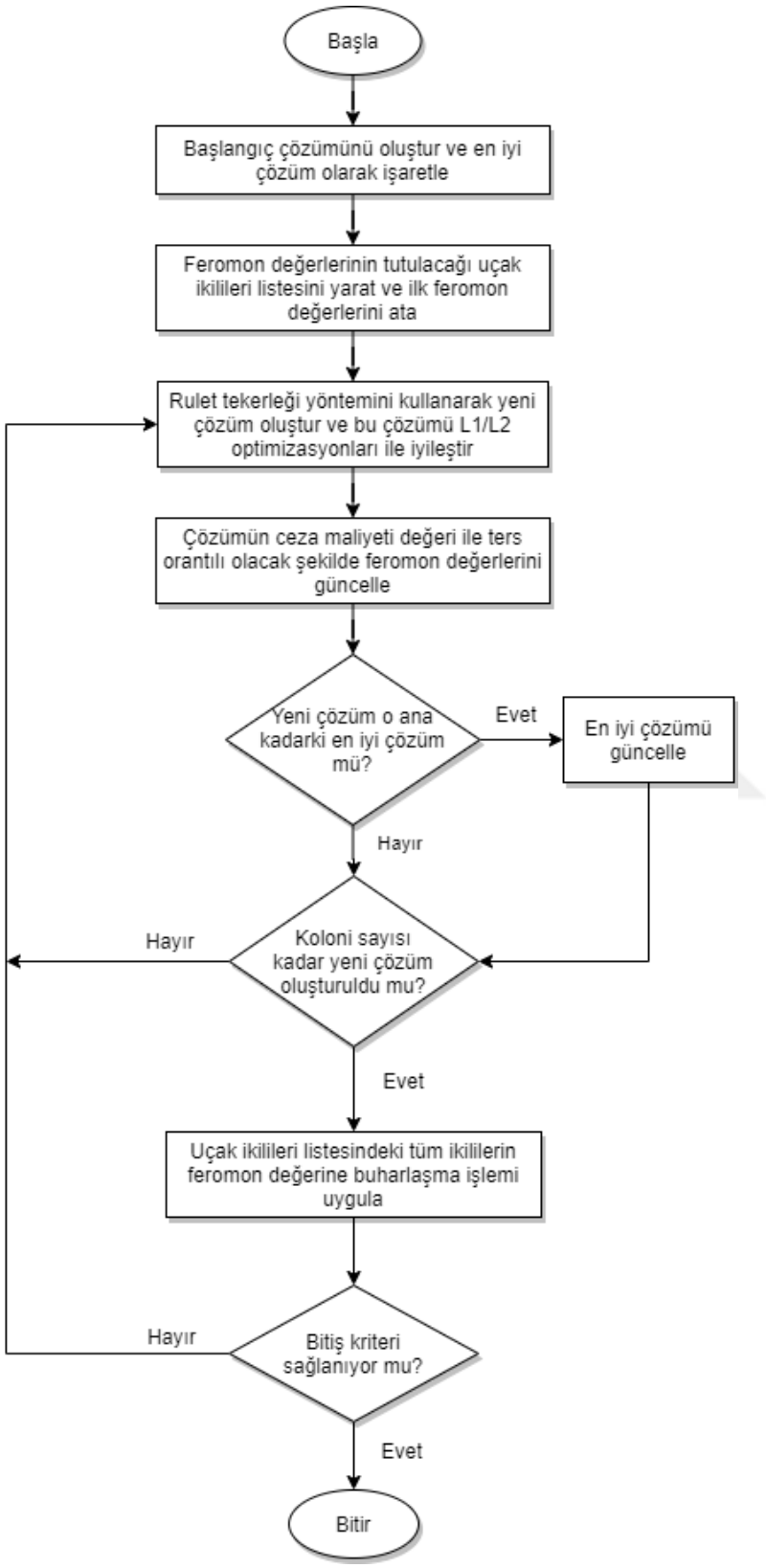
Şekil 3.2: L1 (Seviye1 Opt.), L2 (Seviye 2 Opt.) Tabu arama akış şeması.



Şekil 3.3: L1 (Seviye1 Opt.), L2 (Seviye 2 Opt.) Benzetim tavlaması akış şeması.



Şekil 3.4: L1 (Seviye1 Opt.), L2 (Seviye 2 Opt.) Genetik algoritma akış şeması.



Şekil 3.5: L1 (Seviye1 Opt.), L2 (Seviye 2Opt.) Karınca kolonisi akış şeması.

3.2.2. Sözde Kod

AO: Uçağa atanan iniş zamanı

TO: Uçağın hedeflenen iniş zamanı

TS: Tabu Arama algoritması

SA: Benzetim Tavlaması algoritması

GA: Genetik Algoritma

ACO: Karınca Kolonisi algoritması

```
Parametre dosyasını oku
Simülasyonu yarat
Seçilen algoritmayı verilen parametreler ile başlat

if seçilen algoritma = Tabu Arama
    Sıralama algoritması fonksiyonu := TA
else if seçilen algoritma = Benzetim Tavlaması
    Sıralama algoritması fonksiyonu := SA
else if seçilen algoritma = Genetik Algoritma
    Sıralama algoritması fonksiyonu:= GA
else if seçilen algoritma = Karınca Kolonisi
    Sıralama algoritması fonksiyonu:= ACO
end if

Simülasyona ilk zaman ataması yap
while sıralamaya girmeyen uçak varken
    Radar alanında olan uçakları tespit et
    Başlangıç çözümünü FCFS ile oluştur
    Sıralama algoritmasını çalıştır ve radar alanındaki uçakları sırala
    Tüm uçaklar için iniş zamanı ataması yap
        Yeni oluşturulan sıralamadaki ilk uçağı öncelikli uçak olarak al
        Öncelikli uçak için AO değerini TO ile eşitlemeye çalış
        Sıralamadaki diğer uçaklar için AO değerlerini ata
```

```
if L1 optimizasyonu aktifse
    for sıralamadaki her uçak için
        Döngüde sıradaki uçağı öncelikli uçak olarak ata
        Öncelikli uçak için AO değerini TO ile eşitlemeye çalış
        Sıralamadaki diğer uçaklar için AO değerlerini ata
    end for
end if

if L2 optimizasyonu aktifse
    while sıralamanın optimizasyonu mümkün olduğu sürece
            first_aircraft := sıralamadaki ilk uçak
            last_aircraft := sıralamadaki son uçak
            for sıralamadaki her uçak için
                if sonraki uçak için AO – şimdiki uçak için AO = ayrım zamanı
                    last_aircraft := sonraki uçak
                else
                    last_aircraft := şimdiki uçak
                    group := [first_aircraft, last_aircraft]

                    if first_aiarcraft != last_aircraft
                        EP := gruptaki erken iniş ceza maliyeti
                        LP := gruptaki geç iniş ceza maliyeti

                        if EP > LP
                            Gruptaki tüm AO değerlerini artır
                        else if LP > EP
                            Gruptaki tüm AO değerlerini azalt
                        end if

                    end if
```

first_aircraft := şimdiki uçak
last_aircraft := şimdiki uçak
end if
end for
end while
end if

Simülasyon zamanını güncelle
end while

3.3. KULLANILAN ALGORİTMALARIN ve L1, L2 OPTİMİZASYONLARI

Bu bölümde, çalışmada uçak çizelgeleme problemi (UÇP) için kullanılan algoritmalar ve seviye 1 (L1) ve seviye 2 (L2) optimizasyonlarının çalışma şekli 10 uçak için detaylı olarak anlatılmıştır. Bunun için uçaklara kimlik numaraları (Aircraft ID) verilerek, OR Library'deki uçak operasyon zamanları kullanılmıştır.

3.3.1. Veri Seti

Uçak Kimlik Numarası (Aircraft ID): 0

Uçak Algılanma Zamanı (Aircraft detected time): 54

En Erken Operasyon Zamanı (Earliest operation time): 129

En Geç Operasyon Zamanı (Latest operation time) : 559

Hedeflenen Operasyon Zamanı (Target operation time) : 155

Geç Operasyon Cezası (Late operation penalty) : 10

Erken Operasyon Cezası (Early operation penalty): 10

Uçak Kimlik Numarası (Aircraft ID): 1

Uçak Algılanma Zamanı (Aircraft detected time): 120

En Erken Operasyon Zamanı (Earliest operation time): 195

En Geç Operasyon Zamanı (Latest operation time) : 744

Hedeflenen Operasyon Zamanı (Target operation time) : 258

Geç Operasyon Cezası (Late operation penalty) : 10

Erken Operasyon Cezası (Early operation penalty): 10

Uçak Kimlik Numarası (Aircraft ID): 2

Uçak Algılanma Zamanı (Aircraft detected time): 14

En Erken Operasyon Zamanı (Earliest operation time): 89

En Geç Operasyon Zamanı (Latest operation time) : 510

Hedeflenen Operasyon Zamanı (Target operation time) : 98

Geç Operasyon Cezası (Late operation penalty) : 30

Erken Operasyon Cezası (Early operation penalty): 30

Uçak Kimlik Numarası (Aircraft ID): 3

Uçak Algılanma Zamanı (Aircraft detected time): 21

En Erken Operasyon Zamanı (Earliest operation time): 96

En Geç Operasyon Zamanı (Latest operation time) : 521

Hedeflenen Operasyon Zamanı (Target operation time) : 106

Geç Operasyon Cezası (Late operation penalty) : 30

Erken Operasyon Cezası (Early operation penalty): 30

Uçak Kimlik Numarası (Aircraft ID): 4

Uçak Algılanma Zamanı (Aircraft detected time): 35

En Erken Operasyon Zamanı (Earliest operation time): 110

En Geç Operasyon Zamanı (Latest operation time) : 555

Hedeflenen Operasyon Zamanı (Target operation time) : 123

Geç Operasyon Cezası (Late operation penalty) : 30

Erken Operasyon Cezası (Early operation penalty): 30

Uçak Kimlik Numarası (Aircraft ID): 5

Uçak Algılanma Zamanı (Aircraft detected time): 45

En Erken Operasyon Zamanı (Earliest operation time): 120

En Geç Operasyon Zamanı (Latest operation time) : 576

Hedeflenen Operasyon Zamanı (Target operation time) : 135

Geç Operasyon Cezası (Late operation penalty) : 30

Erken Operasyon Cezası (Early operation penalty): 30

Uçak Kimlik Numarası (Aircraft ID): 6

Uçak Algılanma Zamanı (Aircraft detected time): 49

En Erken Operasyon Zamanı (Earliest operation time): 124

En Geç Operasyon Zamanı (Latest operation time) : 577

Hedeflenen Operasyon Zamanı (Target operation time) : 138

Geç Operasyon Cezası (Late operation penalty) **:** 30

Erken Operasyon Cezası (Early operation penalty): 30

Uçak Kimlik Numarası (Aircraft ID): 7

Uçak Algılanma Zamanı (Aircraft detected time): 51

En Erken Operasyon Zamanı (Earliest operation time): 126

En Geç Operasyon Zamanı (Latest operation time) : 573

Hedeflenen Operasyon Zamanı (Target operation time) : 140

Geç Operasyon Cezası (Late operation penalty) : 30

Erken Operasyon Cezası (Early operation penalty): 30

Uçak Kimlik Numarası (Aircraft ID): 8

Uçak Algılanma Zamanı (Aircraft detected time): 60

En Erken Operasyon Zamanı (Earliest operation time): 135

En Geç Operasyon Zamanı (Latest operation time) : 591

Hedeflenen Operasyon Zamanı (Target operation time) : 150

Geç Operasyon Cezası (Late operation penalty) : 30

Erken Operasyon Cezası (Early operation penalty): 30

Uçak Kimlik Numarası (Aircraft ID): 9

Uçak Algılanma Zamanı (Aircraft detected time): 85

En Erken Operasyon Zamanı (Earliest operation time): 160

En Geç Operasyon Zamanı (Latest operation time) : 657

Hedeflenen Operasyon Zamanı (Target operation time) : 180

Geç Operasyon Cezası (Late operation penalty) : 30

Erken Operasyon Cezası (Early operation penalty): 30

Uçak algılanma zamanı (Aircraft detected time), uçağın radara girdiği zamanı belirtir ve veri setinde olduğu gibi yazılmıştır. Ancak algoritmalar gösterim amacıyla Statik durumda koşturulmuştur, bu nedenle burada görünen radara giriş zamanları yerine tüm uçaklar için 0 değeri kullanılmıştır, yani algoritma koşmaya başladığı anda tüm uçakların bilgileri kullanılmıştır. Bu problemin optimum değeri 700'dür.

Tablo 3.1: Veri Setinin Ayrım Zamanları Bilgisi.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	X	3	15	15	15	15	15	15	15	15
1	3	X	15	15	15	15	15	15	15	15
2	15	15	X	8	8	8	8	8	8	8
3	15	15	8	X	8	8	8	8	8	8
4	15	15	8	8	X	8	8	8	8	8
5	15	15	8	8	8	X	8	8	8	8
6	15	15	8	8	8	8	X	8	8	8
7	15	15	8	8	8	8	8	X	8	8
8	15	15	8	8	8	8	8	8	X	8
9	15	15	8	8	8	8	8	8	8	X

Burada diagonal çizgisi tablonun okunurluğunu artırmak amacıyla "X" ile gösterilmiştir. Veri setindeki gerçek değeri 99999'dur.

3.3.2. Uçak Çizelgeleme Algoritmasında Kullanılan Kavramlar ve Açıklamaları

AD: Uçak Algılanma Zamanı (Aircraft Detected Time)

Veri setinde belirtilen, uçağın radara girdiği zaman değeri.

EO: En Erken Operasyon Zamanı (Earliest Operation Time)

Veri setinde belirtilen, bir uçağın en erken iniş yapabileceği operasyon zaman değerini belirtir. Uçağın bu zaman değerinden önce iniş yapması mümkün değildir.

LO: En Geç Operasyon Zamanı (Latest Operation Time)

Veri setinde belirtilen, bir uçağın en geç iniş yapabileceği operasyon zaman değerini belirtir. Uçağın bu zaman değerinden sonra iniş yapması mümkün değildir.

TO: Hedeflenen Operasyon Zamanı (Target Operation Time)

Veri setinde belirtilen operation time değeri, yani optimum sonuç için mümkün olduğunca yaklaşılması gereken değer.

LP: Geç Operasyon Cezası (Late Operation Penalty)

Veri setinde belirtilen, bir uçağın belirtilen hedef operasyon zamanı (target operation time) değerinden sonra inmesi durumunda, her birim süre (saniye) için alacağı ceza (penalty) değeridir.

EP: Erken Operasyon Cezası (Early Operation Penalty)

Veri setinde belirtilen, bir uçağın belirtilen hedef operasyon zamanı (target operation time) değerinden önce inmesi durumunda, her birim süre (saniye) için alacağı ceza (penalty) değeridir.

AO: Atanan Operasyon Zamanı (Assigned Operation Time)

Algoritmanın uçak için atamış olduğu operasyon zamanı değeri. TO ile arasındaki fark arttıkça ceza (penalty) değeri de artmaktadır.

3.4. KULLANILAN ORTAK FONKSİYONLAR

Bu başlık altında simulasyonda kullanılan ortak fonksiyonlar yer almaktadır. Çalışma şekilleri ve mantıkları anlatılmaktadır.

