

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**UÇAK İNİŞLERİNİN GERÇEK ZAMANLI ÇİZELGELENMESİNİN
TASARIMI VE SİMÜLASYONU**

Beraat ALDEMİR

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2018**

Her hakkı saklıdır

TEZ ONAYI

Beraat Aldemir tarafından hazırlanan “UÇAK İNİŞLERİNİN GERÇEK ZAMANLI ÇİZELGELENMESİNİN TASARIMI VE SİMÜLASYONU” adlı tez çalışması 06/06/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Danışman : Prof. Dr. Şahin Emrah



Jüri Üyeleri:

Başkan: Doç. Dr. Süleyman Tosun
Hacettepe Üniversitesi/Bilgisayar Mühendisliği



Üye : Dr. Öğr. Üyesi Bülent Tuğrul
Ankara Üniversitesi/Bilgisayar Mühendisliği



Üye : Prof. Dr. Şahin Emrah
Ankara Üniversitesi/Bilgisayar Mühendisliği




Yukarıdaki sonucu onaylarım.

Prof. Dr. Atila YETİŞEMİYEN
Enstitü Müdürü

ETİK

Ankara Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

19.06.2018



Beraat ALDEMİR

ÖZET

Yüksek Lisans Tezi

UÇAK İNİŞLERİNİN GERÇEK ZAMANLI ÇİZELGELENMESİNİN TASARIMI VE SİMÜLASYONU

Beraat ALDEMİR

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Şahin EMRAH

Globalleşen dünyada ulaşım taşıtları büyük önem taşımaktadır. En hızlı ulaşımı sağlayan hava taşımacılığı ise günden güne daha önemli hale gelmektedir. Uzun mesafeli seyahatlerde konfor ve hız açısından hava taşıtları tercih edilmektedir. Hava taşıtlarının ulaşım düzenini ve güvenilirliğini sağlaması için hava trafik kontrolü sistemi (HTKS) etkin bir rol oynamaktadır. Bu tezin amacı, hava trafik kontrol sistemlerinin önemli alt işlerinden biri olan uçuş zamanlama ve sıralama problemlerini ve bunların çözümü için sunulan algoritmaları ele alıp bu çözümlerin geliştirilmesinin mümkün olup olmadığını değerlendirmek ve simülasyonlarını gerçekleştirmektir. Tez içerisinde değerlendirilecek algoritmalar genetik algoritma ve açgözlü algoritma olacaktır. Bu algoritmaların çalışma zamanları ve verdikleri sonuçların optimuma yakınlığı karşılaştırılacaktır. Algoritmaların sonuçları hali hazırda kullanılan ilk gelene ilk hizmet tekniğinin sonuçlarıyla karşılaştırılacaktır. Aynı zamanda mevcutta bulunan algoritmaların geliştirilmelerinin mümkün olup olmadığı tartışılacaktır. Hava yolu taşımacılığının maliyetleri çok yüksek olduğundan uçuş sıralaması sırasındaki gecikmeler firmalara büyük miktarlarda para kaybı yaşatmaktadır. Amerikan Havacılık İdaresi'nin 2017 yılında açıkladığı verilere göre hava yolu şirketleri için bir yıllık toplam gecikme maliyeti 6.3 milyar dolar olmuştur. Bu bilgiler ışığında düşünüldüğünde bu tip gecikmeyi azaltma algoritmalarının maliyet açısından şirketlere büyük katkılar sağlayacağı görülmektedir.

Haziran 2018, 40 sayfa

Anahtar Kelimeler: Çizelgeleme, Eniyileme, Hava trafik kontrolü, Uçuş inişleri sıralama, Açgözlü algoritma

ABSTRACT

Master Thesis

DESIGN AND SIMULATION OF AIRCRAFT LANDING

Beraat ALDEMİR

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Şahin EMRAH

Transportation vehicles have a big importance in a globalizing world. Air transportation which is the fastest transportation option is getting more and more important. Air transportation is preferred in long distance journeys because of comfort and speed. Air traffic control system has an effective and critical role in securing and ordering air transportation vehicles. Air traffic control systems have all the responsibility about aircrafts' landings, take offs and flights in their responsibility zones. The purpose of this thesis is discussing flight scheduling and ordering problems which is one of the sub works of air traffic control systems and algorithms to solve those problems. Genetic algorithm and greedy algorithm techniques will be compared for the solution of problem. Working time and results of algorithms will be compared. Results of algorithms will be compared to existing first come first served technique. Another purpose of this thesis is discussing if it is possible to improve those algorithms and providing simulations of existing and suggested algorithms. Latencies during scheduling of air traffic causes a huge amount of money to companies as the cost of air transportation is very high. According to data by Federal Aviation Administration in 2017, total cost of latencies was estimated as 6.3 billion US dollars. Accordingly, algorithms to decrease these latencies can be so useful for airline companies to save a lot of money.

June 2018, 40 pages

Key Words: Scheduling, optimization, Air Traffic Control, Aircraft Landing Sorting, Greedy Algorithm

TEŐEKKÜR

Lisans ve yüksek lisans hayatım boyunca elinden gelen tüm yardımları yapan danışman hocam sayın Prof. Dr. Őahin EMRAH'a, (Ankara Üniversitesi Bilgisayar Mühendisliđi Anabilim Dalı) tezimle ilgili yardımlarını esirgemeyen Bilgisayar Mühendisliđi Anabilim Dalı Araő. Gör. Zeynep YILDIRIM ve Araő. Gör. Özge MERCANOĐLU SİNCAN'a, yüksek lisans öğrenimimi tamamlamama olanak tanıyan kurumum TÜBİTAK ULAKBİM'e, belge süreçlerinin hızlı tamamlanması için elinden geleni yapan Öğrenci İşleri çalışanı Seval KANDEMİR'e ve her zaman yanımda olan ailem ve eşime en içten duygularıyla teşekkür ederim.

Beraat ALDEMİR

Ankara, Haziran 2018

İÇİNDEKİLER

TEZ ONAY SAYFASI

ÖZET.....	ii
ABSTRACT	iii
TEŞEKKÜR	iv
SİMGELER DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
ÇİZELGELER DİZİNİ	ix
1. GİRİŞ	1
1.1 Problemin Tanımı	1
1.1.1 İniş zamanı.....	2
1.1.2 Ayırma.....	2
1.1.3 Hava durumu.....	3
1.2 Problemin Matematiksel Modellenmesi.....	3
1.3 Önceki Çalışmalar.....	6
2. HAVA TRAFİK KONTROLÜ	8
2.1 Dinamik Planlayıcı.....	9
2.2 Dinamik Planlayıcı Girişleri	9
2.2.1 Uçuş planları.....	9
2.2.2 Takip güncellemeleri.....	10
2.2.3 Tahmini varış zamanları (TVZ'ler)	10
2.3 Dinamik Planlayıcıdan Çıkışlar.....	11
2.4 İlk Gelene İlk Hizmet Prensipleri(İGİH).....	11
3. PROBLEM ÜZERİNDE DAHA ÖNCE YAPILMIŞ ÇALIŞMALAR.....	12
3.1 Genetik Algoritma ile Problemin Optimizasyonu.....	12
3.1.1 Genetik Algoritma.....	12
3.1.2 Örnek çalışma.....	14
3.2 Açgözlü Algoritma ile Optimizasyon.....	17
3.2.1 Açgözlü Algoritma	17
3.2.2 Örnek çalışma.....	18
4. ÖNERİLEN YENİ ALGORİTMA.....	21
4.1 Genetik Algoritmanın Özelleştirilmesi.....	23

4.2 Önerilen Algoritmanın Karşılaştırılması.....	24
4.3 Algoritma Karşılaştırmalarının Sonuçları	29
5. SİMÜLASYON.....	30
5.1 Genetik Algoritma Simülasyonu.....	30
5.2 Ağgözlü Algoritma Simülasyonu	31
5.3 Geliştirilmiş Ağgözlü Algoritma Simülasyonu	33
6. SONUÇ.....	35
KAYNAKLAR	36
EK 1 Geliştirilmiş Ağgözlü Algoritma Örnek Verileri ve Sonuçları.....	38
ÖZGEÇMİŞ.....	40



SİMGELER DİZİNİ

Σ	Kartezyan Toplam
μ	Mutasyon oranı

Kısaltmalar

SDP	Varişların sıralama ve zamanlama
TRACON	Terminal radar yaklaşma kontrolü
CTAS	Merkez-TRACON otomasyon sistemi
DA	Açgözlü Algoritma
DP	Dinamik planlayıcı
TVZ	Tahmini varış zamanı
İĞİH	İlk gelene ilk hizmet verilir yöntemi
GA	Genetik Algoritma
HTK	Hava trafik kontrolü
rand	Rastgele sayı
PVS	Planlanan varış süresi
TMC	Trafik yönetim koordinatörü

ŞEKİLLER DİZİNİ

Şekil 3.1 Agözlü algoritma akış şeması.....	19
Şekil 4.1 Geliştirilmiş Agözlü Algoritma Akış Diyagramı.....	23
Şekil 4.2 Çalışma zamanı kıyaslaması (nano saniye).....	25
Şekil 4.3 Çalışma zamanı kıyaslaması (nano saniye).....	26
Şekil 4.4 Toplam iniş tamamlanma süresi.....	26
Şekil 4.5 Toplam iniş tamamlanma süresi.....	27
Şekil 4.6 Tüm test dosyalarının çalışma zamanı kıyaslaması (nano saniye).....	27
Şekil 4.7 Tüm test dosyalarının toplam iniş tamamlanma süresi.....	28
Şekil 5.1 Simülasyonda kullanılan uçak ikonları.....	31
Şekil 5.2 Genetik algoritma simülasyon ekran görüntüsü.....	32
Şekil 5.3 Agözlü Algoritma Simülasyon Ekran Görüntüsü.....	34
Şekil 5.4 Geliştirilmiş Agözlü Algoritma Simülasyon Ekran Görüntüsü.....	35
Şekil 5.2 Genetik algoritma simülasyon ekran görüntüsü.....	32

ÇİZELGELER DİZİNİ

Çizelge 4.1 Tüm test dosyalarında algoritmaların çalışma süreleri (milisaniye).....	28
Çizelge 4.2 Tüm test dosyalarının toplam iniş tamamlanma süreleri.....	29



1. GİRİŞ

1960'lı yılların sonlarından itibaren, ticari uçaklarında kullanılmaya başlanması ile hava trafiğinde çok büyük bir artış yaşandı. Günümüzde, artık bu trafiği kısıtlayan dünyadaki uçak sayısından ziyade havalimanlarındaki pist sayısı olmaktadır. Yoğun olarak çalışan havalimanlarında pist sayıları olabildiğince arttırılmış, hatta artık pist eklemek imkansız duruma gelmiştir. Bu sebepten dolayı, pistlerin verimliliğini arttırmak daha büyük önem taşımaktadır. Uçakların pistleri beklerken harcadıkları küçük zamanlar bile havayolu şirketlerine büyük zararlar olarak dönmektedir. Çünkü havayolu ulaşımındaki gecikmelerin şirketlere maliyetleri çok yüksektir. 2011 yılında Amerikan Havacılık İdaresi'nin açıkladığı verilere göre Amerika'da havayolu şirketlerinin gecikmeler yüzünden ettiği zarar 8 milyar dolar seviyesindedir. Hali hazırda hava trafik kontrol uzmanları gelen uçakları genellikle ilk gelene ilk hizmet (İĞİH) prensibiyle iniş yaptırmaktadırlar. Oysaki optimizasyon çözümleriyle İĞİH'ten daha verimli iniş sıralamaları elde edilebilir. Bu optimizasyon çözümleriyle saatte sadece 1 fazla uçağın inişi sağlansa bile yılda 8760 adet fazladan iniş yaptırılmış olacaktır. Bu da hem havayolu şirketlerine hem de havalimanı işletmelerine katma değer olarak yansıtacaktır. Bu konuda geliştirilen algoritmalar havalimanlarındaki pistlerin en optimize şekilde kullanılmasını, gecikmelerden doğacak zararların minimuma indirilmesini ve iniş trafiğinin güvenli şekilde sağlanmasını amaçlar.

1.1 Problemin Tanımı

Uçak inişlerinin sıralanması oldukça zor bir problemdir. Çünkü iniş sıralamasını optimize ederken göz önünde bulundurulması gereken çok sayıda parametre vardır.

Bu parametreler:

1.1.1 İniş zamanı

Her uçağın bir piste en erken inebileceği bir zaman vardır. Bu zaman hiçbir gecikmenin, yavaşlamanın olmadığı durumda geçerli olan zamandır. Bir de her uçağın iniş yapabileceği en geç zaman vardır. Bu zamanın olmasının sebebi ise uçağın havada inişi beklerken yakıtının bitme ihtimali olmasındandır. Bu iki zaman uçağın iniş yapabileceği zaman aralığını belirler. Bu zamanların yanı sıra bir de hedef iniş zamanı kavramı mevcuttur. Bu kavram ise bazı ekonomik parametreler göz önüne alarak hesaplanmış, bahsedilen zaman aralığı içerisindeki inişin yapılması hedeflenen zamanı belirtir. Uçak inişleri planlanırken her uçağa bir tampon zaman verilir. Bu sayede, kalkıştaki veya havadaki bazı sebeplerden yaşanan gecikmeler tolere edilerek hedeflenen zamanda iniş gerçekleşebilir. Bu tampon zamandan kaynaklı olarak uçak bazen hedeflenen zamandan daha erken havalimanına ulaşabilir. Statik durum için planlama yapılırken en erken, hedeflenen ve en geç iniş zamanları hava yolu şirketlerinin verdikleri bilgilere ve isteklerine göre şekillenir. Bu şekilde hava yolu şirketlerinin de uçakları verimli şekilde diğer uçuşlar için planlaması sağlanmış olur.