3.4.1. Zaman Atama Fonksiyonu

Zaman atama fonksiyonu, parametre olarak aldığı zaman değerleri atanmamış sıralamada yer alan uçaklar için aşağıdaki kriterleri baz alarak zaman ataması yapar;

1. İlk sırada yer alan uçak öncelikli kabul edilir ve eğer mümkünse uçağın hedeflenen operasyon süresi değeri neyse atanan operasyon süresi (assigned operation time) değeri olarak da o kullanılır. Mümkün olup olmadığını kontrol etmek için ise bu uçaktan önce iniş yapmış bir uçak olup olmadığı kontrol edilir;
    a. Eğer daha önce iniş yapmış uçak varsa, bu iniş yapan uçak ile listenin ilk sırasında yer alan uçak arasındaki ayrım zamanı (separation time) değeri bulunur ve önceki uçağın operasyon süresi (operation time) değeri ile aralarındaki ayırım zamanı (separation time) değeri toplanır. Böylece listenin ilk sırasındaki uçağın mümkün olan en erken iniş yapabileceği zaman bulunmuş olur.
    b. Daha önce iniş yapmış uçak olmaması durumunda ise listenin ilk sırasındaki uçağın mümkün olan en erken iniş yapabileceği zaman şu anki zamandır, yani simulasyonun o anki zamanıdır.
    c. İlk sıradaki uçağın mümkün olan en erken iniş zamanı, hedeflenen operasyon süresi değerinden daha küçükse, yani hedeflenen operasyon süresi değerinden daha önce iniş yapabilecek durumda ise, atanan operasyon süresi değeri olarak hedeflenen operasyon süresi değeri kullanılır, böylelikle bu uçak nedeniyle herhangi bir ceza (penalty) değeri uygulanmamış olur.
        i. Mümkün olan en erken iniş zamanının hedeflenen operasyon süresi değerinden büyük olması durumunda ise, atanan operasyon süresi değeri olarak mümkün olan en erken iniş zamanı kullanılır.
2. Sıralamada ilk sırada olmayan uçaklar öncelikli uçak değillerdir, yani atanan operasyon süresi değerleri bir önceki uçağın durumuna göre atanır. Burada yapılan kontroller ve işlemler aşağıdaki gibidir;

a. Bir önceki uçağın atanan operasyon süresi değeri ile aralarındaki ayırım zamanı değeri toplanarak sıradaki uçak için mümkün olan en erken iniş zamanı bulunmuş olur. Daha sonra mümkün olan en erken iniş zamanı ile hedeflenen operasyon süresi değerleri karşılaştırılır. Eğer mümkün olan en erken iniş zamanı hedeflenen operasyon süresi değerinden küçük ise; atanan operasyon süresi değeri olarak hedeflenen operasyon süresi değeri kullanılır, aksi durumda atanan operasyon süresi değeri olarak mümkün olan en erken iniş zamanı değeri kullanılır.

3. Bu işlem listenin tamamında uygulanır.

Tablo 3.2: Veri setindeki ilk üç uçak (Uçak Kimlik No: 0, 1, 2) için örnekleme

Sıralama	0	1	2
TO (Hedeflen Operasyon Zamanı)	155	258	98
AO (Atanan Operasyon Zamanı)	-	-	-

1) Sıralamadaki öncelikli uçak 0 nolu uçaktır. Bundan önce iniş yapmış başka uçak olmadığı için hedeflenen operasyon süresi değeri atanan operasyon zamanı değeri olarak kullanılabilir. Yani 0 nolu uçak için atanan operasyon süresi değeri 155 olarak bulunur.
2) Sıradaki uçak 1 nolu uçaktır. Önünde yer alan 0 nolu uçak ile arasındaki ayırım zamanı değeri tablodan 3 olarak kulunur. Bu durumda 1 nolu uçak için mümkün olan en erken iniş zamanı önceki uçağın atanan operasyon zamanı değeri ile aralarındaki ayırım zamanı değerinin toplamı, yani 155+3 = 158'dir. Ancak ceza maliyeti değerini minimumda tutmak istediğimiz için, 1 nolu uçağın atanan operasyon zamanı değeri olarak hedeflenen operasyon zamanı değerinin kullanılmasının mümkün olup olmadığı kontrol edilir. 158 < 258 olduğu için, yani mümkün olan en erken iniş zamanı hedeflenen operasyon zamanı değerinden küçük olduğu için, 1 nolu uçakta atanan operasyon zamanı değeri olarak hedeflenen operasyon zamanı değerinin kullanılması

mümkündür. Böylelikle 1 nolu uçak için atanan operasyon zamanı değeri olarak hedeflenen operasyon zamanı değeri, yani 258 kullanılır.

3) Sıradaki uçak 2 nolu son uçaktır. Önünde yer alan 2 nolu uçak ile arasındaki ayırım zamanı değeri tablodan 15 olarak bulunur. Bu durumda 2 nolu uçak için mümkün olan en erken iniş zamanı önceki uçağın atanan operasyon zamanı değeri ile aralarındaki ayırım zamanı (separation time) değerinin toplamı, yani 258+15 = 273'dür. Ancak ceza değerini minimumda tutmak istediğimiz için, 2 nolu uçağın atanan operasyon zamanı değeri olarak hedeflenen operasyon zamanı değerinin kullanılmasının mümkün olup olmadığı kontrol edilir. 273 > 98 olduğu için, yani mümkün olan en erken iniş zamanı hedeflenen operasyon zamanı değerinden büyük olduğu için, 2 nolu uçakta uçağın atanan operasyon zamanı değeri olarak hedeflenen operasyon zamanı değerinin kullanılması mümkün değildir, bu sebeple mümkün olan en erken iniş zamanı değeri kullanılacaktır. Böylelikle 2 nolu uçak için uçağın atanan operasyon zamanı değeri olarak mümkün olan en erken iniş zamanı değeri, yani 273 kullanılır.

Bu işlem sonrasında uçağın atanan operasyon zamanı değerleri ile birlikte zamanlama aşağıdaki gibi olmaktadır.

Tablo 3.3: TO ve AO'ya göre uçak sıralama.

Sıralama	0	1	2
TO (Hedeflenen Z.)	155	258	98
AO (Atanan Z.)	155	258	273

3.4.2. Sıralama Doğrulama Fonksiyonu

Zaman Atama Fonksiyonu sıralamayı değiştirmediği ve sadece zamanları her uçak için, sıralamaya etki etmeden mümkün olan en iyi değer olarak belirlemeye çalıştığı için bazı durumlarda oluşan sıralama geçersiz olabilir. Doğrulaması yapılan bir sıralamanın geçersiz olduğu tespit edildiği durumda bu sıralama silinir ve üzerinde daha fazla işlem yapılmaz.

Örneğin aşağıdaki gibi hayali (veri setinden bağımsız) bir sıralamada atanacak zamanların, sıralama değiştirilmeden geçerli olması mümkün değildir. Örnekte tüm uçaklar arası ayrım zamanı değerinin 20 olduğu kabul edilmiştir.

Tablo 3.4: EO, LO, TO ve AO'ya göre uçak sıralama.

Sıralama	6	4	5
EO (En erken Z.)	100	40	150
LO (En Geç Z.)	250	98	500
TO(Hedeflenen Z.)	155	58	350
AO (Atanan Z.)	155	175	350

Yukarıda görüldüğü gibi değerlerle doğrulama fonksiyonuna gelen bir sıralama için aşağıdaki koşullar sırayla kontrol edilir;

1. İlk olarak sıralamada birinci sırada yer alan uçak ile, eğer varsa, en son iniş yapmış olan uçak arasında yeterli ayırım zamanı olup olmadığı kontrol edilir.
2. Daha sonra sıralamadaki tüm uçaklar için;

a. Atanan zaman değerinin, o uçak için veri setinde verilmiş olan en erken iniş zamanı (EO) değerinden büyük veya eşit olup olmadığı kontrol edilir.
b. Atanan zaman değerinin, o uçak için veri setinde verilmiş olan en geç iniş zamanı (LO) değerinden küçük veya eşit olup olmadığı kontrol edilir. Örneğin yukarıdaki örnekte 4 nolu uçak için belirlenmiş olan AO değeri (175), hayali veri setinde verilmiş olan LO değerinden (98) küçük veya eşit değildir. Bu durumda doğrulaması yapılan sıralamanın geçersiz olduğu belirlenmiş olur.
c. Atanan zaman değerinin simülasyon zamanı değerinden, yani o anki zaman değerinden büyük olup olmadığı kontrol edilir.
d. Sıralamada, zamanlaması kontrol edilen uçaktan sonra başka uçak da var ise;
    i. Şuanki uçak ile sonraki uçak arasındaki ayırım zamanı değerinin, uçakların atanan zamanları arasındaki değerden küçük veya eşit olup olmadığı kontrol edilir.

Tüm bu kontrollerin sonucunda uyumsuzluk tespit edilmezse doğrulaması yapılan sıralamanın geçerli olduğuna karar verilir.

3.4.3. L1 Optimizasyon Fonksiyonu

Birinci seviye optimizasyon (L1) fonksiyonu "Zaman Atama Fonksiyonu" ile benzer mantıkta çalışır. Aralarındaki tek fark zaman atama fonksiyonunda öncelikli uçak ilk sıradaki uçaktır ve atamanın ilk önceliği, eğer mümkünse, ilk sıradaki uçağın atanan operasyon zamanı değeri olarak hedeflenen operasyon zamanı değerini atamaktır.

L1 optimizasyonda ise uçak sayısı kadar yeni sıralama oluşturulur ve her sıralamadaki öncelikli uçak değişir, böylelikle her uçağın sırayla öncelikli uçak olduğu sıralamalar ve zaman atamaları yapılmış olur. L1 optimizasyonun diğer bir önemli ayrıntısı ise, var olan sıralamalar üzerinden, uçakların sırasını değiştirmeden, yeni zaman atama değerleri ile aynı sıralamanın farklı varyasyonlarını oluşturuyor olmasıdır.

Zaman Atama Fonksiyonu'nda kullanılan sıralama ile örnekleme yapılacak olunursa;

Tablo 3.5: L1 optimizasyonla TO ve AO'ya göre uçak sıralama.

Sıralama	0	1	2
TO (Hedef)	155	258	98
AO (Atanan)	-	-	-

Sıralamada 3 adet uçak olduğu için, L1 optimizasyon fonksiyonu 3 adet döngü ile toplamda 3 adet yeni sıralama üretecektir, ancak bu yeni üretilen sıralamalardaki uçakların sırası değişmeyecektir, sadece uçaklara atanan uçağın atanan operasyon zamanı değişecektir.

3.4.3.1.Yeni Üretilen 1. Sıralama

Birinci sıralama olduğu için öncelikli uçak olarak birinci sıradaki yani 0 nolu uçak kabul edilir, ve bu uçağın uçağın atanan operasyon zamanı değerinin hedeflenen operasyon zamanı değerine eşit olması sağlanmaya çalışılır. L1 optimizasyon fonksiyonun ilk döngüsünde atadığı zaman değerleri ile Zaman Atama Fonksiyonu'nda atanan zamanlar aynı olacaktır.

Aynı kurallar ve uygulamalar geçerli olacağı için zaman atama sonrasında sıralamanın son hali aşağıdaki gibi olacaktır;

Tablo 3.6: L1 optimizasyonla TO ve AO'ya göre uçak sıralama.

Sıralama	0	1	2
TO (Hedef)	155	258	98
AO (Atanan)	155	258	273

3.4.3.2.Yeni Üretilen 2. Sıralama

İkinci sıralama olduğu için öncelikli uçak olarak ikinci sıradaki yani 1 nolu uçak kabul edilir, ve bu uçağın atanan operasyon zamanı değerinin hedeflenen operasyon zamanı değerine eşit olması sağlanmaya çalışılır. Öncelikli uçak ikinci sıradaki, 1 nolu, uçak olduğu için ilk olarak bu uçağın atanan operasyon zamanı değeri olarak hedeflenen operasyon zamanı değeri atanır, yani 258. Daha sonra önündeki ve arkasındaki uçakların zaman atama işlemi yapılır.

Önündeki, 0 nolu, uçak için yapılabilecek en geç atama hesaplanır, bu da 1 nolu uçak ile arasındaki ayırım zamanı ile belirlenir. Tablodan bakıldığında 0 nolu uçak ile 1 nolu uçak arasındaki ayırım zamanı değerinin 3 olduğu görülür. Bu durumda 0 nolu uçak için yapılacak zaman ataması değeri en fazla (en geç) 258-3 = 255 olabilir. Sonraki adım olarak 0 nolu uçağın zaman atamasında yapılabilecek en erken zaman bulunur. 0 nolu uçak sıralamanın en başında olduğu için daha önce iniş yapmış uçak varsa, bununla arasındaki ayırım zamanı değeri hesaplanır, ancak bu örnekte daha önce iniş yapmış uçak olmadığı için ayırım zamanı kontrolüne gerek yoktur; simulasyon zamanı, yani şuanki zaman, 0 nolu uçak için atanabilecek en erken iniş zamanını verir. Sonraki adım olarak da 0 nolu uçağın zaman ataması olarak hedeflenen operasyon zamanı değerinin kullanılıp kullanılamayacağına bakılır. 155 değeri uygun olduğu için bu değer kullanılır.

Öncelikli uçağın, yani ikinci sıradaki (1 nolu) uçağın önünde 0 no.lu uçaktan başka uçak olmadığı için arkasındaki uçaklarla zaman atama işlemine devam edilir ve arkasında bulunan tek uçak olan 2 nolu uçakta ayırım zamanı kısıtı nedeniyle atanan operasyon zamanı değeri olarak 273 kullanılır. 2 nolu uçak için hedeflenen operasyon zamanı değeri kullanılamaz çünkü hedeflenen operasyon zamanı değeri, atama yapılabilecek mümkün olan en erken zamandan, 273'ten, küçüktür. Sıralamanın son hali aşağıdaki gibi olacaktır;

Tablo 3.7: L1 optimizasyonla TO ve AO'ya göre uçak sıralama.

Sıralama	0	1	2
TO(Hedeflenen)	155	258	98
AO (Atanan)	155	258	273

3.4.3.3. Yeni Üretilen 3. Sıralama

Üçüncü sıralama olduğu için öncelikli uçak olarak üçüncü sıradaki yani 2 nolu uçak kabul edilir, ve bu uçağın atanan operasyon zamanı değerinin hedeflenen operasyon zamanı değerine eşit olması sağlanmaya çalışılır. Öncelikli uçak üçüncü sıradaki, 2 nolu, uçak olduğu için ilk olarak bu uçağın atanan operasyon zamanı değeri olarak hedeflenen operasyon zamanı değeri atanır, yani 98. Daha sonra önündeki ve arkasındaki uçakların zaman atama işlemi yapılır. Bu

örnekte üçüncü uçağın arkasında uçak olmadığı için, yani lsitedeki son uçak olduğu için, sadece önündeki uçakların zaman ataması yapılacaktır.

Önündeki, 1 nolu, uçak için yapılabilecek en geç atama hesaplanır, bu da 2 nolu uçak ile arasındaki ayırım zamanı ile belirlenir. Tablodan bakıldığında 1 nolu uçak ile 2 nolu uçak arasındaki ayırım zamanı değerinin 15 olduğu görülür. Bu durumda 1 nolu uçak için yapılacak zaman ataması değeri en fazla (en geç) 98-15 = 83 olabilir. 1 nolu uçağın hedeflenen operasyon zamanı değeri, yapılabilecek en geç atama değerinden büyük olduğu için kullanılamaz ve 1 nolu uçak için atanan operasyon zamanı değeri olarak 83 kullanılır.

1 nolu uçağın önündeki 0 nolu uçak için de aynı durum geçerlidir; 1 nolu uçak ile 0 nolu uçak arasındaki kısıtlar kontrol edildiğinde, 0 nolu uçak için de hedeflenen operasyon zamanı değerinin kullanılmasının mümkün olmadığı ve 1 nolu uçak ile arasındaki ayırım zamanı değeri, 3 saniye, kadar fark olacağı görülür. Bu durumda 0 nolu uçak için atanan operasyon zamanı değeri de 83-3 = 80 olarak bulunur ve sıralamanın son hali aşağıdaki gibi olacaktır;

Tablo 3.8: L1 optimizasyonla TO ve AO'ya göre uçak sıralama.

Sıralama	0	1	2
TO (Hedeflenen)	155	258	98
AO (Atanan)	80	83	98

Bu şekilde 3 adet yeni sıralama oluşturulmuş olunur ve bunlar, geçerli olmaları durumunda listeye eklenir; aksi durumda ise geçersiz olanlar silinir.

Not: Fark edileceği gibi ilk iki sıralama geçerli sonuçlar verirken sonuncu sıralama geçersizdir. Çünkü 0 nolu uçak ile 1 nolu uçak için belirlenen atanan operasyon zamanı değerleri, bu uçaklar için veri setinde verilmiş olan EO (En Erken Operasyon Zamanı) değerlerinden daha küçüktür. Tüm algoritma ve optimizasyon (L1, L2) işlemleri sonucunda üretilen sıralamaların hepsi

“Sıralama Doğrulama Fonksiyonu” ile geçerli olup olmadıkları kontrol edilir ve geçersiz olduğu tespit edilen sıralamalar silinir.

3.4.4. L2 Optimizasyon Fonksiyonu

İkinci seviye optimizasyon (L2) fonksiyonunun temel mantığı, ayırım zamanı ile birbirinden ayrılmış olan, yani aralarındaki AO (atanan operasyon zamanı) değerleri arasındaki delta’nın aralarındaki ayırım zamanına eşit olduğu uçakların gruplanması ve bu grup içindeki uçakların ceza değerlerinin incelenmesine dayanır. Kabaca şöyle söylenebilir; birbirine ayırım zamanı ile bağlı uçakların oluşturduğu bir alt sıralama grubu içinde erken ceza (early penalty) alanların sayısı ile geç ceza alanların sayısı karşılaştırılır, eğer erken ceza alanların sayısı geç ceza alanlardan daha fazla ise, bu grubun tamamının AO (atanan operasyon zamanı) değerlerinin 1’er artırılması ile, yani tüm grubun zaman diliminde kaydırılması ile, toplam ceza maliyeti değerinde iyileştirme elde edilebilir. Aynı işlemin tersi de mümkündür, yani geç ceza (late penalty) alanların sayısı erken ceza alanların sayısından fazla ise, grubunun tamamının AO değerleri 1’er azaltılır. Temel amaç çoğunluğun olduğu kısmın etkisini azaltmak, dolayısı ile toplam ceza maliyeti değerini azaltmaktır. L2 optimizasyonu, tamamen yeni AO değeri ataması yapmaz, var olan AO değerlerini değiştirerek iyileştirme sağlamaya çalışır.

Aşağıdaki hayali (Veri setine dayanmayan) örnek ile açıklamak için aşağıdaki sıralamanın optimizasyon için fonksiyona gönderildiğini kabul edersek; Örneğin anlaşılırlığını artırmak için tüm uçaklar arasındaki ayrım zamanı değerinin 5 olduğu ve her birim süre için ceza maliyeti katsayısının 1 olduğu kabul edilmiştir.

Tablo 3.9: L2 optimizasyonla TO ve AO'ya göre uçak sıralama.

Sıralama	0	1	2	3	4	5	6	7	8	9	10	11
TO	8	15	22	27	32	45	56	70	73	78	87	92
AO	10	15	20	25	30	45	56	70	75	80	85	90

Algoritma ilk olarak birbirine ayırım zamanı ile bağlı olan ardışık uçaklardan oluşan grupları bulmaya çalışacaktır, ve bu gruplar üzerinde sırayla işlem yapacaktır. İlk uçaktan itibaren incelemeye başlandığında ilk 5 uçağın (0 nolu uçaktan başlayıp 4 nolu uçakta biten) birbirine ayırım zamanı ile bağlı olduğu ve bir grup oluşturduğu görülür;

Tablo 3.10: L2 optimizasyonla TO ve AO'ya göre uçak sıralama ve ceza maliyetleri

Sıralama	0	1	2	3	4
TO	8	15	22	27	32
AO	10	15	20	25	30
Ceza	2	0	2	2	2
Ceza. Türü	LP	-	EP	EP	EP

Ceza Maliyeti: Bu uçağa yapılan AO (Atanan Operasyon Zamanı) değeri nedeniyle oluşan ceza maliyeti değeri.

Ceza Tipi: Bu uçağın aldığı ceza tipi; **EP:** Erken Ceza Maliyeti, **LP:** Geç Ceza Maliyeti

Grup içindeki EP ve LP değerleri karşılaştırılır ve eşit olmadıkları görüldüğü için iyileştirme yapılmaya başlanır. Bu grup içindeki toplam ceza maliyeti değeri 8'dir. 1 uçak hedeflenen operasyon süresi değerini AO (atanan operasyon zamanı) olarak kullanabilmektedir, ancak 1

uçak LP, geç ceza maliyeti, alırken 3 uçak ise EP, erken ceza maliyeti, almıştır. Erken cezaların, geç cezaların sayısından fazla olması nedeniyle; tüm grubun AO değerlerinin 1'er artırılması durumunda toplam ceza maliyeti değeri azalacaktır. Bu durumda yeni AO ve ceza maliyeti değerleri aşağıdaki gibi olacaktır.

Tablo 3.11: L2 optimizasyonla TO ve AO'ya göre uçak sıralama ve ceza maliyeti.

Sıralama	0	1	2	3	4
TO	8	15	22	27	32
AO	11	16	21	26	31
Ceza	3	1	1	1	1
Ceza Türü	LP	LP	EP	EP	EP

Yeni değerlerle elde edilen ceza maliyeti değeri 7'ye inmiştir, ve toplamda 1 birim iyileşme elde edilmiştir. Yeni durumda da iyileştirme yapmak mümkündür ancak fonksiyon iyileştirmeye devam etmez, bir sonraki grubu (birbirine ayırım zamanı ile bağlı uçakları) aramaya başlar. Bu kapsamda diğer grubu sıralamanın sonunda bulur, ikinci grup 7 nolu uçak ile başlar ve 11 nolu, sonuncu, uçak ile biter. Grup aşağıdaki gibidir;

Tablo 3.12: L2 optimizasyonla TO ve AO'ya göre uçak sıralama ve ceza maliyeti.

Sıralama	7	8	9	10	11
TO	70	73	78	87	92
AO	70	75	80	85	90
Ceza	-	2	2	2	2
Ceza Türü	-	LP	LP	EP	EP

Ancak bu grup içindeki EP ve LP sayıları eşit olduğu için iyileştirme yapılamayacağından bu grupta herhangi bir işlem yapılmaz.

Not: L2 optimizasyon fonksiyonunun, iyileşme sağlanabildiği ve geçerli sıralama elde edildiği sürece algoritma tarafından tekrar tekrar çağrılması ile elde eilebilecek tüm iyileştirmelerin yapılması sağlanmaktadır.

3.5. TABU ARAMA ALGORİTMASI

S_0 (başlangıç çözümü, FCFS): {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

S_0, algoritma uygulanmadan önce havadaki görüntüdür, FCFS kuralını uygular, radara ilk giren uçağın ilk sırada olduğu, yani ilk iniş hakkını kazandığı sıralamadır.

Başlangıç çözümü algoritmaya sokulmadan önce zaman atama fonksiyonuna sokulur, ve sıralaması değiştirilmeden uçakların operasyon zamanı değerleri hesaplanır ve atanır. İlk anda hesaplanmış bir operasyon zamanı değeri olmayan uçakların, zaman atama fonksiyonu sonrasındaki operasyon zamanı değerleri ve var olan sıralamanın ceza maliyeti değeri aşağıdaki gibi olmaktadır.

Tablo 3.13: Tabu arama algoritması TO ve AO'ya göre uçak sıralama

Sıralama	0	1	2	3	4	5	6	7	8	9
TO	155	258	98	106	123	135	138	140	150	180
AO	155	258	273	281	289	297	305	313	321	329

Ceza maliyeti değeri: 40140

Algoritmanın başlangıcında iki en iyi ceza değeri tutulur; bir tanesi genel en iyi ceza, diğeri de yerel en iyi ceza değeridir. Genel en iyi ceza maliyeti değeri algoritmanın o ana kadar bulduğu en iyi değerdir, yerel en iyi ceza değeri ise algoritmanın bir döngüsü boyunca bulunan en iyi ceza maliyeti değeridir.

3.5.1. Tabu Arama Döngüsü

Her döngüde üzerinde işlem yapılmak üzere baz olarak alınan bir sıralama vardır, ilk döngüde baz olarak alınan döngü, algoritma ilk kez koştuğu için, başlangıç çözümüdür.

Döngünün sonuç sıralamasını bulmak için; her uçak ikilisi birbiri ile yer değiştirilerek (swap) yeni bir sıralama üretilmiş olunur. Daha sonra bu sıralamaya zaman ataması yapılması için zaman atama fonksiyonuna gönderilir. Zaman ataması tamamlanan sıralamanın doğrulamasının yapılması için sıralama doğrulama fonksiyonuna gönderilir. Eğer bu sıralama geçerli bir sıralama ise ceza değeri hesaplanır.

Her bir yer değiştirme işlemi ile yeni bir sıralama üretilmiş olunur. Veri seti baz alınacak olunursa, her bir döngüde yapılan yer değiştirme işlemleri aşağıdaki gibidir;

0 nolu uçak ile sırasıyla: 1, 2, 3, 4, 5, 6, 7, 8, 9 nolu uçaklar tek tek yer değiştirme işlemine sokulur;

1 nolu uçak ile sırasıyla: 2, 3, 4, 5, 6, 7, 8, 9 nolu uçaklar tek tek yer değiştirme işlemine sokulur;

2 nolu uçak ile sırasıyla: 3, 4, 5, 6, 7, 8, 9 nolu uçaklar tek tek yer değiştirme işlemine sokulur;

...

8 nolu uçak ile 9 nolu uçağın da yer değiştirme işlemine sokulmasıyla birlikte yer değiştirme işlemleri tamamlanmış olur.

Not: Yer değiştirme işlemi yapılmadan önce, bu yer değiştirme hareketinin Tabu listesinde olup olmadığı kontrol edilir. Tabu listesinde olması durumunda ise, bu yer değiştirme ile elde edilen sıralamada genel en iyi ceza maliyeti değerinden de daha iyi bir değer elde ediliyorsa Tabu gözardı edilir, aksi durumda bu yer değiştirme işlemine izin verilmez.

Sonuç olarak her bir döngüde; uçak sayısının 2'li kombinasyonu kadar yeni sıralama üretilmiş olunur. Bizim örneğimizde ise C(10,2) = 45 adet yeni sıralama üretilmiştir.

Aynı döngü içinde üretilen yeni sıralamaların, döngünün başındaki baz sıralamadan tek farkı sadece iki uçağın yerlerinin birbirleri ile değiştirilmiş olmasıdır. İlk döngüde üretilen yeni sıralamaların bazıları ve hangi uçaklar arasında yer değiştirme işlemi yapıldığı aşağıda görülebilir;

1. Sıralama:

1	0	2	3	4	5	6	7	8	9

2. Sıralama:

2	1	**0**	3	4	5	6	7	8	9

3. Sıralama:

3	1	2	0	4	5	6	7	8	9

4. Sıralama:

4	1	2	3	0	5	6	7	8	9

...

...

43. Sıralama:

0	1	2	3	4	5	6	8	7	9

44. Sıralama:

0	1	2	3	4	5	6	9	8	7

45. Sıralama:

0	1	2	3	4	5	6	7	9	8

Daha sonra yeni üretilen her sıralama için zaman ataması ve sıralamanın geçerli olup olmadığının kontrolü yapılır ve geçerli olan sıralamalar için ceza maliyeti değerleri hesaplanırken, geçersiz olan sıralamalar silinir.

L1 optimizasyonun aktif olması durumunda; geçerli olan tüm sıralamalar L1 optimizasyon fonksiyonuna gönderilir ve L1 optimizasyonu ile uçak sayısı kadar, bu örnekte 10, yeni sıralama üretilir. Bu yeni üretilen sıralamalar için de zaman ataması ve geçerli olup olmadıklarının kontrolü yapılır, geçersiz olan sıralamalar silinir.

Bu örnekte; L1 optimizasyonu ile maksimum 45 (Tabu ile üretilen sıralama sayısı) x 10 (her Tabu sıralaması için L1 ile üretilen sıralama sayısı) = 450 adet yeni sıralama üretilecektir ancak pratikte bunların bir kısmı geçersiz oldukları için silineceklerdir.

Son durumda ise; döngü boyunca üretilen yeni sıralamaların tamamının içindeki en iyi sıralama bulunur ve L2 optimizasyonun aktif olması durumunda bu en iyi sıralama L2 optimizasyon fonksiyonuna gönderilir. L2 optimizasyonu ile daha iyi (ceza maliyeti değeri daha küçük) bir sıralama bulunması durumunda bu yeni sıralama, aksi durumda ise önceki en iyi sıralama yerel en iyi sıralama olarak kaydedilir. Daha sonra genel en iyi sıralama ile yerel en iyi sıralamanın

ceza maliyeti değerleri karşılaştırılır, en son bulunan yerel en iyi sıralama daha iyi ise genel en iyi sıralama güncellenir.

Sonrasında ise bu sıralamayı elde etmek için yapılan yer değiştirme işlemi sırasındaki hareket (örneğin 3 nolu uçak ile 7 nolu uçağın yer değiştirmesi) Tabu listesine eklenir.

Döngü sonunda Tabu listesindeki kayıtlar güncellenir, unutulma vakti gelen (listeden çıkacak yasaklı yer değiştirme) silinir.

Tüm döngüler (bu örnekte 100) bittikten sonra elde edilen en iyi sıralama ise aşağıdaki gibi olmaktadır;

3.5.2. Tabu Arama Sonuç

Sadece Tabu Arama:

Tablo 3.14: Tabu arama algoritması TO ve AO'ya göre uçak sıralama ve ceza maliyeti

Sıralama	2	3	4	5	7	8	6	9	0	1
TO	98	106	123	135	140	150	138	180	155	258
AO	98	106	123	135	143	151	159	180	195	258
LP	30	30	30	30	30	30	30	30	10	10
EP	30	30	30	30	30	30	30	30	10	10
Ceza Etkisi	-	-	-	-	90	30	630	-	400	-

Ceza maliyeti değeri: 1150

L1 Aktif:

Tablo 3.15: L1 optimizasyonla –Tabu arama algoritmasında TO ve AO'ya göre uçak sıralama ve ceza maliyeti

Sıralama	2	3	4	5	6	7	8	0	9	1
TO	98	106	123	135	138	140	150	155	180	258
AO	98	106	118	126	134	142	150	165	180	258
LP	30	30	30	30	30	30	30	10	30	10
EP	30	30	30	30	30	30	30	10	30	10
Ceza Etkisi	-	-	150	270	120	60	-	100	-	-

Ceza maliyeti değeri: 700 (Optimum)

L2 Aktif:

Tablo 3.16: L2 optimizasyonla –Tabu arama algoritmasında TO ve AO'ya göre uçak sıralama ve ceza maliyeti

Sıralama	2	3	4	5	6	7	8	0	9	1
TO	98	106	123	135	138	140	150	155	180	258
AO	98	106	118	126	134	142	150	165	180	258
LP	30	30	30	30	30	30	30	10	30	10
EP	30	30	30	30	30	30	30	10	30	10
Ceza Etkisi	-	-	150	270	120	60	-	100	-	-

Ceza maliyeti değeri: 700 (Optimum)

3.6. TAVLAMA BENZETİMİ ALGORİTMASI

S_0 (başlangıç çözümü, FCFS): {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

S_0, algoritma uygulanmadan önce havadaki görüntüdür, FCFS kuralını uygular, radara ilk giren uçağın ilk sırada olduğu, yani ilk iniş hakkını kazandığı sıralamadır.