1.1.2 Ayırma

Uçaklar uçuş ve iniş sırasında türbülans yaratırlar. Büyük uçakların yarattığı türbülans küçük uçakların yarattığı türbülansa göre daha fazladır. Bu türbülans iniş sırasında sorun yaratabilir. Bu sebepten ötürü iki uçak inişi sırasında ayrılan bir zaman vardır. Eğer bu zaman gereğinden az ayrılırsa uçaklar zarar görebilir, kazalar oluşabilir. Her uçağın tipine göre yaratacağı türbülans farklı olacağından ayrılması gereken zaman da farklı olacaktır. Aynı şekilde bu zaman iniş yapan uçağın tipiyle birlikte kendisinden sonra inecek uçağın tipine göre de farklılık gösterir. Örneğin büyük bir uçağın arkasından gelen küçük uçak için ayrılması gereken süre, küçük uçağın arkasından gelen büyük bir uçak için ayrılması gereken süreden daha uzundur.

1.1.3 Hava durumu

Hava durumu inişlerin gerçekleştirilmesi için önemli parametrelerden biridir. O an pist dolaylarındaki sis, yağış rüzgar durumları iniş planlamalarına doğrudan etki etmektedir. Hatta bazı havalimanlarında rüzgarın estiği yöne göre uçakların ineceği pist bile değişiklik gösterebilmektedir. Ayrıca bu tip durumların her uçağa olan etkisi de farklı olmaktadır. Örneğin; şiddetli rüzgarın küçük uçaklara etkisi fazla olmasına rağmen büyük uçaklara etkisi daha az olacaktır. Planlama yaparken bu değerler de göz önünde bulundurulmaktadır. Bir uçak bir piste indiğinde diğer pistlerin konumuna göre o pistlerde bile türbülansa yol açabilmektedir. Ama bu gibi durumlar uçak iniş zamanı planlaması probleminde göz önünde bulundurulmaz. İlerleyen bölümlerde anlatılacak algoritmalarda da bu tip durumlar göz önüne alınmamaktadır.

1.2 Problemin Matematiksel Modellenmesi

Bu bölümde birden fazla pistin bulunduğu statik uçak iniş probleminin (Beasley vd. 2000) formülasyonu temel alınarak tam sayı formülasyonu anlatılacaktır.

Uçak kümesi P verilmiş olsun, her i uçağının inişi için bir zaman penceresi vardır $[E_i, L_i]$, ve bir de hedeflenen iniş zamanı $T_i (E_i \leq T_i \leq L_i)$ vardır. Uçak bu T_i zamanında indirilmesi durumunda ek maliyet 0 olacaktır. S_{ij} ise 1.1.2 bölümünde açıklanan i ve j uçağının aynı piste inmesi için aralarında geçmesi gereken ayırma zamanıdır. Bölüm 1.1.3 te bahsedilen uçakların ayrı pistlere inmesi durumunda bile diğer pistteki uçağa etki eden ayırma zamanı 0 olarak kabul edilmiştir. g_i ve h_i değerleri ise sırasıyla i uçağının erken ve geç inmesi durumunda oluşan ekstra birim maliyeti ifade etmektedir.

Karar parametreleri aşağıdaki gibidir:

$$x_i = i \text{ uçağının iniş zamanı } (i \in P)$$

$$\alpha_i = \text{uçak planlanan zamandan } (T_i) \text{ ne kadar erken indi } (i \in P)$$

$$\beta_i = \text{uçak } T_i \text{ 'den ne kadar geç indi } (i \in P)$$

$\delta_{ij} =$ eğer i uçağı j 'den önce indiyse 1, diğer durumlarda 0, ($i, j \in P; i \neq j$)

$z_{ij} =$ eğer i ve j uçağı aynı piste inmişse 1, diğer durumlarda 0, ($i, j \in P; i \neq j$)

$y_{jr} =$ eğer j uçağı r pistine inmişse 1, diğer durumlarda 0, ($j \in P; r \in R$)

Buradaki problem iniş zamanı olan x 'i ve iniş pistini belirleyen y 'yi bulmaktır. Öyle ki bu bulduğumuz değerler aşağıdaki şartları sağlamalıdır:

- her uçak belirlenmiş olan zaman penceresinde iniş yapmalıdır;

$$X_i \in [E_i, L_i] \forall i \in P; \quad (1.1)$$

- Bir uçak indikten sonra kendisinden sonra aynı piste inen tüm uçaklar için ayırma zamanı kriteri sağlanmalıdır. Eğer $\delta_{ij} = 1$ ve $z_{ij} = 1$ ise aşağıdaki koşullar sağlanmalıdır;

$$x_j \geq x_i + S_{ij} \forall i, j \in P; i \neq j \quad (1.2)$$

Bu bilgiler ışığında aşağıdaki gibi bir matematiksel formül yazabiliriz:

$$\text{Min } \sum_{i=1}^P (g_i \alpha_i + h_i \beta_i) \quad (1.3)$$

$$E_i \leq x_i \leq L_i, \forall i \in P \quad (1.4)$$

$$\delta_{ij} + \delta_{ji} = 1, \forall i, j \in P; i \neq j \quad (1.5)$$

$$\sum_{r=1}^R y_{ir} = 1, \forall i \in P; r \in R \quad (1.6)$$

$$z_{ij} = z_{ji}, \forall i, j \in P; i \neq j \quad (1.7)$$

$$z_{ij} > y_{ir} + y_{jr} - 1, \forall i, j \in P; i \neq j \quad (1.8)$$

$$x_j \geq x_i + S_{ij} z_{ij} - (L_i + S_{ij} - E_j) \delta_{ji}, \forall i, j \in P; i \neq j \quad (1.9)$$

$$\alpha_i \geq T_i - x_i, \forall i \in P \quad (1.10)$$

$$0 \leq \alpha_i \leq T_i - E_i, \forall i \in P \quad (1.11)$$

$$\beta_i \geq x_i - T_i, \forall i \in P \quad (1.12)$$

$$0 \leq \beta_i \leq L_i - T_i, \forall i \in P \quad (1.13)$$

$$x_i = T_i - \alpha_i + \beta_i, \forall i \in P \quad (1.14)$$

$$x_i, \alpha_i, \beta_i \geq 0, \forall i \in P \quad (1.15)$$

$$\delta_{ij}, y_{ij}, z_{ij} \text{ binary}, \forall i, j \in P \quad (1.16)$$

1.3'teki formül hedeflenen zamana göre yaşanan sapmaya göre doğacak ekstra maliyetlerin toplamını en aza indirmeyi amaçlar. 1.4'teki kısıtlama uçağın belirlenen kendi zaman penceresinde iniş yapmış olmasını sağlar. 1.5'teki kısıtlama i uçağının j 'den önce inmiş olduğunu(bu durumda δ_{ij} değeri 1 olur) veya j uçağının i uçağından önce inmiş olduğunu(bu durumda δ_{ji} değeri 1 olur) belirtir. 1.6'daki kısıtlama her bir uçağın sadece 1 piste iniş yapabileceğini belirtir. 1.7'deki kısıtlama ise simetriktir. Eğer i ve j uçağı aynı piste inmişse, j ve i uçağı da aynı piste inmiştir. 1.8'deki kısıtlamada eğer i ve j uçaklarının ikisinin de indiği bir r pisti varsa ($y_{ir} = y_{jr} = 1$) z_{ij} 'yi 1 olarak belirleriz. Eğer $z_{ij} = 0$ ise 1.8 deki kısıtlama şu şekli alır; $0 \geq y_{ir} + y_{jr} - 1$. Bu da demektir ki i ve j uçakları aynı piste inmiş olamaz. 1.9'daki kısıtlama i ve j uçakları arasındaki ayırma zamanını belirtir. Bunun için 4 durumu göz önünde bulundurmalıyız.

Eğer $z_{ij} = 0$ ve $\delta_{ji} = 1$ ise i ve j farklı pistlere inmiş ve j uçağı i 'den önce iniş yapmıştır. Böylelikle

$$x_j \geq x_i - (L_i + S_{ij} - E_j)$$

aşağıdaki formülle aynıdır:

$$x_j - E_j \geq x_i - L_i - S_{ij}$$

$x_j - E_j \geq 0$ ve $x_i - L_i - S_{ij} \leq 0$ olduğu için yukarıdaki formül de her zaman doğru olacaktır.

- Eğer $z_{ij} = 0$ ve $\delta_{ji} = 0$ (j 'nin i 'den sonra farklı bir piste indiğini belirtir), 1.9'daki kısıtlama aşağıdaki hale dönüşür.

$$x_j \geq x_i + 0 - 0$$

$\delta_{ji} = 0$ değeriyle uyumlu hale gelir.

- Eğer $z_{ij} = 1$ ve $\delta_{ji} = 1$ (j 'nin i 'den önce aynı piste indiğini belirtir), 1.9'daki kısıtlama aşağıdaki hale dönüşür.

$$x_j \geq x_i + S_{ij} - (L_i + S_{ij} - E_j)$$

bu da aşağıdaki formülle aynıdır:

$$x_j - E_j \geq x_i - L_i$$

- Eğer $z_{ij} = 1$ ve $\delta_{ji} = 0$ (j 'nin i 'den sonra aynı piste indiğini belirtir), 1.9 kısıtlaması aşağıdaki hale dönüşür.

$$x_j \geq x_i + S_{ij}$$

i ve j arasındaki ayırma zamanının yeterli olmasını sağlar.

1.10 ve 1.11 kısıtlamaları α_i 'nin en az 0 ile T_i ve x_i arasındaki zaman farkı arasında olduğunu belirtir. 1.12 ve 1.13 kısıtlamaları ise benzer şekilde β_i değeri için geçerlidir. 1.14 kısıtlaması ise x_i iniş zamanı ile (α_i) hedeflenen zamandan (T_i) ne kadar erken indiğini veya β_i hedeflenen zamandan ne kadar geç inildiği arasındaki bağlantıyı belirtir.

1.3 Önceki Çalışmalar

Havalimanlarındaki trafiğin, günden güne artmasıyla birlikte uçak iniş ve kalkışlarının hızlı ve etkili şekilde gerçekleştirilmesi büyük önem taşımaktadır. Bu konuda yapılan

arařtırmalar ve alıřmalar gn getike artmaktadır. Sorunun özm iin hem basit deneyimsel metotlar hem de optimal metotlar geliřtirilmiřtir.

Beasley vd. (2000) bu problem iin karıřık tam sayı formln ve bu konuda yayınlanmıř alıřmaların detaylı bir incelemesini sunar. Bu alıřmada 6 eřit yeni kısıtlama parametresi nerilir. Bu sayede problem lineer programlama tabanlı aėa araması ile OR-ktphanesindeki 50 uaėa kadar oluřturulmuř veri setleri iin optimal olarak özlr. Etkili bir deneyimsel algoritmadan da bahsedilmiřtir.

Ernst vd. (1999) iniř zamanını ok hızlı řekilde özen kısmi sıralamaya dayalı zelleřtirilmiř bir simpleks algoritmayı sunar. Bu metot tek ve oklu pist problemlerini özmeye alıřan buluřsal ve diėer bazı algoritmalarda da kullanılır. İřlem ncesi adımlardan bazıları zaman penceresini daraltma ve kısmi sıralama yapma gibi adımlardır.

Jung vd. (2003), segmentasyona dayalı bir sezgisel algoritma nermektedir. Zaman ufku zaman dilimlerine ayrılır ve bu řekilde problemin alt problemleri belirlenir. Her bir alt problem Beasley vd. (2000)'de olduėu gibi karma bir tam sayı problemi řeklinde formle edilir. Hesaplama sonuları OR kitaplıėındaki deėerler iin ve rastgele oluřturulmuř 75 uaėı ve 4 pisti ieren deėerler iin sunulmuřtur.

2. HAVA TRAFİK KONTROLÜ

Hava trafik kontrolü belirlenen bölgedeki hava taşıtlarının uçuş, iniş, kalkış planlamalarının tamamını kapsar. Bu planlamaların içerisinde birden fazla düzey bulunmaktadır. Uçakların kalkıştan uçuşa, uçuştan inişe olan geçişlerinin güvenli bir şekilde sağlanması için gerekli düzenlemeler yapılır. Bu güvenliği sağlayabilmek için çeşitli hesaplamalarla kalkışlar, inişler ve uçuşlar arasındaki süre ve mesafeler belirlenir. Bu bilgiler havayolu şirketleri ve pilotlarla paylaşılarak uçuş trafiğinin güvenliği sağlanır. Uçakların gerektiği durumlarda rota değişiklikleri, iniş yapacakları pistlerin değişimi hatta bazı durumlarda iniş yapacakları havaalanlarının değişimi hava trafik kontrolü görevleri arasındadır. Havayolu şirketleri uçakların gerçek kalkış zamanlarını bildirdikten sonra iniş havalimanındaki iniş zamanı ve iniş sırası gibi işlemler için hesaplamalar başlar. Ancak uçuş sırasında da beklenmeyen durumlardan kaynaklı varış zamanında değişiklikler olabilir. İniş zamanı ve sırası hesaplanırken bu tip değişiklikler de göz önünde bulundurulur. Bu iniş sıralamasının belirlenmesi de uçakların zaman kaybetmelerini önlemek açısından son derece önemlidir.

Şu anki sistemde uçaklar için iniş sıralamaları varış havalimanlarına 200 deniz mili ya da 45 dakikalık zaman kalmışken yapılır (Hansen 2004). Bu planlamanın önceden yapılması uçakların havada harcayacakları zamanın en aza indirilmesi açısından önemlidir. Aynı zamanda pistlerin etkin ve dengeli şekilde kullanımı, güvenlik kısıtlamalarının dikkate alınması ve doğabilecek aksaklıkların önceden kestirilmesi açısından önem arz etmektedir.

Trafiğin yoğun olduğu havalimanlarında bu iniş kalkış trafiğini daha etkili ve hızlı hale getirebilecek bir sistemin geliştirilmesi uzun zamandır havacılık araştırmalarının gündemindedir. Hava trafiğinin günden güne büyümesi, iniş ve kalkış sırasındaki gecikmelerinin sürekli artması bu tip bir sisteme olan ihtiyacı gitgide arttırmaktadır.

2.1 Dinamik Planlayıcı

Dinamik planlayıcılar(DP) terminal alanı içerisinde trafik akışının etkililiğini sağlamak için uygun olan sıralamaları, varış zamanlarını ve uçuş pisti tahsislerini hesaplarlar. Normal durumlarda uçaklar ilk gelene ilk hizmet prensibiyle iniş kalkış işlemlerini gerçekleştirmektedir. Ancak TMC'ler ellerindeki kısıtlamaları değerlendirerek bu sıralamalarda değişikliklere gidebilirler.

2.2 Dinamik Planlayıcı Girişleri

DP girişlerinden başlıca olanları aşağıda özetlenmiştir:

- Uçuş planları
- Radar takibi güncellemeleri
- Referans nokta tahmini varış zamanı (TVZ'ler)
- Üst geçiş zamanları
- Liste kısıtlamaları
- Sıralama kısıtlamaları

2.2.1 Uçuş planları

Uçuş planları merkezin hava sahası içerisinde olan ve hava sahasına girmek üzere olan uçaklar için iniş kalkış açısından önemli bilgiler içermektedir. Havayolu şirketleri merkeze bu uçuş planlarını sunmaktadır. Merkezdeki ana bilgisayarlar bu planları işleme sokar.

DP, CTAS tarafından işleme alınan her bir uçak için bir uçuş planına ihtiyaç duyar. DP her bir uçak için aşağıdaki uçuş planı bilgilerini kullanır:

- Uçağın tanımlanması
- Uçağın türü ve özellikleri
- Uçağın planlanan uçuş rotası

- Uçağın Merkezin hava sahasına tahmini giriş zamanı
- Uçuş planı durumu

Aşağıdakiler ise DP'nin merkezin ana bilgisayarından aldığı olası uçuş planı durumu belirleyicileridir:

- Tahmini uçuş planı: Uçak merkezin hava sahasına bitişikteki bir Merkezden geçiş yapmaktadır.
- Tasarlanan uçuş planı: Uçağın Merkezin hava sahası ile birlikte bir hava alanından ayrılacağı tahmin edilmektedir.
- Ayrılan uçuş planı: Uçak Merkezin hava sahası ile birlikte bir hava alanından ayrılmaktadır.

2.2.2 Takip güncellemeleri

CTAS merkezin ana bilgisayarından, merkezin hava sahasında bulunan uçaklarla ilgili radar takip bilgilerini alır. Radar takip bilgileri alınan bu uçakları aktif uçaklar olarak tanımlamaktadır. DP hem aktif olan hem de aktif olmayan uçaklar için uçuş planlarını hazırlar.

2.2.3 Tahmini varış zamanları (TVZ'ler)

TMA'ya ait Rota Analizcisi (RA) ve Rota Birleştiricisi (TS) programları, Şekil 2.1'de gösterilen her bir referans noktasına Tahmini Varış Zamanlarını (TVZ'leri) DP'ye sağlamak için birlikte çalışırlar. RA varmak üzere olan uçakların yatay rotalarını ve aynı zamanda rota doğrultusunda çeşitli noktalarda çeşitli hız kısıtlamalarını hesaplar. TS daha sonrasında bu bilgiyi uçak ile ve hava durumu modelleri ile birlikte alır ve uçağın mevcuttaki yerinden uçuş pistindeki değme noktasına kadar olan 4 boyutlu bir rota hesaplar. Bu rotadan, RA referans noktaları için TVZ'leri çıkartabilir ve bu TVZ'leri DP'ye gönderir. TVZ'ler her bir radar güncellemesi için tekrardan hesaplanır (Wong 2000).

2.3 Dinamik Planlayıcıdan Çıktılar

DP özetlenen tüm bu işlemleri yaparak aşağıdaki çıktıları üretir.

- Planlanan Varış Süreleri(PVS'ler)
- Uçuş Pisti Tahsisleri

2.4 İlk Gelene İlk Hizmet Prensibi(İĞİH)

Uçaklar hem inişlerde hem de geçişlerde ayırma zamanları dikkate alınarak ilk gelene ilk hizmet prensibiyle sıralanırlar. Piste inişte olması gereken uzaklık kriterleri uçağın ağırlık sınıfı ve FAA tarafından güvenli iniş için belirlenen diğer özelliklerine bakılarak bulunur. Bir merkeze gelen uçakların İĞİH sıralaması havayolu şirketleri tarafından belirtilen TVZ'lerine göre belirlenir.

3. PROBLEM ÜZERİNDE DAHA ÖNCE YAPILMIŞ ÇALIŞMALAR

Problem üzerinde daha önce birçok algoritma denenmiştir. Literatürde en çok yer eden çalışmalar genetik algoritma ile yapılan çözüm önerileridir. Tez içerisinde de genetik algoritmanın problem üzerine uygulanışı anlatılacaktır.

3.1 Genetik Algoritma ile Problemin Optimizasyonu

Merkez hava sahalarına varış yapan uçakların iniş sıralama ve zamanlamaları hava trafik kontrol sisteminde önemli bir rol oynamaktadır. Pistlere iniş yapmak için kuleden izin alan uçaklar, ilk gelene ilk hizmet prensibiyle belirli bir sıralamaya tabi tutulurlar. Fakat bu yöntem ciddi şekilde gecikmelere neden olmaktadır. En uygun ve optimal sıralamayı bulmak için farklı yöntemler kullanılabilir. Bu yöntemlerden biri de genetik algoritmalarıdır. Genetik algoritmalar tüm sıralama ihtimallerini değerlendirip en uygun sıralamayı seçmeyi sağlamaktadır. Genetik algoritmanın çalışması için bir uygunluk fonksiyonu seçmek gerekir. Algoritma seçenekleri bu uygunluk fonksiyonuna göre değerlendirir ve uygun bireyleri seçer. Bahsettiğimiz durumda iki farklı uygunluk fonksiyonu ortaya çıkmaktadır. Bu fonksiyonlardan biri tahmini varış süresi ile planlanan varış süresi arasındaki farkı minimize eden uçakları seçmeyi amaçlamaktadır. İkinci uygunluk fonksiyonu ise tüm sıralama gerçekleştiğinde ortaya çıkan toplam bekleme süresini minimize etmeyi amaçlamaktadır.

3.1.1 Genetik algoritma

Genetik algoritmalar uygun bir bölgede uygun sonuçları bulup o sonuçlar üzerinden en uygun sonuca ulaşma konusunda oldukça etkilidir. Genetik algoritmanın temelindeki düşünce Charles Darwin tarafından ortaya konan evrim teorisine dayanır. Bazı kötü ve iyi genlere sahip bireylerden oluşan bir topluluk vardır. Bu topluluktan iki kişi bir çocuk sahibi olduğunda oluşan birey genlerini anne ve babasından alır. Bazı durumlarda çoğunlukla iyi genler çocuğa aktarılır, ancak bazı durumlarda bu gerçekleşmeyebilir. En güçlü olanın yaşamını sürdürmesi yaklaşımı kullanıldığında sadece en iyi genlere sahip

çocuk yaşamını sürdürebilecek ve bir sonraki nesli oluşturabilecektir. Bazen genlerin mutasyona uğraması durumu gerçekleşebilir. Bu mutasyon genlerin daha iyi veya daha kötü hale gelmesiyle sonuçlanabilir, sadece iyi genlere sahip bireyler evrimleşmeye devam eder. Bazı bireylerin yaşam ortamındaki değişimler sonucu hayatta kalması genlerinde meydana gelen mutasyon sonucu mümkün olmaktadır. Genetik algoritmalarda mutasyon daha iyi sonuçlar oluşturmak için kullanılır. Bu düzenle nüfus zamanla uygun bireylerle artar ve sonunda en iyi bireyin oluşması umut edilir. Bu en iyi birey algoritmamızın sonucunda ortaya çıkan optimal sonuçtur.

Genetik algoritmanın ilerleyişi başlangıç nüfusu ile başlar, sonrasında rastgele veya belirli bir ölçüde göre ebeveynler seçilir ve bu bireylerin genlerinin birleştirilmesi ile bir veya birden fazla çocuk birey oluşturulur. Sonrasında nüfusun tamamı veya bir kısmı yeni oluşan bireylerle değiştirilir ve algoritma yeniden başlatılır. Bu belirlenen bir durma kriteri oluşana kadar devam ettirilir. Bu durma kriteri belirli bir ilerleme sayısı, belirli bir işlemci zamanı veya belirli bir ilerleme sonucu bulunan çözümün geliştirilememesi gibi kriterler olabilir (Emel ve Taşkın 2002).

Bir probleme genetik algoritmayı uygulamadan önce yapılması gereken seçimler vardır;

- Nüfus miktarı
- İlk nüfusun oluşturulması
- Ebeveynlerin nasıl seçileceği
- Ebeveynlerin genlerinin çocuk bireyi oluşturmak için nasıl birleştirileceği
- Bireylerin nüfus içerisinde nasıl değiştirileceği
- Bir çözümün nasıl mutasyona uğratılacağı
- Durma kriterinin ne olacağı

Daha önceki araştırmalar göz önüne alındığında genetik algoritmalar uçak iniş problemlerinin çözümünde kullanılan en yaygın algoritmalarıdır. Genetik algoritmaların diğer tekniklerden farkları göz önünde bulundurulduğunda aşağıdaki avantajları göze çarpmaktadır.

- Problemin çözümü için çok sayıda değişkeni değerlendirebilir.
- Yapısı itibariyle paralel bilgisayarlarda çalıştırılmaya uygundur.
- Maliyet yüzeylerinden alınmış geniş örnekler üzerinde aynı anda arama yapar.
- Son derece karmaşık maliyet yüzeyleri olan değişkenleri optimize eder.
- Sonucunda tek bir çözüm değil, optimum değişkenler listesi sunar. Bu duruma göre hedeflenen amaca en uygun olanı seçme olanağı tanır.
- Sayısal olarak oluşturulan verilerle, deneysel verilerle veya analitik verilerle çalışabilir.

3.1.2 Örnek çalışma

İlgili makale (Beasley vd. 2000) Birleşik Krallık Ulusal Hava Trafik Hizmetleri tarafından yapılan Londra Heathrow Havalimanındaki pist kullanım oranlarını arttırmaya yönelik araştırmanın sonuçlarını değerlendirmiştir. Kullanılan algoritma genetik algoritma tabanlıdır. Problem genelden özele indirildiği için algoritma içerisine birçok parametre eklenmiştir. Bunlardan bazıları uçak tipleri arasında iniş sırasındaki ayırma zamanları ve uçakların iniş hızları gibi değerlerdir.

Birçok veri setleri üzerinde algoritma denenmiş ancak bir setle yapılan testlerin sonuçları yayın içerisinde verilmiştir. Havalimanının 100 deniz mili çevresinde olan 20 uçaklı bir veri seti üzerinde yapılan testler yayınlanmıştır.

Uygunluk fonksiyonu

Uygunluk fonksiyonu toplam uygunluk değerini maksimize etmeye çalışır.

$$\sum_{i=1}^P Z_i , \quad Z_i = \begin{cases} -(D_i)^2 & \text{if } D_i \geq 0 \\ +(D_i)^2 & \text{otherwise} \end{cases}$$

$$D_i = (x_i - T_i)$$

Burada görüleceği üzere uçağın hedeflenen zamandan erken inmesi tercih edilen durumdur.

Uygunsuzluk değerinin hesaplanması için ise aşağıdaki fonksiyon kullanılır.

$$\sum_{i=1}^P \sum_{j=1}^P \max[0, S_{ij} - (x_j - x_i)]$$

Bir çözüm uygulanabilir ise tüm kısıtlamalar sağlandı demektir. Bu da aynı zamanda uygunsuzluk değerinin sıfır olduğu anlamına gelir.

Başlangıç Nüfusu

Başlangıç nüfusunun 100 bireyden oluşur. Tüm çözümler ise rastgele oluşturulur.

Ebeveyn Seçimi

Öncelikle rastgele iki birey seçilir. Bu bireylerden daha yüksek uygunluk değerine sahip olan ilk ebeveyn olarak seçilir. Sonrasında ise yine iki rastgele birey seçilir ve bunlardan daha yüksek uygunluk değerine sahip olan birey ikinci ebeveyn olur.