Başlangıç çözümü algoritmaya sokulmadan önce zaman atama fonksiyonuna sokulur, ve sıralaması değiştirilmeden uçakların operasyon zamanı değerleri hesaplanır ve atanır. İlk anda hesaplanmış bir operasyon zamanı değeri olmayan uçakların, zaman atama fonksiyonu sonrasındaki operasyon zamanı değerleri ve var olan sıralamanın ceza maliyeti değeri aşağıdaki gibi olmaktadır.

Tablo 3.17: Tavlama Benzetiminde TO ve AO'ya göre uçak sıralama

Sıralama	0	1	2	3	4	5	6	7	8	9
TO	155	258	98	106	123	135	138	140	150	180
AO	155	258	273	281	289	297	305	313	321	329

Ceza maliyeti değeri: 40140

Algoritmanın başlangıcında genel en iyi ceza maliyeti değeri tutulur; genel en iyi ceza maliyeti değeri algoritmanın o ana kadar bulduğu en iyi değerdir.

Algoritmanın sıcaklık değişim döngüsü başlamadan önce sıcaklık değeri, başlangıç sıcaklık değeri ile güncellenir. Sıcaklık; algoritmanın sonlanmasını sağlayan iki adet kriterden ilkidir. Ayrıca algoritmanın sonlanma kriterlerinden bir diğeri olan daha iyi çözüm bulunamama sayacı da 0'a eşitlenir.

3.6.1. Tavlama Benzetimi Sıcaklık Değişim Döngüsü

Her döngüde üzerinde işlem yapılmak üzere baz olarak alınan bir sıralama vardır, ilk döngüde baz olarak alınan sıralama, algoritma ilk kez koştuğu için, başlangıç çözümüdür.

Döngünün sonuç sıralamasını bulmak için; rastgele seçilen iki uçak birbiri ile yer değiştirilerek (swap) yeni bir sıralama üretilmiş olunur. Daha sonra bu sıralamaya zaman ataması yapılması

için zaman atama fonksiyonuna gönderilir. Zaman ataması tamamlanan sıralamanın doğrulamasının yapılması için sıralama doğrulama fonksiyonuna gönderilir. Eğer bu sıralama geçerli bir sıralama ise ceza maliyeti değeri hesaplanır.

Sonraki aşamada ise yeni üretilen sıralamanın kabul edilip edilmeyeceğinin kontrolü yapılır. Bunun için ise aşağıdaki formül uygulanarak kabul olasılığı belirlenir:

$$P_T = \begin{cases} 1, & f(x_j) < f(x_i) \\ e^{\left(\frac{f(x_i)-f(x_j)}{T}\right)}, & f(x_j) \geq f(x_i) \end{cases}$$

P_T : $Sıralamanın\ kabul\ edilme\ olasılığı$

T : $O\ andaki\ sıcaklık\ değeri$

$f(x_i)$: $O\ andaki\ mevcut\ sıralamanın\ ceza\ maliyeti\ değeri$

$f(x_j)$: $Yeni$ ü$retilmiş\ olan\ sıralamanın\ ceza\ maliyeti\ değeri$

Kabul olasılığı değeri, rastgele üretilen 0 ile 1 arasındaki bir sayı ile karşılaştırılır ve bunun sonucunda, yeni sıralamanın kabul edilmesi durumunda ise; eğer aktif ise L1 ve L2 optimizasyonları da sırası ile uygulanır. En son durumda ise elde 1 adet sıralama olur, bu sıralama döngünün baz sıralaması olarak atanır.

Sıcaklık değeri değiştirilmeden önce, yani daha kötü çözümlerin kabul edilme ihtimali azaltılmadan önce rastgele yer değiştirme işlemleri parametre olarak belirtilen değer kadar uygulanır ve her seferinde elde bir adet sıralama olmuş olur. Sonrasında ise sıcaklık değeri düşürülerek aynı sıcaklık değerinde tekrar döngü içinde yer değiştirme işlemlerine devam edilmiş olunur.

Tüm döngüler bittikten sonra elde edilen en iyi sıralama ise aşağıdaki gibi olmaktadır;

3.6.2. Tavlama Benzetimi Sonuç

Sadece Tavlama Benzetimi:

Tablo 3.18: Tavlama benzetiminde TO ve AO'ya göre uçak sıralama ve ceza maliyeti

Sıralama	2	3	4	5	7	8	6	9	0	1
TO	98	106	123	135	140	150	138	180	155	258
AO	98	106	123	135	143	151	159	180	195	258
LP	30	30	30	30	30	30	30	30	10	10
EP	30	30	30	30	30	30	30	30	10	10
Ceza Etkisi	-	-	-	-	90	30	630	-	400	-

Ceza maliyeti değeri: 1150

L1 Aktif:

Tablo 3.19: L1 optimizasyonla-Tavlama benzetiminde TO ve AO'ya göre uçak sıralama ve ceza maliyeti.

Sıralama	2	3	4	5	6	7	8	0	9	1
TO	98	106	123	135	138	140	150	155	180	258
AO	98	106	118	126	134	142	150	165	180	258
LP	30	30	30	30	30	30	30	10	30	10
EP	30	30	30	30	30	30	30	10	30	10
Ceza Etkisi	-	-	150	270	120	60	-	100	-	-

Ceza maliyeti değeri: 700 (Optimum)

L2 Aktif:

Tablo 3.20: L2 optimizasyonla-Tavlama benzetiminde TO ve AO'ya göre uçak sıralama ve ceza maliyeti.

Sıralama	2	3	4	5	6	7	8	0	9	1
TO	98	106	123	135	138	140	150	155	180	258
AO	98	106	118	126	134	142	150	165	180	258
LP	30	30	30	30	30	30	30	10	30	10
EP	30	30	30	30	30	30	30	10	30	10
Ceza Etkisi	-	-	150	270	120	60	-	100	-	-

Ceza maliyeti değeri: 700 (Optimum)

3.7. GENETİK ALGORİTMA

S_0 (başlangıç çözümü, FCFS): {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

S_0, algoritma uygulanmadan önce havadaki görüntüdür, FCFS kuralını uygular, radara ilk giren uçağın ilk sırada olduğu, yani ilk iniş hakkını kazandığı sıralamadır.

Başlangıç çözümü algoritmaya sokulmadan önce Zaman Atama Fonksiyonu'na sokulur, ve sıralaması değiştirilmeden uçakların operasyon zamanı değerleri hesaplanır ve atanır. İlk anda hesaplanmış bir operasyon zamanı değeri olmayan uçakların, zaman atama fonksiyonu sonrasındaki operasyon zamanı değerleri ve var olan sıralamanın ceza maliyeti değeri aşağıdaki gibi olmaktadır.

Tablo 3.21: Genetik algoritma, TO ve AO'ya göre uçak sıralama.

Sıralama	0	1	2	3	4	5	6	7	8	9
TO	155	258	98	106	123	135	138	140	150	180
AO	155	258	273	281	289	297	305	313	321	329

Ceza maliyeti değeri: 40140

Algoritmanın başlangıcında genel en iyi ceza maliyeti değeri tutulur; genel en iyi ceza maliyeti değeri algoritmanın o ana kadar bulduğu en iyi değerdir.

Öncelikle aşağıdaki yöntem kullanılarak başlangıç popülasyonu oluşturulur;

- FCFS sıralama baz alınır.
- Başlangıç popülasyonunun her bir bireyini oluşturmak için, baz olarak alınan sıralamadaki uçak sayısı kadar yer değiştirme işlemi yapılır. Yani;
    - Baz sıralamada 5 adet uçak varsa, her yeni bireyi oluşturmak için toplamda 5 sefer rastgele 2 uçağın yeri değiştirilir (swap). Bu işlem sonucunda elde edilen yeni, rastgele, sıralama ise başlangıç popülasyonunun bir bireyi olur. Daha sonra bu yeni sıralama Zaman Atama Fonksiyonu'na gönderilir, sonrasında ise Sıralama Doğrulama Fonksiyonu yardımıyla geçerli bir sıralama olup olmadığı kontrol edilir. Yeni sıralama geçerli ise ceza değeri hesaplanır, geçerli olmaması durumunda ise ceza maliyeti değeri olarak DBL_MAX değeri, yani olası en yüksek değer kullanılır.
    - Bu işlem popülasyon sayısı kadar tekrarlanır, her bir yeni bireyi oluştururken baz olarak her seferinde aynı (FCFS) sıralama baz alınır.

3.7.1. Genetik Algoritma Döngüsü

Döngü, maksimum jenerasyon sayısı kadar tekrar eder. Elit birey (Elitism) parametresi aktif ise, elit birey sayısı kadar en iyi birey (sıralama) bir önceki jenerasyondan (döngüden) direkt olarak alınır ve bir sonraki jenerasyona aktarılır.

Popülasyon sayısını tamamlamak için eksik bireyler aşağıdaki yöntem ile bir önceki jenerasyon kullanılarak oluşturulur;

- Turnuva Seçimi (Tournament Selection) yöntemi ile iki adet ebeveyn seçilir,
- Çaprazlama (Crossover)işlemi ile bu iki ebeveyn kullanılarak 2 adet yeni sıralama (child) elde edilir,
- Yeni oluşturulan bu 2 yeni sıralamada mutasyon işlemi uygulanır,
- Popülasyon sayısı kadar birey elde edilene kadar işlem tekrarlanır.

L1 aktif ise; popülasyonun tamamı L1 Optimizasyon Fonksiyonu'na gönderilir, dolayısı ile ara adımda popülasyondaki birey sayısı artmış olur.

L2 aktif ise; L1 uygulanmış popülasyonun tamamı L2 Optimizasyon Fonksiyonu'na gönderilir.

Döngünün sonunda daha iyi ceza değeri olan bir sıralama bulunduysa en iyi sıralama (best cost) değeri güncellenir.

3.7.2. Turnuva Seçimi (Tournament Selection)

Parametre olarak belirtilen turnuva büyüklüğü (tournament size) değeri kadar sıralama (birey) popülasyondan rastgele seçilir ve bunların içinden en iyi sıralama bulunup fonksiyonun geri dönüş değeri olarak döndürülür.

3.7.3. Çaprazlama (Crossover)

Çaprazlama işlemine 2 adet ebeveyn sıralama girer ve sonuç olarak bu 2 ebeveynin çaprazlanmasından 2 adet çocuk sıralama üretilir. Çaprazlama işlemi için rastgele 2 çaprazlama noktası belirlenir, ebeveyn sıralamalarda bu iki nokta arasında kalan uçaklar yerinden oynatılmazken, diğer uçaklar iki ebeveynin çaprazlanması ile yer değiştirir. Sıralamada kullanılan çaprazlama işlemi diğer işlemlere göre biraz daha karmaşık uygulanır, daha net anlaşılması için bir örnek üzerinde açıklamak uygun olacaktır;

Örnekleme için kullanacağımız 2 adet ebeveyn sıralamanın aşağıda verildiği gibi olduğunu düşünelim:

Çaprazlama (Crossover) indeksleri: {2, 4} Not: İndeks 0'dan başlar ve uçakları simgeler, yani indeksin 2 olması, 3. sıradaki uçağı simgeler.

Tablo 3.22: Genetik algoritma çaprazlama işlemi.

İndeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Ebeveyn 1:	1	0	3	4	2	7	6	5
Ebeveyn 2:	3	4	1	5	7	6	0	2

Bu örnekte yeri değiştirilmeyecek olan uçaklar; 1. ebeveyn için {3, 4, 2} nolu uçaklar, 2. ebeveyn için ise {1, 5, 7} nolu uçaklardır.

İlk çocuğun (child) oluşturulması için ilk ebeveynin sabit kısımları aynen kopyalanır; ikinci ebeveynden sağdaki sabit noktadan başlanarak sırayla uçaklar kopyalanır. Kopyalanacak olan uçak eğer sabit aralıkta zaten var olan bir uçak ise bir sonraki uçak ile devam edilir:

Tablo 3.23: Genetik algoritma çaprazlama işlemi

İndeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Çocuk 1:			3	4	2			
Ebeveyn 2:	3	4	1	5	7	6	0	2

Çocuk 1'de mavi ile işaretli sıraya, ebeveyn 2'deki mavi ile işaretli, yani 6 nolu, uçak çaprazlama sonucunda yerleştirilecektir, bunun için öncelikle çocuk 1'de 6 nolu uçağın hali hazırda bulunup bulunmadığı kontrol edilir, bulunmaması durumunda yerleştirme işlemi yapılır ve çocuk 1 ve ebeveyn 2 için mavi indeks bir yana kaydırılır.

Tablo 3.24: Genetik algoritma çaprazlama işlemi

İndeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Çocuk 1:			3	4	2	6		
Ebeveyn 2:	3	4	1	5	7	6	0	2

Bu adımda da 0 nolu uçağın çocuk 1'e çaprazlanması için herhangi bir engel olmadığı için, 0 nolu uçak çocuk 1'de mavi ile işaretli indekse yerleştirilir ve mavi indeksler birer yana kaydırılır.

Tablo 3.25: Genetik algoritma çaprazlama işlemi

İndeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Çocuk 1:			3	4	2	6	0	
Ebeveyn 2:	3	4	1	5	7	6	0	2

Sıra 2 nolu uçağın ebeveynden çocuğa (child) çaprazlanmasına geldiğinde ise bu işlem yapılamaz, çünkü 2 nolu uçak zaten çocuk 1'de sabit aralıkta, yani {2, 4} indeksleri arasında yer almaktadır. Bu nedenle çocuk 1'deki mavi indeks sabit tutulurken, ebeveyn 2'deki mavi indeks bir yana, yani bu örnekte sıralamanın başına, kaydırılarak bir sonraki uçağın çaprazlanmaya uygun olup olmadığı kontrol edilir.

Tablo 3.26: Genetik algoritma çaprazlama işlemi

İndeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Çocuk 1:			3	4	2	6	0	
Ebeveyn 2:	3	4	1	5	7	6	0	2

Mavi indeksin ebeveyn 2'de üzerinde yer aldığı 3 nolu uçak da çocuk 1'in sabit aralığında yer alan uçaklar arasında olması nedeniyle çaprazlama işlemi yine gerçekleştirilemez ve mavi indeks çocuk 1 üzerinde yine sabit kalırken, ebeveyn 2 üzerindeki yeri bir kez daha bir yana kaydırılarak sonraki uçağa geçilir.

Tablo 3.27: Genetik algoritma çaprazlama işlemi

İndeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Çocuk 1:			3	4	2	6	0	
Ebeveyn 2:	3	4	1	5	7	6	0	2

Mavi indeksin ebeveyn 2'de üzerinde yer aldığı 4 nolu uçak da çocuk 1'in sabit aralığında yer alan uçaklar arasında olması nedeniyle çaprazlama işlemi yine gerçekleştirilemez ve mavi indeks çocuk 1 üzerinde yine sabit kalırken, ebeveyn 2 üzerindeki yeri bir kez daha bir yana kaydırılarak sonraki uçağa geçilir.

Tablo 3.28: Genetik algoritma çaprazlama işlemi

İndeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Çocuk 1:			3	4	2	6	0	
Ebeveyn 2:	3	4	1	5	7	6	0	2

Görüldüğü gibi mavi indeksin ebeveyn 2 üzerindeki yeri sabit aralığa kadar gelmiştir, ancak mavi indeksin ebeveyn 2 üzerindeki sabit aralığa gelmesinin bir önemi yoktur, çünkü çaprazlama ile çocuk 1 elde edilmeye çalışılmakta ve sabit aralığın bozulamayacağı sıralama çocuk 1 sıralamasıdır. Yine mavi indeksin ebeveyn 2 üzerindeki uçağın çaprazlama için uygun olup olmadığının kontrolü yapılır. Ebeveyn 2 üzerindeki 1 nolu uçak çaprazlama için engel teşkil etmez ve çaprazlama işlemi, ebeveyn 2 üzerinde mavi indeksin gösterdiği 1 nolu uçağın çocuk 1 üzerindeki mavi indeksin gösterdiği yere taşınması ile gerçekleştirilir. Sonrasında ise

mavi indeks çocuk 1 üzerinde ve ebeveyn 2 üzerinde bir yana kaydırılır. Çocuk 1 üzerindeki mavi indeks sıralamanın sonunda yer aldığı için sıralamanın başına kaydırılır.