Çocuk Seçimi

Seçilen ebeveynlerden bir çocuk birey oluşturmak için tekdüze genetik değişim uygulanır.

Nüfus Değişimi

Yukarıdaki bölümlerde anlatıldığı üzere bir çocuk oluşturulduğu zaman onu nüfus içerisine koymak için bir birey nüfustan çıkarılır ve yerine yeni oluşturulan çocuk koyulur. Oluşan yeni çocuğa göre mevcut nüfus 4 ayrı grupta sıralanır.

1. Grup (G1)

Bu grup nüfus içerisindeki en kötü uygunluk değerine ve en kötü uygunsuzluk değerine sahip bireylerin yerleştirilmesiyle oluşturulur.

2. Grup (G2)

Bu grupta ise uygunluk değerleri daha iyi olan ama uygunsuzluk değerleri daha kötü durumda olan bireyler bulunur.

3. Grup (G3)

Bu gruptaki bireyler ikinci grubun aksine daha kötü uygunluk değerine ancak daha iyi uygunsuzluk değerine sahip bireylerdir.

4. Grup (G4)

Bu grup daha iyi uygunluk değerine ve daha iyi uygunsuzluk değerine sahip bireylerin yerleştirilmesiyle oluşturulur.

Sonrasında algoritma ilk gruptan rastgele bir birey seçer. Algoritma ilk gruptan başlayarak sırasıyla gruplara bakar.

Sonuçlar

Bu makalede sunulan çözümle sonuçlar % 2-5 arasında iyileştirilebilir olarak belirlenmiştir. Ortalama gecikme 49 saniye azaltılmıştır ancak en yüksek gecikme süresi 187 saniye daha fazla olmuştur.

3.2 Agözlü Algoritma ile Optimizasyon

Optimizasyon problemleri için birçok farklı algoritmalar mevcuttur. Bu algoritmalarından biri olan agözlü algoritmanın özellikleri bu bölümde daha detaylı anlatılacaktır.

3.2.1 Agözlü algoritma

Agözlü algoritma basit ve anlaşılır bir algoritmadır. Bu algoritma yapısı itibariyle ileri görüşlü değildir. O an için elde olan verilerle en iyi kararı almaya çalışır. O an alınan kararın daha sonraki aşamalara edeceği etkiyi dikkate almaması bu algoritmanın en belirgin özelliğidir. Anlaşılması ve uygulaması kolay bir algoritmadır. Çoğu zaman etkili çözümler sunar. Agözlü algoritmalar her zaman doğru sonucu vermez. Bu yapısı dolayısıyla daha çok optimizasyon problemlerinde kullanılır.

Agözlü algoritmanın çalışma yapısını anlatmak için bir örnek olarak para üstü verme problemi düşünülebilir. Agözlü yaklaşım o an için en doğru görünen kararı vererek ilerler. Elimizdeki para miktarlarının aşağıdaki gibi olduğunu varsayalım:

- 1 TL (100 kuruş)
- 25 kuruş
- 10 kuruş
- 5 kuruş
- 1 kuruş

3.28 TL'lik bir ücreti tamamlamak için agözlü algoritma elimizde olan ve istenen miktarı geçmeyen en büyük parayı çözüme ekleyerek ilerler. Örnek ücretin altında olan ve elimizde olan en yüksek para 1 TL'dir. Algoritma 3 tur 1 TL seçerek 3.28 TL'nin 3 TL'sini tamamlar. Geriye kalan 28 kuruşluk kısım için önce 25 kuruş arkasından ise 3 kez 1 kuruş seçer ve sonuca ulaşmış olur.

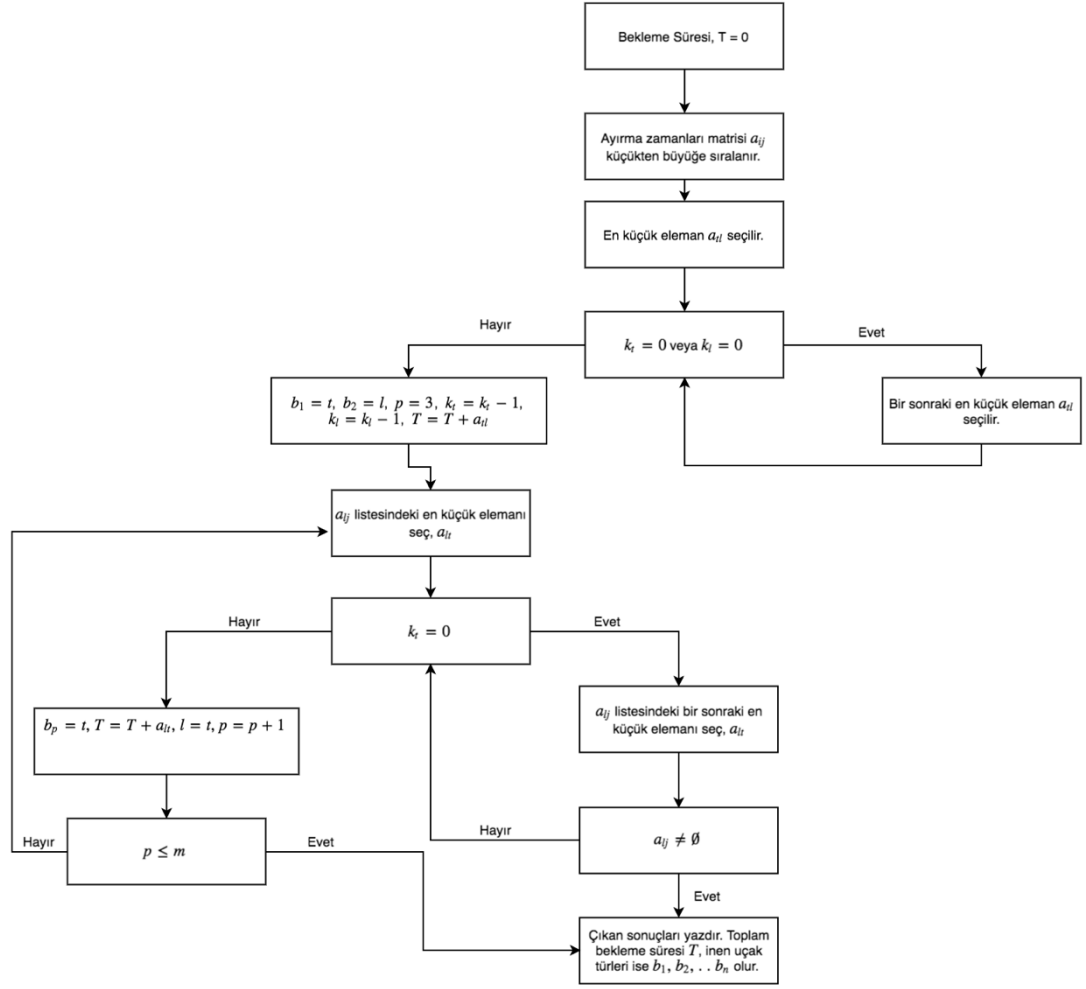
3.2.2 Örnek çalışma

Çizelgeleme probleminde de açgözlü algoritma kullanılabilir. Her adımda ortaya çıkan birkaç alternatiften o an için en iyi olan seçilerek algoritma yoluna devam eder. (Beasley vd. 2000) makalesinde hava trafik kontrolünün alt problemlerinden olan iniş sıralaması probleminin açgözlü algoritmayla nasıl çözüleceği anlatılmıştır.

Problemi özetleyecek olursak; n çeşit uçak tipi vardır. İnen uçaklar arasında geçmesi gereken minimum bir ayırma süresi mevcuttur. Yani i tipindeki bir uçağın arkasından aynı pise j tipinde bir uçak inmesinden önce geçmesi gereken süre a_{ij} 'dir. a_{ij} pozitif bir değerdir. Çoğu durumda $a_{ij} \neq a_{ji}$ 'dir. Bu da demektir ki bekleme süresi matrisi simetrik değildir.

Problemi matematiksel olarak ifade edersek; T değeri i . çeşit uçak inene kadar geçen süredir. Uçak tiplerine göre uçak sayılarını k olarak adlandıralım. Yani k_1 birinci çeşit uçak sayısını, k_2 ikinci çeşit uçak sayısını k_n n . çeşit uçak sayısını belirtsin. Bu şekilde $k_1 + k_2 + \dots + k_n = n$ olur. k değerleri 0 veya 0'dan büyüktür. Uçakların türlerine göre havada bekleyebilme süreleri bulunmaktadır. Bu bekleyebilme sürelerini b_l olarak adlandıralım. Algoritma akışı aşağıdaki diyagramda görselleştirilmiştir.

İlk problemdeki uçakların üzerine belirli bir zamanda r sayısı kadar daha uçak eklendiğini varsayalım. Bu uçakların çeşitleri ise q_1, q_2, \dots, q_n olsun. Dolayısıyla $q_1 + q_2 + \dots + q_n = r$ olur. q_i sıfır veya daha büyük değerler alır. Uçaklar eklendikten sonra havadaki toplam uçak sayısı $m + r - 1$ olur.



Şekil 3.1 Açgözlü algoritma akış şeması

Algoritmayı aşağıdaki örnek üzerinden inceleyelim;

Örneğimizdeki uçak çeşidi sayısı 3 olsun. Uçak çeşitleri arasındaki minimum ayırma süresi matrisi şu şekildedir.

$$\begin{bmatrix} 1 & 5 & 1 \\ 4 & 2 & 3 \\ 3 & 17 & 4 \end{bmatrix}$$

Havada bekleyen uçakların geliş sıralarına göre çeşitleri ise 1, 1, 3, 2, 3, 2, 2 şeklindedir. Uçaklar geliş sıralarına göre indirilirse toplam geçmesi gereken bekleme süresi aşağıdaki şekilde hesaplanır.

$$T = 1 + 1 + 17 + 3 + 17 + 2 = 41$$

Yukarıda anlatılan açgözlü algoritma uygulandığında ise uçakların sıralaması 1, 1, 3, 3, 2, 2, 2 olacaktır. Toplam bekleme süresi ise $T=1+1+4+17+2+2=27$ olur.

Bu örnekte açgözlü algoritma İGİH prensibinden daha iyi sonuç vermektedir. Öte yandan uçakların iniş sıralaması 1, 1, 2, 2, 2, 3, 3 şeklinde olsaydı $T=1+5+2+2+3+4=17$ olurdu. Dolayısıyla açgözlü algoritma sonucu iyileştirmesine rağmen en iyi sonucu vermemiştir.



4. ÖNERİLEN YENİ ALGORİTMA

3.2.1 bölümünde anlatılan açgözlü algoritma mantığı sürekli en küçük değeri bulup onun üzerinden ilerlemek üzerine geliştirilmiştir. Algoritma yeterince hızlı sonuç vermektedir. Her zaman olmasa da çoğunlukla sunduğu çözümler İGİH prensibinden daha iyi performans sergilemektedir. Ancak algoritma sadece uçak tipleri arasındaki ayırma zamanlarını göz önüne almaktadır. Yukarıdaki bölümlerde bahsedilen bir çok değişken bu algorithmada göz önüne alınmamıştır.

3.2.1 bölümünde anlatılan açgözlü algoritmaya başka değişkenler ekleyerek yine aynı prensiple sonuçlar elde edilebilir. Anlatılan problemin en önemli değişkenlerinden en geç iniş zamanı(EGİZ) değişkenini algoritmanın girdileri arasına ekleyebiliriz. Buna göre belli bir anda havada bekleyen her uçağın tipi ve en geç iniş zamanı algoritmamızın girdileri olacaktır.

Algoritmanın genel çalışma mantığı aynı bırakılarak bazı ek kontroller eklenecektir. Bu ek kontroller sayesinde uçak iniş sıralaması hesaplanırken uçaklar kendilerine ait en geç iniş zamanlarından daha önce piste indirilmeye çalışılacaktır.

Belirli bir zamanda 3.2.1 bölümündeki açgözlü algoritma çalıştırılıp aday bir indirilecek uçak belirlenecektir. Sonrasında o an için EGİZ'e ulaşan bir uçak olup olmadığı kontrol edilecektir. Eğer EGİZ'e ulaşan bir uçak yoksa aday uçakla kontrolün ikinci aşamasına geçilecektir. İkinci kontrolde ise aday uçağın indirildiği durum göz önünde bulundurulup onun inmesinden sonra kaynaklanan gecikmenin herhangi bir uçağın EGİZ'i geçmesine neden olup olmayacağı kontrol edilir. Eğer ikinci kontroldeki duruma uyan herhangi bir uçak bulunursa aday uçak yerine duruma uyan ilgili uçak iniş sıralamasına eklenir. Eğer birinci kontrol sonrasında duruma uyan bir uçak bulunursa, ikinci kontrole gerek duymadan duruma uygun olan o uçak sıralamaya eklenecektir.

Bir örnek üzerinden açıklamak gerekirse;

Belirli bir anda iniş sırası bekleyen 3 tip uçağın olduğunu varsayalım. Algoritmamızın ayırma zamanları matrisi aşağıdaki gibi olsun:

$$\begin{bmatrix} 1 & 5 & 4 \\ 3 & 7 & 2 \\ 6 & 4 & 1 \end{bmatrix}$$

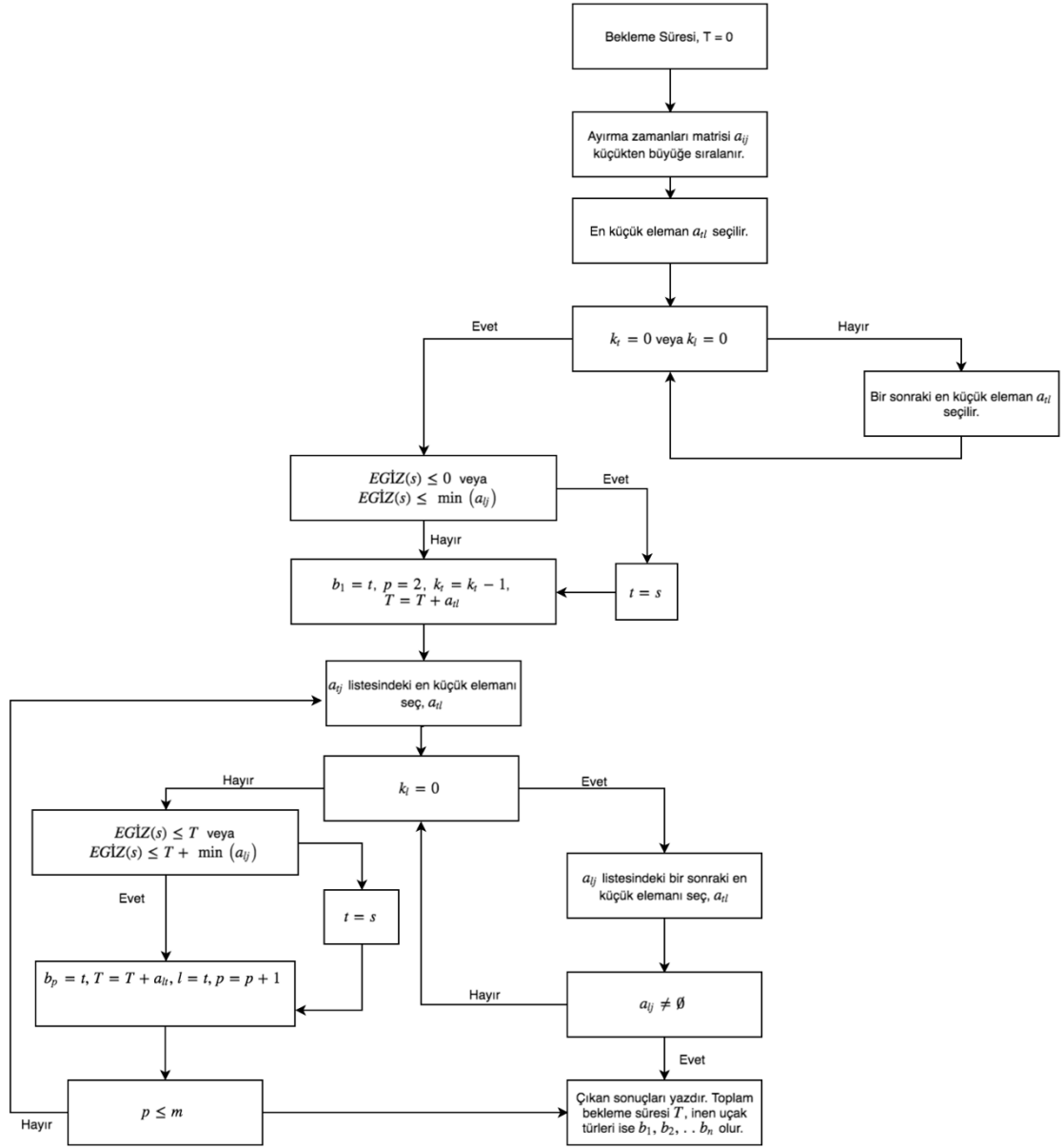
İniş sırası bekleyen uçaklar ise sırasıyla 1, 1, 2, 1, 3, 2, 3 şeklinde olsun. Geliştirilmiş açgözlü algoritmanın yeni girdisi olan EGİZ değerleri ise sırasıyla 5, 10, 7, 12, 18, 15, 20 şeklindedir. Eğer 5.2.1 bölümündeki algoritma uygulanırsa EGİZ değerleri göz önünde bulundurulmayacağı için çıkan sıralama 1,1,1,3,3,2,2 şeklinde olacaktır. Bu durumda birden fazla uçak kendi EGİZ değerinden sonra iniş yapacaktır. Bu ise istenmeyen bir durumdur.

Geliştirilen uygulama yukarıdaki örneğe uygulandığı zaman sıralama 1, 1, 2, 1, 2, 3, 3 şeklinde olacaktır. İniş Sırası - Uçak Tipi - Uçak EGİZ Değeri - Gerçekleşen Uçak İniş Zamanı şeklinde belirtecek olursak sıralama aşağıdaki gibi olacaktır.

1	1	5	0
2	1	10	1
3	2	7	6
4	1	12	9
5	2	15	14
6	3	18	16
7	3	20	17

Görüleceği üzere yine açgözlü algoritma mantığı kullanılmış ancak bu sefer fazladan bir parametre daha göz önünde bulundurulmuştur. İlk algoritma uygulandığı zaman tüm inişlerin gerçekleşmesi için geçmesi gereken süre 18 birim zaman olmuştur. Geliştirilmiş algoritma uygulandığı zaman ise geçmesi gereken süre 17 birim zaman olarak hesaplanmıştır. Burada toplam zaman anlamında da bir iyileştirme görülmektedir. Ancak bu iyileştirme algoritmanın daha optimize sonuç vermesi için tasarlanmasından kaynaklanmamaktadır. Test verilerine bağlı olarak daha iyi sonuçlar çıkabilir. Ancak algoritmanın asıl odak noktası bu değil EGİZ parametresinin dikkate alınarak iniş sıralamasının ortaya çıkarılmasıdır.

Önerilen yeni algoritmanın akış diyagramı şekil 4.1'deki gibi olacaktır.



Şekil 4.1 Geliştirilmiş Açgözlü Algoritma Akış Diyagramı

4.1 Genetik Algoritmanın Özelleştirilmesi

Daha önceki bölümlerde anlatılan ve sonraki bölümlerde simülasyonu yapılacak olan genetik algoritma karar verirken birçok değişkeni göz önünde bulundurmaktadır. Uçaklar hedefledikleri zamandan önce veya sonra inmeleri durumunda belirli cezalar

ödemek zorundadırlar. Algoritma bu cezalarının en aza indirilmesini uygunluk fonksiyonu olarak seçmiştir.

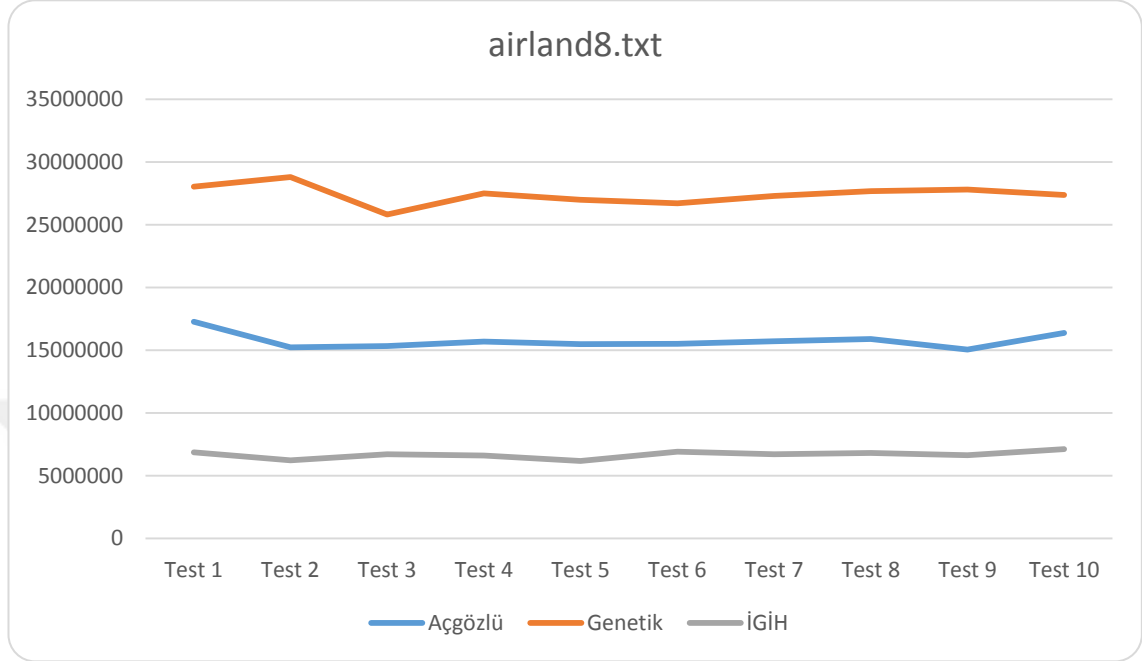
Önerilen algoritma ise inişleri en kısa sürede tamamlamayı hedefler. Bunu yaparken kullandığı kısıtlar uçakların en geç iniş zamanları ve aralarındaki ayırma zamanlarıdır. Bu iki algoritmanın karşılaştırmasını yapabilmek için kısıtlarını ve hedeflerini aynı yapmak gerekmektedir. Dolayısıyla genetik algoritmanın karşılaştırılması için öncesinde önerilen algoritmaya göre özelleştirilmesi gerekmektedir. Bu sayede karşılaştırma daha mantıklı bir hal alacaktır.

Önerilen algoritma uçakların algoritma başlamadan önce belirlendiğini kabul eder ve algoritma sürerken yeni uçak eklenmesi durumunu göz önüne almaz. Bu sebeple genetik algoritmanın kullanacağı verilerde de uçakların ortaya çıkma zamanı sıfır olarak kabul edilecektir. Yani uçakların hepsi algoritma başlangıcında ortaya çıkmış ve daha sonra uçak eklenmemiş şekilde çalışacaktır. Aynı zamanda önerilen algoritma hedeflenen zaman kısıtını da göz önünde bulundurmamaktadır. Bu sebeple genetik algoritmanın uygunluk fonksiyonu değiştirilerek hedef zamana göre değil, inişlerin en kısa sürede tamamlanmasına yönelik geliştirilmiştir.

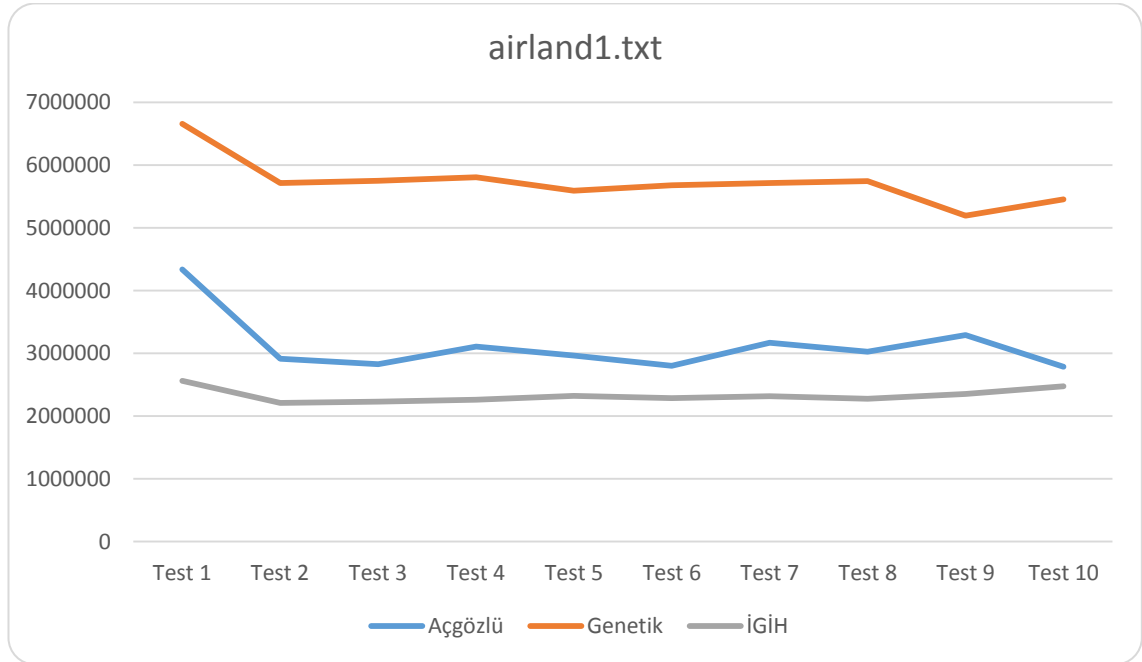
4.2 Önerilen Algoritmanın Karşılaştırılması

Önerilen ağgözlü algoritma benzer girdileri değerlendirmesi için özelleştirilmiş olan genetik algoritmayla karşılaştırılabilir. Karşılaştırma algoritmaların çalışma zamanları açısından ve uçak inişlerini gerçekleştirme süreleri açısından yapılacaktır. Karşılaştırma için kullanılan veriler algoritmanın 2,5 GHz Intel Core i7 işlemciye sahip bir bilgisayar üzerinde çalıştırılmasıyla elde edilmiştir. Genetik algoritma aynı veri setinde her çalışmada farklı sonuçlar verebileceğinden bu gibi durumlarda verdiği sonuçların ortalama değeri karşılaştırma için kıstas olarak alınmıştır. Genetik algoritmanın başlangıç nüfus sayısı ve çaprazlama sayısı yapılan denemelerle optimuma çekilmeye çalışılmıştır. Test ederken tüm uçaklar sıfır zamanında iniş için hazır varsayılacağından yani test dosyasındaki ortaya çıkma zamanları 0 alınacağından EGİZ değerleri de ortaya çıkış zamanları kadar aşağı çekilmiştir. Aksi takdirde EGİZ değeri çok yüksek