Tablo 3.29: Genetik algoritma çaprazlama işlemi.

İndeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Çocuk 1:			3	4	2	6	0	1
Ebeveyn 2:	3	4	1	5	7	6	0	2

Bu adımda da 5 nolu uçağın çocuk 1'e çaprazlanması için herhangi bir engel olmadığı için, 5 nolu uçak çocuk 1'de mavi ile işaretli indekse yerleştirilir ve mavi indeksler birer yana kaydırılır.

Tablo 3.30: Genetik algoritma çaprazlama işlemi.

İndeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Çocuk 1:	5		3	4	2	6	0	1
Ebeveyn 2:	3	4	1	5	7	6	0	2

Bu adımda da 7 nolu uçağın çocuk 1'e çaprazlanması için herhangi bir engel olmadığı için, 7 nolu uçak çocuk 1'de mavi ile işaretli indekse yerleştirilir ve böylelikle çocuk 1 oluşturulmuş olur, çaprazlama işlemi bu şekilde son bulur.

Tablo 3.31: Genetik algoritma çaprazlama işlemi

İndeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Çocuk 1:	5	7	3	4	2	6	0	1
Ebeveyn 2:	3	4	1	5	7	6	0	2

Çocuk 2 oluşturulması için mantık aynı şekildedir, tek farkı ise ebeveyn 2 yerine ebeveyn 1'in kullanılmasıdır.

3.7.4. Mutasyon

Parametre olarak gönderilen sıralamadaki tüm uçaklar, ilk uçaktan sonuncu uçağa kadar, 0 ile 1 arasında rastgele üretilen bir değerin parametre olarak belirlenen mutasyonu oranından (mutation rate) küçük olması durumunda; rastgele seçilen diğer bir uçak ile yer değiştirir, yani yer değiştirme işlemi yapılır.

3.7.5. Genetik Algoritma Sonuç

Sadece Genetik Algoritma:

Tablo 3.32: Genetik algoritmada TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti

Sıralama	2	3	4	5	7	8	6	9	0	1
TO	98	106	123	135	140	150	138	180	155	258
AO	98	106	123	135	143	151	159	180	195	258
LP	30	30	30	30	30	30	30	30	10	10
EP	30	30	30	30	30	30	30	30	10	10
Ceza Etkisi	-	-	-	-	90	30	630	-	400	-

Ceza maliyeti değeri: 1150

L1 Aktif:

Tablo 3.33: L1 optimizasyona- Genetik algoritmada TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti

Sıralama	2	3	4	5	6	7	8	0	9	1
TO	98	106	123	135	138	140	150	155	180	258
AO	98	106	118	126	134	142	150	165	180	258
LP	30	30	30	30	30	30	30	10	30	10
EP	30	30	30	30	30	30	30	10	30	10
Ceza Etkisi	-	-	150	270	120	60	-	100	-	-

Ceza maliyeti değeri: 700 (Optimum)

L2 Aktif:

Tablo 3.34: L2 optimizasyonal-Genetik algoritmada TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti

Sıralama	2	3	4	5	6	7	8	0	9	1
TO	98	106	123	135	138	140	150	155	180	258
AO	98	106	118	126	134	142	150	165	180	258
LP	30	30	30	30	30	30	30	10	30	10
EP	30	30	30	30	30	30	30	10	30	10
Ceza Etkisi	-	-	150	270	120	60	-	100	-	-

Ceza maliyeti değeri: 700 (Optimum)

3.8. KARINCA KOLONİSİ OPTİMİZASYONU ALGORİTMASI

Algoritmada eşik (edge) olarak adlandırılan; sıralamada yer alan ve birbirini takip eden uçak ikilileri kullanılmaktadır, bunların arasında öncü- takip eden (leading – following) ilişkisi vardır. Örneklemek gerekirse;

{0, 3, 1, 2} örnek sıralamasındaki eşik (edge) oluşturan ikililer aşağıdaki gibidir:

{∅, 0} $\Rightarrow$ İlk sırada ∅ olması, ikinci sıradaki, yani 0 numaralı uçağın sıralamada ilk sırada yer aldığını, ondan önce başka uçak sıralanmadığını belirtir.

{0, 3} $\Rightarrow$ 0 nolu uçak önde, 3 nolu uçak onu takip ediyor.

{3, 1} $\Rightarrow$ 3 nolu uçak önde, 1 nolu uçak onu takip ediyor.

{1, 2} $\Rightarrow$ 1 nolu uçak önde, 2 nolu uçak onu takip ediyor.

{2, ∅} $\Rightarrow$ İkinci uçağın ∅ olması, ilk sıradaki, yani 2 nolu uçağın sıralamada son sırada olduğunu belirtir.

S_0 (başlangıç çözümü, FCFS): {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

S_0, algoritma uygulanmadan önce havadaki görüntüdür, FCFS kuralını uygular, radara ilk giren uçağın ilk sırada olduğu, yani ilk iniş hakkını kazandığı sıralamadır.

Başlangıç çözümü algoritmaya sokulmadan önce Zaman Atama Fonksiyonu'na sokulur, ve sıralaması değiştirilmeden uçakların operasyonu zamanı (operation time) değerleri hesaplanır ve atanır. İlk anda hesaplanmış bir operasyonu zamanı (operation time) değeri olmayan uçakların, zaman atama fonksiyonu sonrasındaki operasyon zamanı değerleri ve var olan sıralamanın ceza maliyeti değeri aşağıdaki gibi olmaktadır.

Tablo 3.35: Karınca kolonisi optimizasyonunda TO ve AO'ya göre uçak sıralama.

Sıralama	0	1	2	3	4	5	6	7	8	9
TO	155	258	98	106	123	135	138	140	150	180
AO	155	258	273	281	289	297	305	313	321	329

Ceza maliyeti değeri: 40140

Algoritmanın başlangıcında genel en iyi ceza maliyeti değeri tutulur; genel en iyi ceza maliyeti değeri algoritmanın o ana kadar bulduğu en iyi değerdir.

Algoritmada toplamda, parametre ile belirtilen, maksimum döngü sayısı kadar döngü döner. Her döngüde karınca sayısı kadar yeni sıralama oluşturulur, yani her karınca bir sıralama üretir. Sonrasında üretilen bu sıralamalara ceza maliyeti değerleri ile ters orantılı olarak feromon eklenir. Her döngünün sonunda tüm sıralamalardaki feromon miktarı belirli oranda buharlaştırma işlemine tabi tutulur.

3.8.1. Karınca Kolonisi Algoritması Döngüsü

Sıralama Üretme Fonksiyonu ile kolonideki karınca sayısı kadar yeni sıralama üretilir. Geçerli olan sıralamalar için ceza değeri hesaplanır.

L1 aktif ise; popülasyonun tamamı L1 Optimizasyon Fonksiyonu'na gönderilir, ancak en iyi sıralama bulunarak işlemlere bu sıralama ile devam edilir.

L2 aktif ise; L1 adımında bulunan en iyi sıralama L2 Optimizasyon Fonksiyonu'na gönderilir. Her aşamada sonuç olarak sadece 1 adet sıralama elde edilmiş olur.

Elde edilen sıralamaya, Feromon Ekleme Fonksiyonu kullanılarak feromon değerleri eklenir. Sıralamaya feromon eklenirken; daha önce belirtilen eşik (edge) , yani uçak ikilisi mantığı kullanılır. Tüm karıncalar kendi sıralamalarını ürettikten, diğer bir deyişle karınca sayısı kadar sıralama üretildikten sonra "Feromon Buharlaştırma Fonksiyonu" ile tüm uçak ikililerindeki feromon değerleri azaltılır.

3.8.2. Sıralama Üretme Fonksiyonu

Yeni sıralama üretilirken Rulet Tekerleği (Roulette Wheel) yöntemi kullanılır. Tüm uçak ikililerinin feromon değerleri ve sezgisel değerine etkileri, parametre olarak belirtilen katsayılar kullanılarak hesaba katılır. Feromon değerinin etkisinin ne kadar olacağı ve sezgisel değerinin etkisinin ne kadar olacağı parametre olarak verilen bu değerler ile belirlenir.

Feromon değeri hesaba katılırken, daha önce hesaplanan ve uçak ikilileri için oluşturulan değerler kullanılır. Sezgisel değeri ise, literatürde de örnekleri olan aşağıdaki yöntem ile hesaplanır;

Eğer sıralamada önde yer alan uçağın Hedeflenen Operasyon Zamanı (Target Operation Time) ile, takip eden uçağın TO zamanlamaları arasındaki süre, bu iki uçak arasında yer alması gereken minimum zamana (ayırım zamanı da dahil) eşit veya daha fazla ise, geri döndürülen değer 1 olur. Bu durum, iki uçağın bu sıralamayı ikisinden birisi ceza değeri almadan sağlayabileceği anlamına gelmektedir. Aksi durumda, yani aralarındaki süre ceza almalarını engelleyecek kadar değilse, aralarındaki süreyle ters orantılı olarak değer geri döndürülür.

Örneğin;

Sıralama oluşturulurken ilk uçağın seçimi için, uçak ikililerinden öndeki uçak ∅ olanlara bakılır, çünkü ilk elemanın ∅ olması, o ikilide yer alan ikinci uçağın ilk sırada yer alma durumundaki feromon değerini tutmaktadır. Yani {∅, 0} ikilisi, ilk sırada 0 nolu uçağın yer alma olasılığını belirtecektir, aynı şekilde {∅, 1} de 1 nolu uçağın sıralamada ilk sıradaki uçak olma olasılığını belirtmektedir.

Bu şekilde tüm {∅, ...} ikilileri ile Rulet Tekerleği (Roulette Wheel) oluşturulur, daha sonra da Rulet Tekerleği (Roulette Wheel) mantığı kullanılarak üretilen rastgele değere göre ikililerden bir tanesi, dolayısıyla sıralamadaki ilk uçak, seçilmiş olur.

Sıralamadaki tüm uçaklar belirlenene kadar Rulet Tekerleği (Roulette Wheel) uygulanır; Örneğin ilk sıra için seçilen uçağın, 3 nolu uçak olması durumunda, {3, ...} ikilileri kullanılarak Rulet Tekerleği (Roulette Wheel) oluşturulur ve böylelikle sıralamada 3 nolu uçaktan sonra gelecek olan uçak belirlenmiş olur.

3.8.3. Feromon Ekleme Fonksiyonu

Feromon eklenecek olan uçak ikilisi eğer yeni oluşturulmuş ve üzerinde hiç feromon olmayan bir ikili ise, parametre olarak belirtilen ilk feromon değeri atanır. Daha önce üzerinde feromon olan bir ikili olması durumunda da ise; parametre olarak belirtilen Q değerinin uçak ikilisinin kullanıldığı sıralamanın toplam ceza maliyeti değerine bölünmesi ile bulunan değerin, var olan feromon değerinin üzerine eklenmesi, yani toplanması, ile yeni feromon değeri bulunmuş olur.

Aynı uçak ikilisi için, aynı döngü içinde birden fazla feromon ekleme işlemi gerçekleştirilebilir, bu durum uçak ikilisinin birden fazla sıralamada kullanılması ile ilgilidir.

3.8.4. Feromon Buharlaştırma Fonksiyonu

Feromon buharlaştırma işleminde, tüm uçak ikililerindeki feromon değerleri, algoritmaya parametre olarak geçilen rho (ρ) değeri oranında azaltılır;

$$\tau_{ij} = (1 - \rho) \times \tau_{ij}$$

τ_{ij}: $i\ ve\ j\ uçak\ ikilisi\ için\ atanan\ feromon\ değeri$

ρ: $Rho\ katsayısı$

3.8.5. Karınca Kolonisi Optimizasyonu Sonuç

Sadece Karınca Kolonisi Optimizasyonu:

Tablo 3.36: Karınca kolonisi optimizasyonunda TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti

Sıralama	2	3	4	5	7	8	6	9	0	1
TO	98	106	123	135	140	150	138	180	155	258
AO	98	106	123	135	143	151	159	180	195	258
LP	30	30	30	30	30	30	30	30	10	10
EP	30	30	30	30	30	30	30	30	10	10
Ceza Etkisi	-	-	-	-	90	30	630	-	400	-

Ceza maliyeti değeri: 1150

L1 Aktif:

Tablo 3.37: L1 optimizasyona- Karınca kolonisi optimizasyonunda TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti

Sıralama	2	3	4	5	6	7	8	0	9	1
TO	98	106	123	135	138	140	150	155	180	258
AO	98	106	118	126	134	142	150	165	180	258
LP	30	30	30	30	30	30	30	10	30	10
EP	30	30	30	30	30	30	30	10	30	10
Ceza Etkisi	-	-	150	270	120	60	-	100	-	-

Ceza maliyeti değeri: 700 (Optimum)

L2 Aktif:

Tablo 3.38: L2 optimizasyonla-Karınca kolonisi optimizasyonunda TO, AO, LP, EP'ye göre uçak sıralama ve ceza maliyeti.

Sıralama	2	3	4	5	6	7	8	0	9	1
TO	98	106	123	135	138	140	150	155	180	258
AO	98	106	118	126	134	142	150	165	180	258
LP	30	30	30	30	30	30	30	10	30	10
EP	30	30	30	30	30	30	30	10	30	10
Ceza Etkisi	-	-	150	270	120	60	-	100	-	-

Ceza maliyeti değeri: 700 (Optimum)

4. BULGULAR

Geliştirilen algoritmaların C programında kodlaması yapılmıştır. Tüm algoritmalar, 2.70GHz Intel Core i5 işlemcili ve 8.00 GB RAM.li, macOS 10.14.6 bilgisayarda çalıştırılmıştır.

Çalışmada kullanılan optimize edilmiş algoritma sonuçları; sadece statik değil aynı zamanda dinamik durum için olmasına rağmen OR kütüphanesindeki küçük problemlerde Besley ve diğerlerinin (1, 2) statik durum için gelen uçakların çizelgelenmesinde elde ettiği sonuçlardan genel olarak daha iyi sonuç vermiştir.

Optimize edilmiş algoritmaların üçü de optimize edilmiş Karınca Kolonisi Algoritması hariç; optimum değeri bulmada etkili algoritmalar olmuştur. Parametere ayarına devam edilerek Karınca Kolonisi Algoritması'nın da performansı arttırılabilir. Özellikle dinamik durum için büyük test problemlerinde Beasley, Pinol, (3) ve Awasthi , Kramer , Lassig (4) tarafından bahsedilen bilinen en iyi çözüm değerine ulaşmak ya da yaklaşmak için parametre ayarına devam edilerek çözüm değeri iyileştirelebilir.

Optimizasyon yapılan algoritmaların kendi içerisinde kıyaslama sonuçlarına göre; algoritmalar statik durumda çalıştırıldığında 2. Seviye optimizasyon uygulanan (L2) Tavlama Benzetimi Algoritması daha iyi sonuç verirken, algoritmalar dinamik durumda çalıştırıldığında 2. Seviye optimizasyon uygulanan (L2) Tabu Arama Algoritması daha iyi sonuç vermiştir.

Tablo 4.1.e göre, L1 (1.Seviye optimizasyon) uygulanmış 4 metasezgisel algoritma OR library'deki küçük problemlerde kullanılan 8 veri seti için dinamik durumda olmasına ragmen, Beasley'in statik durum için elde ettiği sonuçlardan bazı veri setleri için daha iyi sonuç elde etmiş ve çoğunlukla optimum sonuca ulaşmıştır. Ayrıca standart sapma değerlerine ve en iyi değere (best cost ratio) ulaşma oranlarına bakılarak optimize edilmiş karınca kolonisi hariç optimize edilmiş Tabu Arama, Tavlama Benzetimi, Genetik Algoritma sonuçlarının optimum sonuçlara ulaşmada oldukça başarılı olduğu söylenebilir.