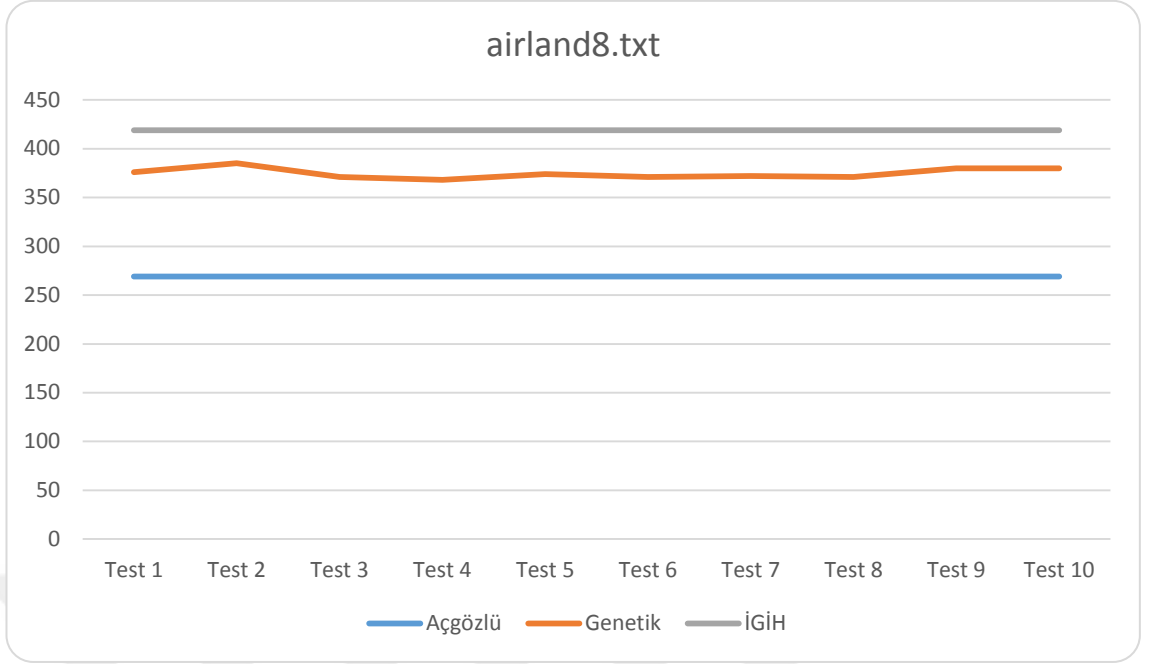
olacağından algoritmaların EGİZ'e göre çalışma performanslarını görmek mümkün olmayacaktır. EGİZ değeri aşağı çekilerek bu mümkün kılınmıştır.



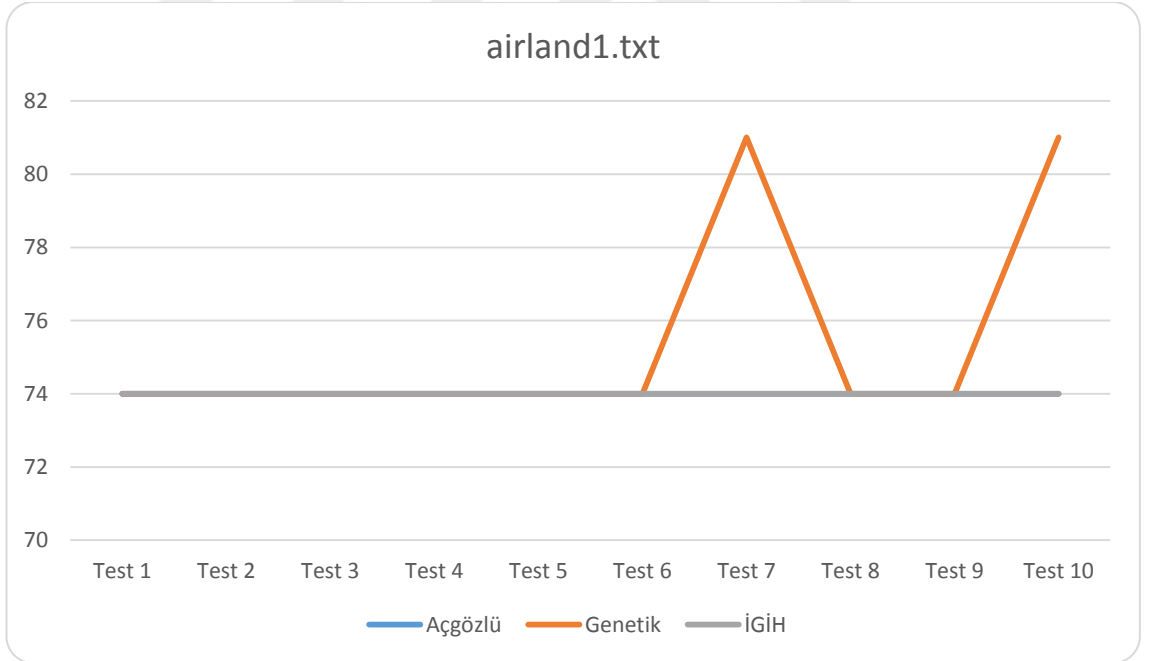
Şekil 4.2 Çalışma zamanı kıyaslaması (nano saniye)



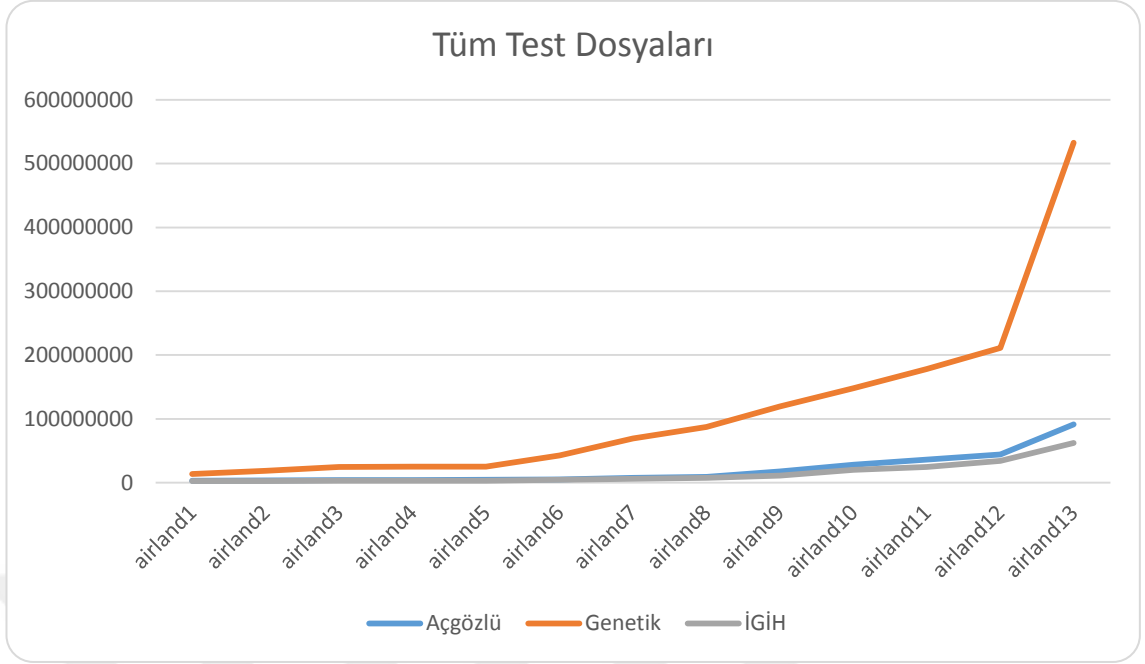
Şekil 4.3 Çalışma zamanı kıyaslaması (nano saniye)



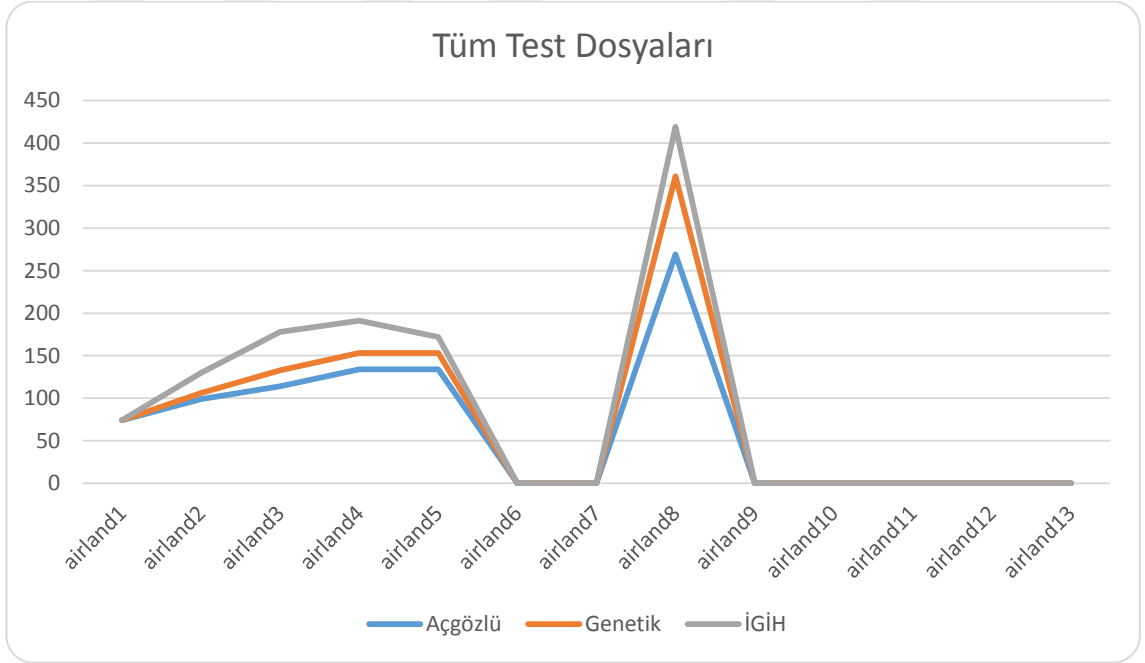
Şekil 4.4 Toplam iniş tamamlanma süresi



Şekil 4.5 Toplam iniş tamamlanma süresi



Şekil 4.6 Tüm test dosyalarının çalışma zamanı kıyaslaması (nano saniye)



Şekil 4.7 Tüm test dosyalarının toplam iniş tamamlanma süresi

Çizelge 4.1 Tüm test dosyalarında algoritmaların çalışma süreleri (milisaniye)

	Açgözlü	Genetik	İGiH
airland1.txt	3,085	13,848	2,790
airland2.txt	3,407	18,862	2,760
airland3.txt	4,136	24,538	3,205
airland4.txt	4,186	25,107	3,128
airland5.txt	4,554	25,284	3,155
airland6.txt	5,260	42,758	4,413
airland7.txt	7,435	69,524	6,146
airland8.txt	8,990	87,257	7,839
airland9.txt	17,801	119,699	11,152
airland10.txt	28,292	147,939	20,263
airland11.txt	36,083	178,338	24,765
airland12.txt	44,455	211,153	34,302
airland13.txt	91,463	533,061	62,435

Çizelge 4.2 Tüm test dosyalarının toplam iniş tamamlanma süreleri

	Açgözlü	Genetik	İGiH
airland1.txt	74	74	74
airland2.txt	99	106	130
airland3.txt	114	133	178
airland4.txt	134	153	191
airland5.txt	134	153	172
airland6.txt	-	-	-
airland7.txt	-	-	-
airland8.txt	269	361	419
airland9.txt	-	-	-
airland10.txt	-	-	-
airland11.txt	-	-	-
airland12.txt	-	-	-
airland13.txt	-	-	-

4.3 Algoritma Karşılaştırmalarının Sonuçları

Algoritmalar aynı girdilerle gerekli karşılaştırma bilgilerin toplanması için çalıştırılmıştır. Test verileri olarak uçak iniş zamanı algoritmalarının kullandığı Operations Research(OR) kütüphanesindeki(Beasley, 1990) dosyalar kullanılmıştır. Çıkan sonuçlar değerlendirildiğinde genetik algoritmanın çalışma zamanı açgözlü algoritma ve ilk gelene ilk hizmet algoritmasına göre çok daha yüksektir. Test dataları üzerinde genetik algorithmada popülasyon başlangıcı 50000'e kadar çıkarılmıştır ancak çözebildiği test dosyası sayısı açgözlü algoritmayla aynı olmuştur. Üç algoritma da 13 test dosyasından 6'sı için kriterlere uyan sonuç bulmuştur ancak 7 test dosyası için uygun sonuç bulunamamıştır.