Tablo 4.2'ye göre, dinamik durum için L1 (1.Seviye optimizasyon) uygulanmış 4 metasezgisel algoritma OR library'deki büyük problemlerde kullanılan 5 veri seti için Beasley'in statik

durumda elde ettiği kadar iyi sonuç elde edemese de yakın değerler elde etmiştir. Parametre ayarı yapılarak çözüm sonuçları iyileştirilebilir.

Tablo 4.3.e göre, L2 (2.Seviye optimizasyon) uygulanmış 4 metasezgisel algoritma OR library'deki küçük problemlerde kullanılan 8 veri seti için dinamik durumda olmasına ragmen, Beasley'in statik durum için elde ettiği sonuçlardan bazı veri setleri için daha iyi sonuç elde etmiş ve çoğunlukla optimum sonuca ulaşmıştır. Ayrıca standart sapma değerlerine ve en iyi değere (best cost ratio) ulaşma oranlarına bakılarak optimize edilmiş karınca kolonisi hariç optimize edilmiş Tabu Arama, Tavlama Benzetimi, Genetik Algoritma sonuçlarının optimum sonuçlara ulaşmada oldukça başarılı olduğu söylenebilir.

Tablo 4.4'e göre, dinamik durumda L2 (2.Seviye optimizasyon) uygulanmış 4 metasezgisel algoritma OR library'deki büyük problemlerde kullanılan 5 veri seti için Besley'in statik durumda elde ettiği kadar iyi sonuç elde edemese de yakın değerler elde etmiştir. Parametre ayarı yapılarak çözüm sonuçları iyileştirilebilir.

Tablo 4.5'e göre OR Library'deki 13 Veri seti için Beasley'in statik durum için yazılan makalesi ve en iyi değer ya da bilinen en iyi değer sonuçlarıyla kıyaslamaya göre 2.Seviye Optimizasyon (L2) uygulaması, 1. Seviye optimizasyon (L1) uygulamasına göre genel olarak daha iyi sonuç vermiştir.

Tablo 4.6'ya göre, L1 (1.Seviye optimizasyon) uygulanmış 4 metasezgisel algoritma OR library'deki küçük problemlerde kullanılan 8 veri seti için, Beasley'in elde ettiği sonuçlardan bazı veri setleri için daha iyi sonuç elde etmiş ve optimize edilmiş Karınca kolonisi hariç genellikle optimum sonuca ulaşmıştır. Parametre ayarı yapılarak sonuçlar daha da fazla iyileştirilebilir. Bulunan en iyi değerlere, standart sapmaya ve en iyi değere ulaşma oranlarını bakılarak 1.seviye optimizasyon (L1) yapılan Tabu Arama algoritmasının daha iyi sonuç verdiği söylenebilir.

Tablo 4.7'ye göre, 2.seviye optimizasyon (L2) uygulanmış 4 metasezgisel algoritma OR library'deki küçük problemlerde kullanılan 8 veri seti için, Beasley'in elde ettiği sonuçlardan daha iyi daha iyi sonuç elde etmiş ve veri setlerinden biri hariç optimum sonuca ulaşmıştır. Ayrıca standart sapma değerlerine ve en iyi değere (best cost ratio) ulaşma oranlarına bakılarak

optimize edilmiş karınca kolonisi hariç optimize edilmiş Tabu Arama, Tavlama Benzetimi, Genetik Algoritma sonuçlarının optimum sonuçlara ulaşmada oldukça başarılı olduğu söylenebilir. Bulunan en iyi değerlere bakılarak 2. Seviye optimizasyon yapılan Tavlama Benzetimi algoritmasının daha iyi sonuç verdiği söylenebilir.

Tablo 4.8.'e göre, OR Library'deki 8 Veri seti için Beasley'in statik durum için yazılan makalesi ve en iyi değer ya da bilinen en iyi değer sonuçlarıyla kıyaslamaya göre 2.Seviye Optimizasyon (L2) uygulaması, 1. Seviye optimizasyon (L1) uygulamasına göre genel olarak daha iyi sonuç vermiştir. Sonuçlara, standart sapmaya ve en iyi değere ulaşma oranlarına bakarak 2. Seviye (L2) optimizasyon uygulanan Tabu Arama Algoritmasının statik durumda daha iyi sonuç verdiği görülmektedir.

Küçük ve büyük problem olmalarına bağlı olarak; farklı uçak sayıları (500'e kadar) ve 13 veri seti için, statik, dinamik duruma göre karşılaştırma tabloları aşağıdaki gibidir:

Tablo 4.1: Dinamik L1 Optimizasyonu Uygulanan OR Kütüphanesi Küçük Problemler Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Dinamik)

				Tabu Arama - L1				Tavlama Benzetimi - L1				Genetik Algoritma - L1				Karınca Kolonisi Optimizationu - L1			
Veri Seti	Uçak Sayısı	En İyi Çözüm Değeri	Statik Beasley	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı
1	10	700	700	700	700	0	100%	700	700	0	100%	700	700	0	100%	700	700	0	100%
2	15	1480	1500	1510	1510	0	100%	1510	1510	0	100%	1510	1510	0	100%	1510	1510	0	100%
3	20	820	1380	820	820	0	100%	820	820	0	100%	820	820	0	100%	820	831.9	65.11	89%
4	20	2520	2520	3040	3040	0	100%	2980	3045.1	22.98	1%	3040	3043.9	20.74	96%	2980	3232.9	159.27	23%
5	20	3100	5420	3810	3810	0	100%	3810	3810	0	100%	3810	3810	0	100%	3810	3810	0	100%
6	30	24442	24442	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%
7	44	1550	1550	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%
8	50	1950	2690	2130	2130	0	100%	2115	2160.4	87.67	13%	2115	2138.25	18.82	3%	2640	3554.55	322.40	1%

Tablo 4.2: Dinamik L1 Optimizasyonu Uygulanan OR Kütüphanesi Büyük Problemler Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Dinamik)

				Tabu Arama - L1				Tavlama Benzetimi - L1				Genetik Algoritma - L1				Karınca Kolonisi Optimizationu - L1			
Veri Seti	Uçak Sayısı	Z best (Bilinen En İyi Çözüm)	Dağınık Arama / Bionomik Algoritma Beasley	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı
9	100	5611.7	7298.57 / 6425.95	9530.49	9530.49	0.00	100%	9530.49	9535.25	16.21	92%	9530.49	9587.03	73.88	42%	11301.38	12213.53	332.49	1%
10	150	12329.31	17872.56 / 16508.94	20864.13	20864.13	0.00	100%	20013.56	20213.08	182.46	18%	20062.44	20424.08	271.58	2%	27089.39	28565.37	634.37	1%
11	200	12418.32	14647.40 / 14488.45	18671.22	18671.22	0.00	100%	18671.22	18671.22	0.00	100%	18671.22	18728.31	80.53	65%	22095.52	22448.50	142.98	1%
12	250	16209.78	19800.24 / 20032.04	23605.23	23605.23	0.00	100%	23605.23	23626.74	51.67	81%	23605.23	23786.27	87.13	7%	28489.22	29061.93	387.79	1%
13	500	44832.38	46284.84 /45294.15	58973.56	58973.56	0.00	100%	58673.10	58725.00	71.02	1%	57869.68	58841.92	233.06	1%	67512.38	69705.10	1137.82	1%

Tablo 4.3: Dinamik L2 Optimizasyonu Uygulanan OR Kütüphanesi Küçük Problemler Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Dinamik)

				Tabu Arama - L2				Tavlama Benzetimi - L2				Genetik Algoritma - L2				Karınca Kolonisi - L2			
Veri Seti	Uçak Sayısı	En İyi Çözüm Değeri	Statik Beasley	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı
1	10	700	700	700	700	0	100%	700	700	0	100%	700	700	0	100%	700	700	0	100%
2	15	1480	1500	1480	1480	0	100%	1480	1484.2	22.57	96%	1480	1482	8.76	95%	1480	1480	0	100%
3	20	820	1380	820	820	0	100%	820	820	0	100%	820	820	0	100%	820	857.8	127.73	83%
4	20	2520	2520	2840	2840	0	100%	2840	2840	0	100%	2840	2840	0	100%	2840	3192.8	204.00	21%
5	20	3100	5420	3620	3620	0	100%	3620	3620	0	100%	3620	3620	0	100%	3620	3620	0	100%
6	30	24442	24442	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%
7	44	1550	1550	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%
8	50	1950	2690	2100	2100	0	100%	1980	2067.8	67.68	7%	1970	2056.45	49.45	1%	2580	3447.6	306.93	1%

Tablo 4.4: Dinamik L2 – Optimizasyonu Uygulanan OR Kütüphanesi Büyük Problemleri Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Dinamik)

				Tabu Arama - L2				Tavlama Benzetimi - L2				Genetik Algoritma - L2				Karınca Kolonisi - L2			
Veri Seti	Uçak Sayısı	Z best (Bilinen En İyi Çözüm)	Dağınık Arama / Bionomik Algoritma Beasley	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı
9	100	5611.7	7298.57 / 6425.95	9145.77	9145.77	0.00	100%	9025.50	9137.81	33.70	8%	9145.77	9161.30	38.80	81%	10999.43	11489.56	350.80	1%
10	150	12329.31	17872.56 / 16508.94	19517.19	19517.19	0.00	100%	19517.19	19649.02	105.21	31%	19517.19	19757.48	110.79	5%	25551.84	27064.07	609.52	1%
11	200	12418.32	14647.40 / 14488.45	18464.77	18464.77	0.00	100%	18464.77	18464.77	0.00	100%	18464.77	18631.03	52.63	7%	21688.78	22227.45	183.20	1%
12	250	16209.78	19800.24 / 20032.04	22997.90	22997.90	0.00	100%	22808.11	22830.23	57.69	82%	22808.11	22973.36	80.14	5%	27318.20	28260.71	382.70	1%
13	500	44832.38	46284.84 /45294.15	56950.96	56950.96	0.00	100%	56950.96	56985.73	35.27	19%	56827.07	56985.39	57.94	1%	66461.77	67747.69	631.75	1%

Tablo 4.5: Dinamik L1 – L2 Optimizasyonu Uygulanan OR Kütüphanesi Küçük-Büyük Problemler Karşılaştırma Tablosu.(Beasley-Statik, Geliştirilen Algoritma Dinamik)

					Tabu Arama				Tavlama Benzetimi				Genetik Algoritma				Karınca Kolonisi Optimizasyonu			
Veri Seti	Uçak Sayısı	En İyi Değer / Bilinen En İyi Değer (Zbest)	Static Beasley	Optimizasyon Seviyesi	En İyi Değer	Ortalama	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama	Std Sapma	En İyi Değer Oranı
1	10	700	700	L1	700	700	0	100%	700	700	0	100%	700	700	0	100%	700	700	0	100%
				L2	700	700	0	100%	700	700	0	100%	700	700	0	100%	700	700	0	100%
2	15	1480	1500	L1	1510	1510	0	100%	1510	1510	0	100%	1510	1510	0	100%	1510	1510	0	100%
				L2	1480	1480	0	100%	1480	1484.2	22.57	96%	1480	1482	8.76	95%	1480	1480	0	100%
3	20	820	1380	L1	820	820	0	100%	820	820	0	100%	820	820	0	100%	820	831.9	65.11	89%
				L2	820	820	0	100%	820	820	0	100%	820	820	0	100%	820	857.8	127.73	83%
4	20	2520	2520	L1	3040	3040	0	100%	2980	3045.1	22.98	1%	3040	3043.9	20.74	96%	2980	3232.9	159.27	23%
				L2	2840	2840	0	100%	2840	2840	0	100%	2840	2840	0	100%	2840	3192.8	204.00	21%
5	20	3100	5420	L1	3810	3810	0	100%	3810	3810	0	100%	3810	3810	0	100%	3810	3810	0	100%
				L2	3620	3620	0	100%	3620	3620	0	100%	3620	3620	0	100%	3620	3620	0	100%
6	30	24442	24442	L1	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%
				L2	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%
7	44	1550	1550	L1	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%
				L2	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%
8	50	1950	2690	L1	2130	2130	0	100%	2115	2160.4	87.67	13%	2115	2138.25	18.82	3%	2640	3554.55	322.40	1%
				L2	2100	2100	0	100%	1980	2067.8	67.68	7%	1970	2056.45	49.45	1%	2580	3447.60	306.93	1%
9	100	5611.7	7298.57 / 6425.95	L1	9530.49	9530.49	0	100%	9530.49	9535.25	16.21	92%	9530.49	9587.03	73.88	42%	11301.38	12213.53	332.49	1%
				L2	9145.77	9145.77	0	100%	9025.5	9137.81	33.70	8%	9145.77	9161.30	38.80	81%	10999.43	11489.56	350.80	1%
10	150	12329.31	17872.56 / 16508.94	L1	20864.13	20864.13	0	100%	20013.56	20213.08	182.46	18%	20062.44	20424.08	271.58	2%	27089.39	28565.37	634.37	1%
				L2	19517.19	19517.19	0	100%	19517.19	19649.02	105.21	31%	19517.19	19757.48	110.79	5%	25551.84	27064.07	609.52	1%
11	200	12418.32	14647.40 / 14488.45	L1	18671.22	18671.22	0	100%	18671.22	18671.22	0	100%	18671.22	18728.31	80.53	65%	22095.52	22448.50	142.98	1%
				L2	18464.77	18464.77	0	100%	18464.77	18464.77	0	100%	18464.77	18631.03	52.63	7%	21688.78	22227.45	183.20	1%
12	250	16209.78	19800.24 / 20032.04	L1	23605.23	23605.23	0	100%	23605.23	23626.74	51.67	81%	23605.23	23786.27	87.13	7%	28489.22	29061.93	387.79	1%
				L2	22997.9	22997.9	0	100%	22808.11	22830.23	57.69	82%	22808.11	22973.36	80.14	5%	27318.2	28260.71	382.70	1%
13	500	44832.38	46284.84 /45294.15	L1	58973.56	58973.56	0	100%	58673.1	58725.00	71.02	1%	57869.68	58841.92	233.06	1%	67512.38	69705.10	1137.82	1%
				L2	56950.96	56950.96	0	100%	56950.96	56985.73	35.27	19%	56827.07	56985.39	57.94	1%	66461.77	67747.69	631.75	1%

Tablo 4.6: Statik L1 Optimizasyonu Uygulanan OR Kütüphanesi Küçük Problemler Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Statik)

				Tabu Arama - L1				Tavlama Benzetimi - L1				Genetik Algoritma - L1				Karınca Kolonisi Optimizationu - L1			
Veri Seti	Uçak Sayısı	En İyi Çözüm Değeri	Statik Beasley	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı
1	10	700	700	700	700	0	100%	700	700	0	100%	700	700	0	100%	700	700	0	100%
2	15	1480	1500	1510	1510	0	100%	1510	1510	0	100%	1510	1581.8	94.81	59%	2030	2030	0	100%
3	20	820	1380	820	820	0	100%	820	820	0	100%	820	885.8	177.05	84%	2290	2290	0	100%
4	20	2520	2520	2940	2940	0	100%	2940	2966.4	45.65	66%	2940	3145.5	127.35	13%	4480	4480	0	100%
5	20	3100	5420	3520	3520	0	100%	3520	3638.5	114.63	47%	3520	4058.3	302.79	1%	7120	7120	0	100%
6	30	24442	24442	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%
7	44	1550	1550	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%
8	50	1950	2690	2600	2600	0	100%	2600	2781.35	169.94	14%	2910	3583.25	215.61	2%	4390	4390	0	100%