Açgözlü algoritma ve İGİH algoritması her çalışmada aynı test verileri için aynı sonuçları vermektedir. Ancak genetik algoritma her çalışmada sonuçlarında küçük de olsa değişiklikler göstermektedir. Üç algoritmadan en hızlı çalışanı İGİH olmuştur. Ancak İGİH sonuçları çoğu test dosyasında iki algoritmanın verdiği sonuçlardan daha kötü sonuçlar vermiştir.

Test dosyalarından 1, 2, 3, 4, 5 ve 8. dosyalarda hem önerilen algoritma hem de genetik algoritma sonuç vermiştir. Önerilen açgözlü algoritma test dosyaları üzerinde genetik algorithmaya göre sırasıyla % 0, % 6.6, % 14.2, % 12.4, % 12.4, % 25.4 daha iyi sonuç vermiştir. Ortalamada ise önerilen açgözlü algoritma % 11.83 daha iyi sonuç vermiştir. İlk gelene ilk hizmet algoritması ile karşılaştırıldığında ise önerilen algoritma toplamda ortalama % 24.5 daha iyi sonuç vermiştir.

Çalışma zamanları kıyaslandığında ise önerilen algoritma genetik algorithmaya göre % 83.26 daha hızlı çalışmıştır. İlk gelene ilk hizmet algoritmasıyla kıyaslandığında ise önerilen algoritma % 32.08 daha yavaş çalışmıştır.

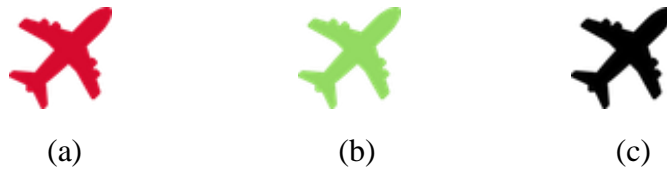
5. SİMÜLASYON

Daha önceki bölümlerde anlatılan genetik algoritma, aç gözlü algoritma ve geliştirilmiş aç gözlü algoritmanın bilgisayar programı üzerinde gösterimi sağlanacaktır. Bu simülasyon sayesinde algoritmalar dolayısıyla oluşan uçak iniş akışı daha iyi bir şekilde görülebilir. Simülasyonların oluşturulması için Java programlama dili kullanılmıştır. Arayüz elemanları için ise JavaFX kütüphanesi kullanılmıştır.

5.1 Genetik Algoritma Simülasyonu

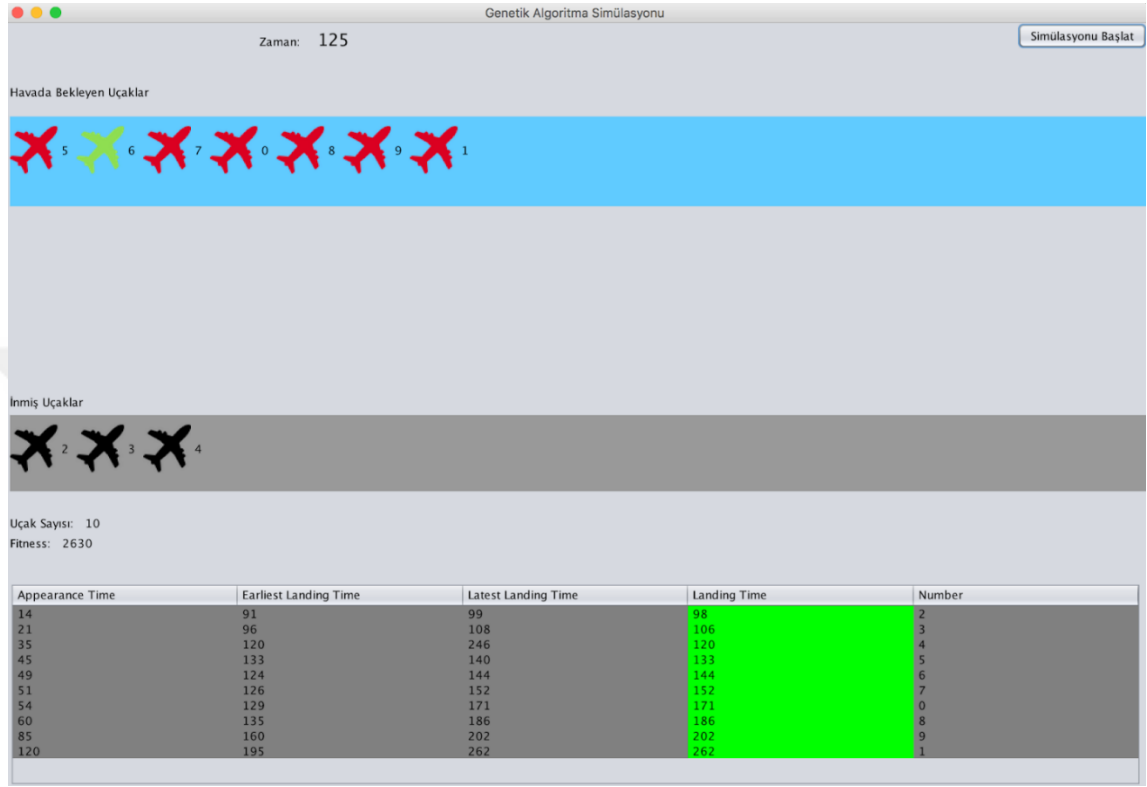
Simülasyon içerisinde kullanılan algoritma Erling Veidal'ın tezinde anlatımını yaptığı genetik algoritmadır. Simülasyonu test etmek için internette bulunan ve bu problem üzerine yapılan araştırmaların çoğunda test verisi olarak kullanılan OR-kütüphanesi içindeki bazı test dosyaları kullanılmıştır.

Simülasyonda havada bekleyen ve iniş yapmış uçaklar ayrı bölümlerde gösterilmiştir. Simülasyon başlatıldıktan sonra arayüzde gösterilen zaman birimi giderek artmakta ve o zamana karşılık gelen uçak inişleri, yeni uçakların bekleme kuyruğuna girmesi vs. simülasyon üzerinde gösterilmektedir. Havada bekleyen uçaklar için kırmızı ve yeşil uçak ikonları kullanılmıştır. Şekil 5.1 simülasyonda kullanılan ikonları göstermektedir. Havada bekleyen ve henüz iniş zaman penceresi içerisine girmeyen uçaklar için şekil 6.1 a'daki ikon kullanılmıştır. Havada bekleyen ve iniş zaman penceresi içerisinde bulunan uçaklar için ise şekil 5.1.b'deki ikon kullanılmıştır. İnişi gerçekleşen uçaklar için şekil 6.1.c'deki ikon kullanılmıştır.



Şekil 5.1 Simülasyonda kullanılan uçak ikonları

Simülasyon üzerinde ayrıca test verilerinde bulunan toplam uçak sayısı ve algoritma sonucu oluşan uygunluk değeri de gösterilmektedir. Simülasyonun rastgele bir zamanında alınmış ekran görüntüsü şekil 5.2’de gösterilmiştir.



Şekil 5.2 Genetik algoritma simülasyon ekran görüntüsü

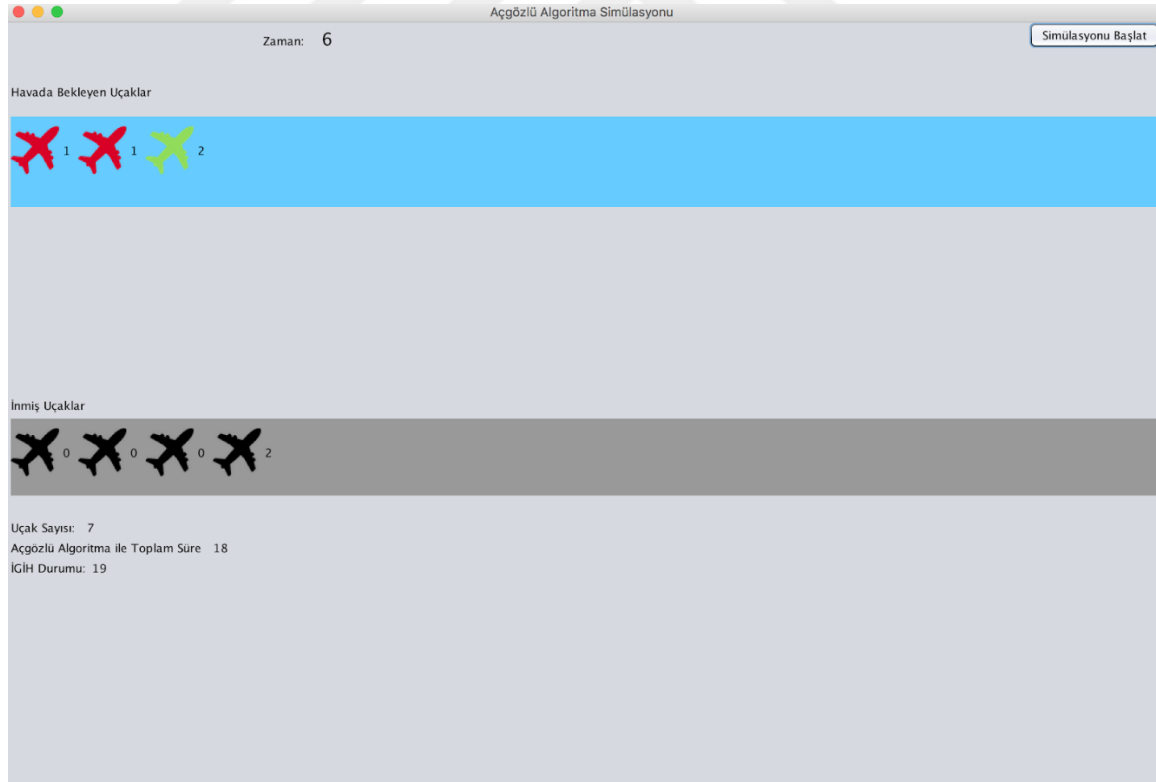
5.2 Açgözlü Algoritma Simülasyonu

Simülasyonu gerçekleştirmek için kullanılan algoritma (Emrah ve Alsalihe, 2011) makalesinde anlattığı açgözlü algoritmadır. Bu algoritmanın göz önünde bulundurduğu kriter uçak tipleri arasındaki ayırma zamanı kriteridir. Algoritma bunun dışındaki en erken iniş zamanı, en geç iniş zamanı gibi kriterleri göz önünde bulundurmamaktadır.

Simülasyonda havada bekleyen uçaklar ve iniş yapmış uçaklar iki ayrı bölümde gösterilmektedir. Uçak inişleri ayırma zamanları dikkate alınarak gerçekleştirilmektedir. Uçak ikonlarının yanında ilk simülasyondakinin aksine uçak numaraları değil uçak tiplerini belirten numaralar yer almaktadır.

Açgözlü algoritmada iniş için uygun zaman penceresi olmadığından ikonlar ilk algoritmaya göre farklı anlamlar belirtmektedir. Şekil 6.1 a'daki ikon belirttiği uçak tipiyle son inen uçak arasındaki ayırma zamanının yeterli olmadığını belirtir. Yani belirli bir an için o uçağın inmesinin mümkün olmadığını gösterir. Şekil 6.1 b'deki ikon ise son inen uçaktan sonra belirttiği uçağın inebilmesi için gerekli zamanın geçmiş olduğunu belirtir. Yani belirli bir an için o uçağın iniş yapabilecek uçaklardan biri olduğunu gösterir. Şekil 5.1.c'deki ikon ise ilk simülasyonda olduğu gibi inişi gerçekleşmiş olan uçağı belirtir.

Simülasyon üzerinde toplam uçak sayısı, açgözlü algoritma uygulandığı zaman inişlerin gerçekleşmesi için geçen toplam süre ve İGİH prensibi uygulanması durumunda inişlerin gerçekleşmesi için geçen toplam süre bilgileri de gösterilmiştir. Simülasyonun rastgele bir zamanında alınmış ekran görüntüsü şekil 5.3'te gösterilmiştir.