Tablo 4.7: Statik L2 Optimizasyonu Uygulanan OR Kütüphanesi Küçük Problemler Karşılaştırma Tablosu. (Beasley-Statik, Geliştirilen Algoritma Statik)

				Tabu Arama - L2				Tavlama Benzetimi - L2				Genetik Algoritma - L2				Karınca Kolonisi Optimizationu - L2			
Veri Seti	Uçak Sayısı	En İyi Çözüm Değeri	Statik Beasley	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama Değer	Std Sapma	En İyi Değer Oranı
1	10	700	700	700	700	0	100%	700	700	0	100%	700	700	0	100%	700	700	0	100%
2	15	1480	1500	1480	1480	0	100%	1480	1484.8	8.58	76%	1480	1486.8	9.52	66%	1480	2001.6	117.86	3%
3	20	820	1380	820	820	0	100%	820	820.6	6	99%	820	903.9	189.46	79%	1710	2259.5	110.72	2%
4	20	2520	2520	2520	2520	0	100%	2520	2520	0	100%	2520	2520	0	100%	4480	4480	0	100%
5	20	3100	5420	3100	3100	0	100%	3100	3100	0	100%	3100	3492	296.50	25%	7120	7120	0	100%
6	30	24442	24442	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%
7	44	1550	1550	3974	3974	0	100%	1550	1550	0	100%	1550	1550	0	100%	3974	3974	0	100%
8	50	1950	2690	2170	2170	0	100%	2170	2313.8	132.44	17%	2455	2870.9	200.58	1%	4390	4390	0	100%

Tablo 4.8: Statik L1 – L2 Optimizasyonu Uygulanan OR Kütüphanesi Küçük Problemler Karşılaştırma Tablosu (Beasley-Statik, Geliştirilen Algoritma Statik)

					Tabu Arama L1-L2				Tavlama Benzetimi L1-L2				Genetik Algoritma L1-L2				Karınca Kolonisi Optimizasyonu L1-L2			
Veri Seti	Uçak Sayısı	En İyi Değer / Bilinen En İyi Değer (Zbest)	Static Beasley	Optimizasyon Seviyesi	En İyi Değer	Ortalama	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama	Std Sapma	En İyi Değer Oranı	En İyi Değer	Ortalama	Std Sapma	En İyi Değer Oranı
1	10	700	700	L1	700	700	0	100%	700	700	0	100%	700	700	0	100%	700	700	0	100%
				L2	700	700	0	100%	700	700	0	100%	700	700	0	100%	700	700	0.00	100%
2	15	1480	1500	L1	1510	1510	0	100%	1510	1510	0	100%	1510	1581.8	94.81	59%	2030	2030	0	100%
				L2	1480	1480	0	100%	1480	1484.8	8.58	76%	1480	1486.8	9.52	66%	1480	2001.6	117.86	3%
3	20	820	1380	L1	820	820	0	100%	820	820	0	100%	820	885.8	177.05	84%	2290	2290	0	100%
				L2	820	820	0	100%	820	820.6	6	99%	820	903.9	189.46	79%	1710	2259.5	110.72	2%
4	20	2520	2520	L1	2940	2940	0	100%	2940	2966.4	45.65	66%	2940	3145.5	127.35	13%	4480	4480	0	100%
				L2	2520	2520	0	100%	2520	2520	0	100%	2520	2520	0	100%	4480	4480	0.00	100%
5	20	3100	5420	L1	3520	3520	0	100%	3520	3638.5	114.63	47%	3520	4058.3	302.79	1%	7120	7120	0	100%
				L2	3100	3100	0	100%	3100	3100	0	100%	3100	3492	296.50	25%	7120	7120	0.00	100%
6	30	24442	24442	L1	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%
				L2	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%	24442	24442	0	100%
7	44	1550	1550	L1	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%	3974	3974	0	100%
				L2	3974	3974	0	100%	1550	1550	0	100%	1550	1550	0	100%	3974	3974	0	100%
8	50	1950	2690	L1	2600	2600	0	100%	2600	2781.35	169.94	14%	2910	3583.25	215.61	2%	4390	4390	0	100%
				L2	2170	2170	0	100%	2170	2313.8	132.44	17%	2455	2870.9	200.58	1%	4390	4390	0.00	100%

5. TARTIŞMA VE SONUÇ

Hava trafiği için yapılan harcamaların fazla olması, Uçak Çizelgelme Problemlerinde (UÇP) optimum çözüme ulaşmak için yeni teknik ve çözümleri ihtiyaç haline getirmiştir. Yeni havaalanları ve yeni pistlerin inşaatı kısa sürede uygulanabilecek bir çözüm olmadığı için, havaalanlarında pist çıkış kapasitesini geliştirmek, uçak iniş ve kalkış planlamalarını bu doğrultuda yeniden yapmak önemlidir. Bu sayede pistler daha verimli şekilde kullanılabilir ya da mevcut ya da beklenen verimsizliklerin önüne geçilebilir ve hatta maliyet kazançları sağlanabilir.

Uçak çizelgeleme problemlerinde, karar verilecek problem (Gelen ya da kalkan uçaklar, pist kullanımı gibi), problemin büyüklüğü gibi konular sezgisel algoritma seçimini etkileyen faktörlerden bazılarıdır. Algoritmalara parametre ayarı yapılması, problem tipi ve büyüklüğü çalışmaların sonuçlarını etkiler. Bununla birlikte, iyi çözümler elde ederken sezgisel yöntemlerle birlikte Tabu Arama Algoritması'nı uygulamanın kolay olduğu belirgindir.

Bu çalışmada, tek pistin kullanıldığı uçak çizelgleme problemi incelenmiştir. Uçak çizelgeleme problemlerini (UÇP) çözmek için, dört metasezgisel algoritma (Tabu Arama, Tavlama Benzetimi, Genetik Algoritma, Karınca Kolonisi Optimizasyonu) özgün seviye 1, seviye 2 optimizasyonlarıyle birlikte kullanılmıştır.

Ele alınan uçak çizelgeleme problemi (UÇP) için L1 ve L2 optimizasyonlu sezgisel yöntemlerle kullanılan algoritmalar C programında kodlanmıştır. Algoritmaların çözüm sonuçları, karışık tamsayılı optmum sonuçlar ve Beasley'in (Beasley ve diğerleri, 2001) bulduğu sonuçlar ile kıyaslanmıştır. Algoritma performansları, uçak sıralamalarının ceza maliyeti değerleri yönünden değerlendirilmiştir. Algoritma dinamik durumda çalıştırıldığında dahi, Beasley'in (Beasley ve diğerleri, 2001) statik durum için elde ettiği sonuçlardan genel olarak daha iyi sonuçlar elde edilmiştir.

Optimize edilen algoritmaların kendi içerisinde kıyaslandığında; algoritmaların statik durumda çalıştırıldığında, Tavlama Benzetimi Algoritması'nın daha iyi sonuç verdiği görülürken, algoritmalar dinamik durumda çalıştırıldığında, Tabu Arama Algoritması'nın daha iyi sonuç verdiği görülmüştür.

Problem çözümleri, çalışmada yapıldığı gibi uçak çizelgeleme problemlerindeki statik ve dinamik durumun ikisi için de kullanılabilir. Tablolarda yer alan algoritma sonuçlarına göre optimize edilmiş önerilen algoritmalardan özellikle üçünün (Tabu Arama, Tavlama Benzetimi, Genetik Algoritma) hava trafik kontrolörleri tarafından pist kapasitesi kullanım oranları ve zaman ve maliyet kazancı sağlama açısından, bir karar destek aracı olarak kullanılabileceği gösterilmiştir.

Çalışmaya göre sadece uçağın inişi planlanamaz, aynı zamanda aynı pist kullanılarak uçağın kalkışı da planlanabilir.

Tüm sonuçlara genel olarak bakıldığında optimize edilmiş algoritmalardan özellikle üçünün (Tabu Arama, Genetik Algoritma, Tavlama Benzetimi) , Beasley'in OR kütüphanesindeki veri setleri için optimum ya da optimuma yakın sonuçlar elde ettiği görülmüştür.

Çalışmaya dahil edilen algoritmalar 500 uçağa kadar olan veri setlerine uyarlanmış olsa da daha büyük test örnekleri için kullanılabilir veya havaalanlarındaki gerçek veri setleriyle çalışılabilir.

Bu çalışmada kullanılan yöntemler hava trafik kontrolörlerinin karar verme sırasında, hava alanı hava sahasında bulunan ve yere inmeyi bekleyen uçakların çizelgelenmesi için zaman ve maliyet kavramları açısından problem çözücü olarak düşünülebilir. Çalışmada aynı zamanda gelen uçakların piste inişe değil aynı zamanda kalkışa geçme zamanları da dikkate alınmıştır. Ayrıca çalışmada geliştirilen model ve sezgisel yöntemle birlikte kullanılan algoritmalar sadece tek pist için değil birden çok pist olması durumunda da kullanılabilecek şekilde oluşturulmuştur.

Tablolaradaki sonuçlara bakıldığında, sonuçların başarılı olduğu görülmektedir. Geliştirilen algoritmanın pist kapasite kullanım oranlarını arttırmak amacıyla bir karar destek aracı olarak hava trafik kontrolörleri tarafından kullanılabileceği ortaya çıkmıştır.

Gelecekteki çalışmalar için, problem hem iniş hem de kalkış operasyonları ile birden fazla piste yayılabilir. Uygun havaalanlarında gerçek veri setleriyle analizler yapılarak çalışılabilir. Uçuş sırasında bağlantılı uçuşları olan transit yolcular planlanırken dikkate alınabilir. Özellikle gece uçuşları için daha büyük boyutlu uçaklar planlanırken uçak gürültü tahminleri dikkate alınabilir ve bu uçakların geliş saatleri öngörülen uçak gürültü tahminine göre çizelgelenebilir. Araştırılıp geliştirilmesi yapılan yeni metasezgisel yöntemler Uçak çizelgelme problemlerine (UÇP), zaman ve maliyet kazancı sağlayabilir. Uçakların piste inişini etkileyen öngörülemeyen durumlar (sıklıkla karşılaşılmayan ama karşılaşılması halinde ciddi sonuçları olan durumlar) da dinamik uçak çizelgeleme konusuna dahil edilebilir.

KAYNAKLAR

Akay, B., 2009, Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi, Fen Bilimleri Enstitüsü, Erciyes Üniversitesi, Kayseri.

Alharkan I., Saleh M., Ghaleb M., Kaid H., Farhan A., Almarfadi A., 2019, Tabu search and particle swarm optimization algorithms for two ıdentical parallel machines scheduling problem with a single server, Journal of King Saud University – Engineering Sciences, In press

Ahmed M., Mansour Z., Haouari M., 2017, A two-level optimization approach for robust aircraft routing and Retiming, Computers & Industrial Engineering, Vol.112, 586-594.

Antoniou, A. ve Lu, W. S., 2007, Springer US, Verlag Newyork, Practical optimization: Algorithms and engineering applications, ISBN: 978-0-387-71107-2 s.1-3.

Awasthi A. , Kramer O., L¨assig J. , 2013, Aircraft Landing Problem: Efficient Algorithm for a Given Landing Sequence, 2013 IEEE 16th International Conference on Computational Science and Engineering, Sydney, NSW, 20-27.

Aydın A., 2009, Metasezgisel Yöntemlerle Uçak Çizelgeleme Problemi optimizasyonu, Doktora Tezi, Sosyal Bilimler Enstitüsü, Marmara Üniversitesi, İstanbul.

Balakrishnan H., Chandran B., 2006, Scheduling aircraft landings under constrained position shifting, AIAA Guidance, Navigation, and Control Conference and Exhibit, Colarado, America 2006-6320.

Balakrishnan, H. and Chandran, B., 2010, Algorithms for scheduling runway operations under constrained position shifting, Operations Research, 58 (6), 1650-1665.

Barnhart, C., Boland, N.L., Clarke L.W., Johnson E.L. and Nemhauser, G.L., 1998, Flight String Models for Aircraft Fleeting and Routing, Transportation Science. (3), 208-220.

Bäuerle, N., Engelhardt-Funke, O., Kolonko, M., 2007, On the waiting time of arriving aircrafts and the capacity of airports with one or two runways, European Journal of Operational Research, 177 2 (1), 1180-1196.

Beasley J., Krishnamoorthy M., Sharaiha Y., Abramson D., 2004, Displacement problem and dynamically scheduling aircraft landings, Journal of the Operational Research Society, Vol.55, 54–64

Beasley, Sonander, Havelock, 2001, Scheduling aircraft landings at London Heathrow using a population heuristic, Journal of the Operational Research Society,Vol.52, 483-493.

Beasley JE., Krishnamoorthy M., Sharaiha Y.M., Abramson D., 2001, Scheduling Aircraft Landings—The Static Case", Transportation Science, Vol.34, 180-197.

Beasley, J.E., Sonander, J., Havelock, P., 2001, Scheduling Aircraft Landings at London Heathrow Using a Population Heuristic, Journal of Operational Research Society, Vol.52, 483-493.

Beasley J.E., Krishnamoorthy M., Sharaiha Y.M., Abramson D., (2000) Scheduling Aircraft Landings—The Static Case, 34, 2.

Be´langer N., Desaulniers G., Soumis F., Desrosiers J., Lavigne J., 2006, Weekly airline fleet assignment with homogeneity, Transportation Research, Part B Vol. 40, 306-318.

Bennell, J. A., Mesgarpour, M., Potts, C. N., 2011, Airport runway scheduling, OR:Quartterly Journal of Operations Research, 9(2), 115-138.

Bennell J., Mesgarpour M., Potts C., 2017, Dynamic scheduling of aircraft landings, European Journal of Operational Research 258, 315–327.

Bencheikh G. , Boukachour J., Alaoui A., 2011, Improved Ant Colony Algorithm to Solve the Aircraft Landing Problem, International Journal of Computer Theory and Engineering, Vol.3, No.2, 1793-8201.

Bencheikh, Boukachour, Alaoui, 2016, A memetic algorithm to solve the dynamic multiple runway aircraft landing problem, Journal of King Saud University – Computer and Information Sciences, Vol.28, 98–109.

Bianco, L., Rinaldi, G. & Sassano, A., 1987, A combinatorial optimization approach to aircraft sequencing problem. Flow Control of Congested Networks. NATO AS I series, 38, 323-339.

Bianco, L., Dell'Olmo, P. & Giordani, S., 1997, Scheduling models and algorithms for TMA traffic management, Modeling and Simulation in Air Traffic Management. Springer, 139-167.

Blum, C., Roli, A., 2003, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, ACM Computer Survey, 35 (3), 268–308.

Bolat A., 2000, Procedures for providing robust gate assignments for arriving Aircrafts. European Journal of Operational Research, Vol.120, 63-80.

Boysen, N., M. Fliedner., 2011, Scheduling aircraft landings to balance workload of ground staff. Computers & Industrial Engineering. 60(2) 206-217.

Bruinsma F., Rietveld P., Brons M., 2000, Comparative Study of Hub Airports in Europe: Ticket Prices, Travel Time and Rescheduluing Costs. Tijdschrift voor Economische en Sociale Geografie, Vol.91, No.3, 278- 292.

Büyüksaatçi S., 2015, Doğadan Esinlenen Ve Ses Yankisina Dayali Çözüm Arama Algoritmasi Ile Belge Kümeleme, Doktora Tezi, Fen Bilimleri Enstitüsü, İstanbul Üniversitesi, İstanbul.

Carr, G.C., Erzberger, H. & Neuman, F., 2000, Fast-time study of airline-influenced arrival sequencing and scheduling. Journal of Guidance,Control, and Dynamics, Vol.23, 526–531.

Clarke, L.W., Johnson, E.L., Nemhauser, G.L. and Zhu, Z., 1997, The Aircraft Rotation Problem, Annals ocomperative study of Operations Research, Vol.69, 33-46.

Çiftçioğlu A. Ö., Doğan E., 2017, Güncel Optimizasyon Tekniklerinin Matematiksel Problemlerin Çözümündeki Performanslarının Kıyaslanması, Celal Bayar Üniversitesi Fen Bilimleri Dergisi, Vol.13, Issue 2, 579 – 591.

Daskin, M.S., Panayotopoulos, N.C., 1989, A lagrangian Relaxation Approach to Assigning Aircraft to routes in Hub and Spoke Networks, Transportation Science. Vol.2, 91-99.

Dear, R. G., Sherif, Y. S., 1989, The dynamic scheduling of aircraft in high density terminal areas. Microelectrons Reliability, Vol.29, 743–749.

Dear, R. G., Sherif, Y. S., 1991, An algorithm for computer assisted sequencing and scheduling of terminal area operations, Transportation Research A, Vol.25, 29-139.

Desaulniers, G., Desrosiers, J. and Dumas, Y., 1997, Daily Aircraft Routing and Scheduling, Management Science. (43), 841-855.

Diallo C., Ndiaye B., Seck D., 2012, Scheduling Aircraft Landings at LSS Airport, American Journal of Operations Research,Vol.2, 235-241.

Drezner T., Drezner Z., 2001, A Note on Applying The Gravity Rule To The Airline Hub Problem. Journal of Reagional Science, Vol. 41, No.1, 67-73.

Dumitrescu, I., Stützle, T., 2003, Combinations of local search and exact algorithms, Applications of Evolutionary Computing, LNCS, Vol.2611, 57-68.

Ernst A., Krishnamoorthy M., Storer R., 1999, Heuristic and Exact Algorithms for Scheduling Aircraft Landings, Operations Research & Management Science, Vol.34, 229–24.

Eröz E., 2020, Çok Amaçli Optimizasyon Algoritmasi: Mogoldsa, Yüksek Lisans Tezi Fen Bilimleri Enstitüsü, Fırat Üniversitesi, Elazığ.

Faye A., 2015, Solving the Aircraft Landing Problem with time discretization approach, European Journal of Operational Research Vol.242, 1028–1038.

Feo, T.A., Bard J. F., 1989, Flight Scheduling and Maintenance Base Planning, Management Science, Vol.35, 1415-1432.

Fink, Voss, 1999, Generic Metaheuristic Applications To Industrial Engineering Problems, Computers and Industrial Engineering Vol.37, 281-284.

García A., Quintero E., 2011, Strategy for Attending Takeoffs and Landings to Reduce the Aircraft Operating Costs and the Passenger Delays, EJTIR, 11(2), 219-233.

Gen, M. & Cheng, R., 1975, Genetic Algorithms and Engineering Design, Wiley Series in Engineering Design and Automation.

Gen, M., 1996, Genetic Algorithms and Industrial Engineering Optimization, Computers and Industrial Engineering, Vol.30, N.4, s.835-837.

Gertsev V., Gerteseva V., 2004, Classification of mathematical models in ecology, Ecological Modelling, Vol.178, Issues 3–4.

Ghoniem A., Sherali H., Baik H., 2014, Enhanced Models for a Mixed Arrival-Departure Aircraft Sequencing Problem, INFORMS Journal on Computing 26(3):514-530.

Ghoniem A., Farhadi F., Reihaneh M., 2015, An accelerated branch-and-price algorithm for multiple runway air craft sequencing problems" European Journal of Operational Research Vol.246, 34–43.

Girish B., 2016, An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem, Applied Soft Computing Vol.44, 200–221.

Gopalan, R. and Talluri, K., 1998, The Aircraft Maintenance Routing Problem, Operations Research. (46), 260–271.

Gürkan H., Gürel S., Aktürk M., 2016, An integrated approach for airline scheduling, aircraft fleeting and routing with cruise speed control, Transportation Research Part C Vol.68, 38–57.

Hane, C.A., Barnhart, C., Johnson E.L., Marsten R.E., Nemhauser G.L. and Sigismindi, G., 1995, The Fleet Assignment Problem: Solving A Large Scale Integer Program, Mathematical Programming, Vol.70, 211-232.

Hancerliogullari G., Rabadi G., Al-Salem A., Kharbeche M., 2013, Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem, Journal of Air Transport Management Vol.32, 39-48.

Hançerlioğulları G., 2013, Approximate Algorithms For The Combined Arrival-Departure Aircraft Sequencing And Reactive Scheduling Problems On Multiple Runways, Doktora Tezi, Mühendislik Yönetimi, Old Dominion Üniversitesi, Ankara.

Hazneder B., 2017, Benzetilmiş tavlama algoritması ile adaptif ağ tabanlı bulanık mantık çıkarım sisteminin (ANFIS) eğitilmesi, Doktora Tezi, Fen Bilimleri Enstitüsü, Erciyes Üniversitesi, Kayseri.

Heidt A., Helmke H., Kapolke M., Liers F., Martin A., 2016, Robust runway scheduling under uncertain conditions, Journal of Air Transport Management Vol.56, 28-37.

Hsu, C.I.., Wen, Y.H., 2000, Application of Grey theory and multiobjective programming towards airline network design. European Journal of Operational Research, Vol.127, 44-68.

Hsu, c.i. , Wen, Y.h., 2005, Airline Flight Frequency Determination in Response to Competitive Interactions Using Fuzzy Logic. Mathematical and Computer Modelling, Vol.42, 1207-1224.

Jamili A., A robust mathematical model and heuristic algorithms for integrated aircraft routing and scheduling, with consideration of fleet assignment problem, 2017, Journal of Air Transport Management Vol.58, 21-30.

Jarrah, A. I., Strehler, J.C., 2000. An optimization model for assigning through Fights, IIE Transactions. Vol.32, pp.237-244.

Kahraman Ü. A., Klasik ve Zeki Optimizasyon Teknikleri İle Problemlerin Çözümlenmesi ve Karşılaştırılması, 2010, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, Sakarya.

Kaleli S., 2019, Toplu Taşima Sistemlerinin Evrimsel Algoritmalarla Optimizasyonu: Erzurum Büyükşehir Belediyesi Örneği, Doktora Tezi, Sosyal Bilimler Enstitüsü, Atatürk Üniversitesi, Erzurum

Karaboğa, D., 2014. Yapay Zeka Optimizasyon Algoritmaları. Nobel Akademik Yayıncılık Eğitim Danışmanlık Tic. Ltd. Şti., Ankara, ISBN:978-605-133-764-7 231s.

Kıran M., 2014, Optimizasyon problemlerinin çözümü için yapay ari kolonisi algoritmasi tabanli yeni yaklaşimlar, Doktora Tezi, Fen bilimleri enstitüsü, Selçuk Üniversitesi, Konya.

Koziel, S., Yang, X-S. (Eds.), 2011, Computational Optimization, Methods and Algorithms, Springer, Verlag Berlin Heidelberg, ISBN: 978-3-642-20858-4.

Köksalan, M. & Erkip, N., 2000. Yöneylem Araştırması - Halim Doğrusöz'e Armağan. 89-128.

Laporte G., Gendreau M., Potvin J., Semet F., 2000, Classical And Modern Heuristics For Vrp, International Transactions In Operational Research Vol.7, 285-300.

Lieder A., Briskorn D., Stolletz R., 2015, A dynamic programming approach for the aircraft landing problem with air craft classes, European Journal of Operational Research Vol.243, 61–69.

Lieder A., Stolletz R., 2016, Scheduling aircraft take-offs and landings on interdependent and heterogeneous runways, Transportation Reaserach Part E: Logistics and Transportation Review, Vol.88, 167-188.

Lee, Y. H., Pinedo, M., 1997, Theory and methodology: Scheduling jobs on parallel machines with sequence-dependent setup times, European Journal of Operational Research, Vol.100, 464-474.

Lee H., Balakrishan H., 2008, A study tradeoffs in scheduling terminal-area operations., Proceedings of the IEEE, Vol.96 (12), 2081-2095.

Lohatepanont, M. and Barnhart, C., 2004, Airline Schedule Planning: Integrated Models and Algorithms for Schedule Design and Fleet Asssignment, Transportation Science, Vol.38, 19-32.

Luo, Q., Duan H., 2014, Chaotic Artificial Bee Colony Optimization Approach to Aircraft Automatic Landing System, Proceedings of the 19th World Congress The International Federation of Automatic Control, 47 (3), 876-881

Mahmud A., 2017, Dynamic Aircraft Scheduling Using Optimized FCFS Algorithm, Jour of Adv Research in Dynamical & Control Systems, 13-Special Issue, 583-593.

Mathaisel, D.F.X, 1997, Decision Support for Airline Schedule Planning. Journal of Combinatorial Optimization , (3), 251–275.

Mirjalili S., 2016, SCA: A sine cosine algorithm for solving optimization problems, Knowledge-BasedSystems, Vol.96, 120–133.

Mukherjee A, Hansen M., 2009, A dynamic rerouting model for air traffic flow management, Transportation Research Part B Vol.43, 59–171.

Orhan, İ, Kapanoğlu, M. and Karakoç, T.H., 2007, Flight-HourBased Optimization for Aircraft Scheduling, INFORMS, Conference, Seattle, 369,WD43.

Önal Z., 2020, Çok Amaçli Esnek Atölye Tipi Çizelgelemenin MetaSezgisel Optimizasyon Yöntemleri Ile Çözümü Ve Kiyaslanmasi, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Yıldız Teknik Üniversitesi, İstanbul.

Panos Y. P., Douglass J. W., Principles of Optimal Design : Modeling and Computation, Cambridge University Press. 2000.

Park, B.J., Choi, H.R. & Kim, H.S., 2003. A hybrid genetic algorithm for the job shop scheduling problems, Vol.45, 597–613.

Pinol H., Beasley J.E., 2006, Scatter Search and Bionomic Algorithms for the aircraft landing problem, European Journal of Operational Research Vol.171, 439–462.

Pulugurtha S. S., Nambisan S. S., 2003, A Decision–Support Tool for Airline Yield Management Using Genetic Algorithms. Computer-Aided Civil and Infrastructure Engineering, Vol.18, No.3, 214-223.

Rexing, B., Barnhart, C., Kniker, T., Jarrah, A. and Krishnamurthy, N., 2000, Airline fleet Assignment with time Windows, Transportation Science. (34), 1-20.

Sabar N., Kendall G., 2015, An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem, Vol.56, Pages 88–98.

Saka E., 2016, Öğretmen Adaylarinin Matematiksel Modelleme Problemlerini Çözme Sürecinde Teknolojinin Rolü, Doktora Tezi, Karadeniz Teknik Üniversitesi, Trabzon.

Salehipour A., Modarres M., Naeni L., 2013, An efficient hybrid meta-heuristic for aircraft landing problem, Computers & Operations Research Vol.40, 207–213.

Samà M., D'Ariano A., Corman F., Pacciarelli D., 2017, Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas, Transportation Research Part C: Emerging Technologies, Vol.80, 485-511.

Samà M., D'Ariano A., D'Ariano P., Pacciarelli D., 2014, Optimal aircraft scheduling and routing at a terminal control area during disturbances, Transportation Research Part C Vol.47, 61–85

Samà M., D'Ariano A., D'Ariano P., Pacciarelli D., 2017, Scheduling models for optimal aircraft traffic control at busy airports: Tardiness, priorities, equity and violations considerations, Omega, Vol.67 81–98.

Samà M., D'Ariano A., Corman F., Pacciarelli D., 2017, Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas, Transportation Research Part C: Emerging Technologies, Vol80, 485-517.

Shenpeng, Cao, Zhang, 2011, A real-time schedule method for Aircraft Landing Scheduling problem based on Cellular Automation, Vol.11, Issue 4,3485–3493.

Sherali H.D., Bish E., Zhu X., 2006, Airline fleet assignment concepts, models, and algorithms, European Journal of Operational Research, Vol.172, 1-30.

Sarac, A., Batta, R. and Rump, C.M., 2006, A Branch-and-Price Aprroach For Operational Aircraft Maintenance Routing, European Journal of Operational Research. (175), 1850-1869.

Sriram, C. and Haghani, A., 2003, An Optimization Model For Aircraft Maintenance Scheduling And Re-Assignment, Transportation Science. (37), 29-48.

Stojkovic, G., Soumis, F., Desrosiers, J., Solomon M.M., 2002, An optimization model for a real-time flight scheduling problem. Transportation Research, Part A Vol.36, 779–788.

Soomer, M.J. & Franx, G.J., 2008, Scheduling aircraft landings using airlines' preferences. European Journal of Operational Research, Vol.190, 277-291.

Soykan B., 2016, A Hybrid Tabu/Scatter Search Algorithm For Simulation-Based Optimization Of Multi-Objective Runway Operations Scheduling, Doktora Tezi, Mühendislik Yönetimi, Old Dominion Üniversitesi, Virginia

Talbi, E., (2009). Metaheuristics From Desing To Implementation, First Edition, A John & Sons, New Jersey, ISBN: 978-0-470-27858-1, 4-18.

Vadlamani S., Hosseini S., 2014, A novel heuristic approach for solving aircraft landing problem with single runway, Journal of Air Transport Management Vol.40, 144-148.

Venkatakrishnan, C.S., Barnett, A.I. & Odoni, A.R., 1993, Landings at Logan Airport: Describing and increasing airport capacity, Transportation Science, Vol.27, 211-227.

Vos H., Santos B., Omondi, 2015, Aircraft Schedule Recovery Problem – A Dynamic Modeling Framework for Daily Operations, Transportation Research Procedia Vol.10, 931 – 940.

Wang X., Regan A., 2006, Dynamic yield management when aircraft assignments are subject to swap. Transportation Research, Part B Vol.40, 563-576.

Wensveen, J. G., 2015, Air transportation: A management perspective: Ashgate Publishing, ISBN: 978-1-4724-3678-8.

Winston, W. L., 2004, Operations Research: Applications and Algorithms, Thomson Brooks / Cole California, ISBN: 978-0-534-38058-8 s.2.

WU, C.L., 2005, Inherent delays and operational reliability of airline schedules. Journal of Air Transport Management, Vol.11, 273-282.

Wu Y., Sun L., Qu X., 2016, A sequencing model for a team of aircraft landing on the carrier Aerospace Science and Technology Vol.54, 72–87.

Yan s., Tang c.-h., Lee m.-c., 2007, A flight scheduling model for Taiwan airlines under market competitions, The International Journal of Management Science, Vol.35, 61-74.

Xu B., 2017, An efficient Ant Colony algorithm based on wake-vortex modeling method for aircraft scheduling problem, Journal of Computational and Applied Mathematics Vol.317, 157–170.

Yang, X-S., 2010, *Engineering Optimization An Introduction with Metaheuristic Applications*, John Wiley & Sons Inc., Hoboken, New Jersey, ISBN: 978-0-470- 58246- 6.

Yang, X-S., 2010, Nature-Inspired Metaheuristic Algorithms, Luniver Press, ISBN: 1-905986-28-9.

Yu, S. P., Cao, X.B., Zhang, J. (2011). A real-time schedule method for aircraft landing scheduling problem based on cellular automation, Applied Soft Computing Journal.

EKLER

ÖZGEÇMİŞ



Kişisel Bilgiler	
Adı Soyadı	Gülnar Çalışkan
Doğum Yeri	Bulgaristan
Doğum Tarihi	03.07.1982
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	+905072325002, +352661308101
E-Posta Adresi	gulnar@caliskan.net
Web Adresi	

Eğitim Bilgileri	
Lisans	
Üniversite	Dumlupınar Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Endüstri Mühendisliği
Mezuniyet Yılı	2006

Yüksek Lisans	
Üniversite	Dokuz Eylül Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Lojistik Mühendisliği Anabilim Dalı
Programı	Lojistik Mühendisliği Programı