Şekil 5.3 Açgözlü Algoritma Simülasyon Ekran Görüntüsü

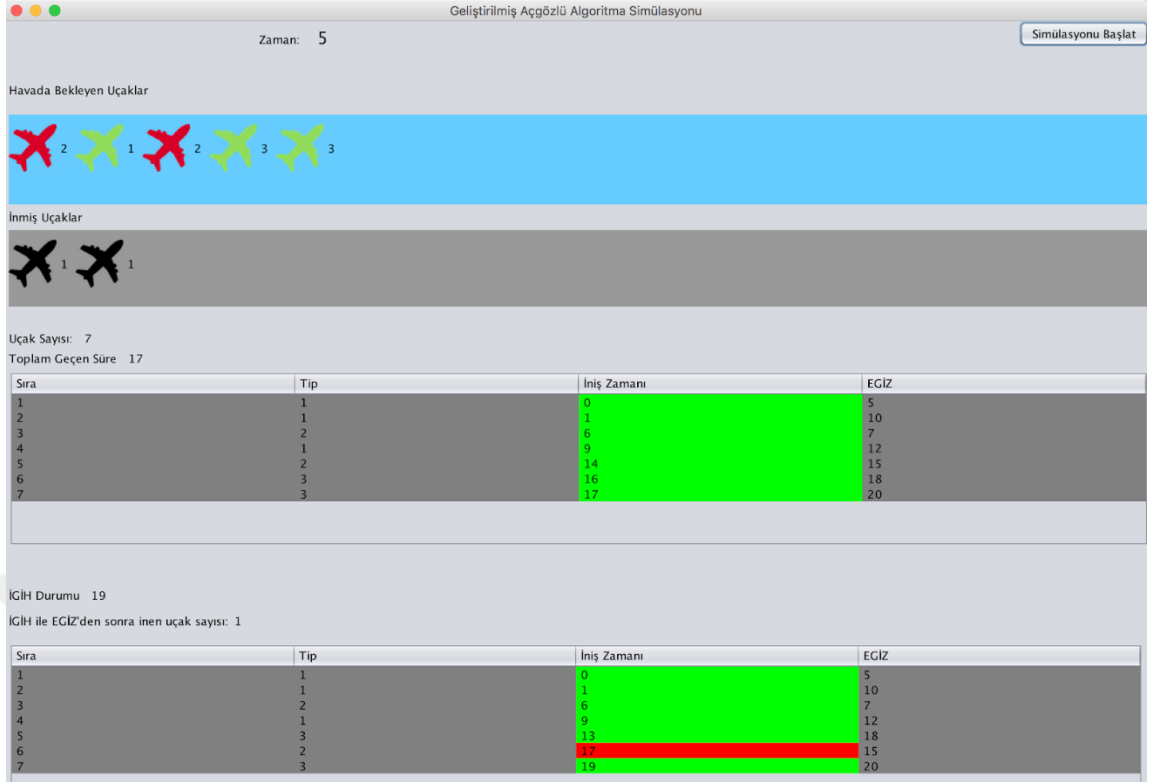
5.3 Geliştirilmiş Açgözlü Algoritma Simülasyonu

Simülasyonda kullanılan uygulama tez içerisinde anlatılan ve (Emrah ve Alsalihe, 2011) makalesindeki algoritmanın geliştirilmiş halidir. 5.2 bölümündeki algorithma göre görüleceği üzere göz önünde bulundurulmuş tek parametre ayırma zamanlarıdır. Geliştirilmiş açgözlü algoritma ayırma zamanlarının yanı sıra EGİZ değerlerini de göz önünde bulundurur.

Simülasyonda havada bekleyen uçaklar ve iniş yapmış uçaklar iki ayrı bölümde gösterilmektedir. Uçak inişleri ayırma zamanları ve EGİZ değerleri dikkate alınarak gerçekleştirilmektedir. Uçak ikonlarının yanında uçak numaraları yer almaktadır.

Simülasyondaki ikonlar 5.2 bölümündeki simülasyondaki ikonlarla aynı anlamları taşımaktadır. Yani Şekil 5.1.a'daki ikon belirttiği uçak tipiyle son inen uçak arasındaki ayırma zamanının yeterli olmadığını, 5.1.b'deki ikon son inen uçaktan sonra belirttiği uçak tipinin inebilmesi için gerekli zamanın geçmiş olduğunu, Şekil 5.1.c'deki ikon ise inişi gerçekleşmiş olan uçağı belirtir. Bu simülasyonda göze çarpan farklılık bazı durumlarda uçakların yeşil ikonla belirtilmesine rağmen iniş yaptırılmaması olacaktır. Bunun sebebi ise EGİZ değerlerinden dolayı o an açgözlü algorithma göre inmesi gereken uçağın iniş yaptırılması durumunda bazı uçakların EGİZ değerinden sonra iniş yaptırılacak olmasıdır. Algoritma bu durumu engellemek için iniş için uygun olsa bile uçağı bekletir ve EGİZ değerlerinden önce uçakları indirecek şekilde bir sıralama yapar.

Simülasyon üzerinde toplam uçak sayısı, geliştirilmiş açgözlü algoritma uygulandığı zaman inişlerin gerçekleşmesi için geçen toplam süre ve İGİH prensibi uygulanması durumunda kaç uçağın EGİZ değerinden sonra iniş yapacağı bilgileri de gösterilmiştir. Simülasyonun rastgele bir zamanında alınmış ekran görüntüsü şekil 5.4'te gösterilmiştir.



Şekil 5.4 Geliştirilmiş Ağgözlü Algoritma Simülasyon Ekran Görüntüsü

6. SONUÇ

Hava trafik kontrolünde uçuş sıralamasının belirlenmesi için çok fazla deęişken bulunmaktadır. Hali hazırdaki algoritmalar iyi sonuçlar vermesine rağmen geliştirilmesi ve daha fazla parametre kullanabilen algoritmaların sonuçlarının iyileştirilmesi gerekmektedir. Genetik algoritmalar aęgözlü algoritmalara göre daha fazla deęişkeni göz önüne alarak karar verebilme yeteneğine sahip olsalar da, sonuçları çok yararlı olmamaktadır. Aynı zamanda genetik algoritmaların sonuç verebilmesi için aęgözlü algoritmalara göre daha fazla zamana ihtiyacı olmaktadır.

Simülasyonu gerçekleştirilen genetik algoritmanın verdiği sonuçlar ayırma zamanı kriterlerine uymamaktadır. Ayırma zamanlarının uçak iniş sıralamasında en önemli kriterlerden biri olduđu göz önünde bulundurulursa bu durum bir çok probleme yol açabilecektir. Simülasyonu gerçekleştirilen ikinci algoritma olan aęgözlü algoritma ayırma zamanlarını göz önünde bulundursa da bir diđer önemli kriter olan en geç iniş zamanı kriterini göz önünde bulundurmamaktadır. Hiçbir uçağın sınırsız sürece havada kalamayacağı düşünülürse EGİZ kriteri de göz ardı edilmemesi gereken parametrelerdendir. Simülasyonu yapılan üçüncü algoritma ikinci simülasyondaki aęgözlü algoritmanın geliştirilmiş halidir. Hem ayırma zamanı kriterini hem de en geç iniş zamanı kriterini göz önünde bulundurarak bir sıralama oluşturmaktadır. Çıkan sonucun her durumda EGİZ değerlerine uyması beklenmemektedir ancak çođu durumda hem EGİZ hem de ayırma zamanı değerlerine uygun sonuç vermektedir. Bölüm 4.3'te belirtildiđi gibi önerilen algoritma uygulanan test verileri üzerinde genetik algoritmaya kıyasla ortalama olarak % 11,83 daha optimale yakın sonuçlar vermiştir. Önerilen algoritma ilk gelene ilk hizmet algoritmasına kıyasla % 24.5 daha iyi sonuçlar vermiştir. Çalışma zamanı kıyaslamasında ise en hızlı çalışan ilk gelene ilk hizmet algoritması olmuştur. Önerilen algoritma ise genetik algoritmaya göre 13 test dosyasının ortalamasında % 83.26 daha hızlı çalışmıştır.

Aęgözlü algoritmaların göz önünde bulundurduđu deęişkenler artırılmalıdır. Tezde anlatılan geliştirilmiş aęgözlü algoritma deęişken sayısını ikiye çıkarmıştır. Bu sayı ileride daha yükseğe çıkarılarak algoritmanın daha anlamlı sonuçlar üretmesi sağlanabilir.

KAYNAKLAR

Amrahov, S.E. and Alsalihe, T.A.I. 2011. Greedy algorithm for the scheduling aircrafts landings in Application of Information and Communication Technologies (AICT), 5th International Conference, Baku, Azerbaijan, 1-3.

Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y. M. and Abramson, D. 2000. Scheduling Aircraft Landings-the static case, Transportation Science, 34(2), 180-197.

Beasley, J. E. 2018. Web Sitesi: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/airlandinfo.html>, Erişim Tarihi: 10.05.2018.

Bianco, L., Dellolmo, P. and Giordani, S. 1999. Minimizing Total Completion Time Subject to Release Dates and Sequence Dependent Processing Times, Annals of Operations Research, 86, 393-415.

Ciesielski, V. and Scerri, P. 1998. Real time genetic scheduling of aircraft landing times, Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC98), 360-364.

Emel, G.G. ve Taşkın, Ç. 2002. Genetik Algoritmalar ve Uygulama Alanları. İktisadi ve İdari Bilimler Fakültesi Dergisi, Cilt: XXI, Sayı: 1, s. 129-152, Türkiye.

Ernst, A. T., Krishnamoorthy, M. and Storer, T. H. 1999. Heuristic and Exact Algorithms for Scheduling Aircraft Landings, Networks, 34, 229-241.

Hansen, J.V. 2004. Genetic search methods for air traffic control. Computers and Operations Research. 31:445–459.

Hillier, F.S. and Lieberman, G.J. 1994. Introduction to Operations Research, 8th edition. McGraw - Hill International Edition.

Jung, G. and Laguna, M. 2003. Time segmenting heuristic for an aircraft landing problem, Working paper, Leeds School of Business, University of Colorado, Boulder, CO 80309-0419, USA.

Pinol, H. and Beasley, J.E. 2006. Scatter search and bionomic algorithms for the aircraft landing problem. European Journal of Operational Research, 171.

Soander, J., Beasley, J.E. and Havelock, P. 2001. Scheduling aircraft landings at London Heathrow using a population heuristic. Journal of the Operation Research Society, 52.

Clausen, J. 2007. Scheduling Aircraft Landings, The Dynamic Case. Yüksek Lisans Tezi(Basılmamış). Technical University of Denmark, Department of Informatic and Mathematical Modelling, 7, Lyngby.

Sharaiha, Y.M., Beasley, J.E., Krishnamoorthy, M. and Abramson, D. 2004. Displacement problem and dynamically scheduling aircraft landings. Journal of the Operation Research Society, 55.



EK 1 Geliştirilmiş Açgözlü Algoritma Örnek Verileri ve Sonuçları

Örnek 1:

İniş Sırası Bekleyen Uçaklar:

{1, 4, 1, 3, 2, 4, 2}

Ayrma Zamanları:

$$\begin{bmatrix} 2 & 4 & 1 & 3 \\ 1 & 12 & 3 & 6 \\ 13 & 7 & 4 & 2 \\ 1 & 2 & 6 & 8 \end{bmatrix}$$

EGİZ Değerleri:

{5, 10, 7, 12, 18, 15, 20}

Sonuçlar:

İGİH Zamanı: 20

EGİZ Sonrası İnen Uçak Sayısı: 1

Geliştirilmiş Açgözlü Algoritma Zamanı: 15

Geliştirilmiş Açgözlü Algoritma Uygulandığında EGİZ Sonrası İnen Uçak Sayısı: 0

Örnek 2:

İniş Sırası Bekleyen Uçaklar:

{1, 1, 2, 1, 3, 2, 3}

Ayrma Zamanları:

$$\begin{bmatrix} 1 & 5 & 4 \\ 3 & 7 & 2 \\ 6 & 4 & 1 \end{bmatrix}$$

EGİZ Değerleri:

{5, 10, 7, 12, 18, 15, 20}

Sonuçlar:

İGİH Zamanı: 19

EGİZ Sonrası İnen Uçak Sayısı: 1

Geliştirilmiş Açgözlü Algoritma Zamanı: 17

Geliştirilmiş Açgözlü Algoritma Uygulandığında EGİZ Sonrası İnen Uçak Sayısı: 0

Örnek 3:

İniş Sırası Bekleyen Uçaklar:

{2, 1, 3, 1, 4, 2, 1};

Ayrma Zamanları:

$$\begin{bmatrix} 2 & 4 & 1 & 3 \\ 1 & 12 & 3 & 6 \\ 13 & 7 & 4 & 2 \\ 1 & 2 & 6 & 8 \end{bmatrix}$$

EGİZ Değerleri:

{7, 11, 5, 14, 19, 14, 18}

Sonuçlar:

İGİH Zamanı: 21

EGİZ Sonrası İnen Uçak Sayısı: 3

Geliştirilmiş Açgözlü Algoritma Zamanı: 14

Geliştirilmiş Açgözlü Algoritma Uygulandığında EGİZ Sonrası İnen Uçak Sayısı: 1

Örnek 4:

İniş Sırası Bekleyen Uçaklar:

{2, 1, 3, 1, 4, 2, 1}

Ayrma Zamanları:

$$\begin{bmatrix} 4 & 2 & 13 & 2 \\ 3 & 11 & 13 & 4 \\ 1 & 2 & 2 & 12 \\ 5 & 1 & 2 & 4 \end{bmatrix}$$

EGİZ Değerleri:

{5, 10, 7, 12, 18, 15, 20}

Sonuçlar:

İGİH Zamanı: 23

EGİZ Sonrası İnen Uçak Sayısı: 5

Geliştirilmiş Açgözlü Algoritma Zamanı: 13

Geliştirilmiş Açgözlü Algoritma Uygulandığında EGİZ Sonrası İnen Uçak Sayısı: 0

ÖZGEÇMİŞ

Adı Soyadı : Beraat ALDEMİR

Doğum Yeri : ALTINDAĞ

Doğum Tarihi : 12.03.1990

Medeni Hali : Evli

Yabancı Dili : İNGİLİZCE

Eğitim Durumu

Lise : Çubuk Anadolu Lisesi, 2008

Lisans : Ankara Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, 2013

Yüksek Lisans : Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı (Haziran 2018)

Çalıştığı Kurum/Kurumlar ve Yıl

TÜBİTAK ULAKBİM, 2013 - Halen

Uluslararası Kongre Sunum

Aldemir, B. and Amrahov, S.E. 2018. International Conference on Mathematics and Mathematics Education (ICMME-2018). Design and simulation of real time scheduling of aircraft landing. 27-29 June 2018; Ordu/Turkey.